Games for Syntactic Control of Interference

Matthew Wall

Submitted for the degree of D. Phil.

University of Sussex

30th September 2004

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signature:

Abstract

This thesis proposes a game semantics for languages that adhere to the design principles first proposed by John Reynolds in Syntactic Control of Interference. A pair of subterms of a program written in a language adherent to the SCI principles enjoy the property that the evaluation of one will have no impact on the subsequent evaluation of the other if they contain no free identifier in common.

The proposed model relies upon a novel notion of view based on partial orders of moves, rather than sequences of moves, to capture the inability of programs to discern the order of execution of non-interfering side-effects. Three languages from the literature that obey the SCI principles are interpreted in the model:

PCF This language controls interference by eschewing any imperative features.

Basic SCI At the other extreme, this language outlaws aliasing and thus prevents subterms containing distinct free identifiers from accessing the same parts of the store.

SCIR Aliasing is outlawed for those identifiers that can affect the store but may be permissible for those that cannot. It incorporates a sophisticated notion of passive type; containing no phrases which assign to non-local store. This notion is distinct from that of passive use: identifiers in the context of a typing judgement that are perhaps not of passive type but that are used only passively in the term.

The games model defined in this thesis is shown to possess the definability property for each of the three languages: every finite element of the model is the denotation of a term of the language in question. Using this property, the model can be used to characterize program equivalence in each of the languages precisely; that is, a fully abstract model can be built.

Acknowledgements

Many thanks to my supervisor Guy McCusker, whose support clearly went far beyond what is expected of a supervisor. He introduced me to the topic of this thesis and patiently answered my questions and explained those things that I needed to know.

I would also like to mention my examiners, Luke Ong and Jim Laird, and thank them for taking the time to read this thesis, and for their helpful comments and suggestions for improvement.

I am also grateful to everyone else in the department, especially my Thesis Committee and all the friends I have made during my studies.

Thanks also to the EPSRC whose financial support made this thesis possible.

Finally, thanks to Kate and my family for their patience and support throughout.

Contents

List of Tables

Chapter 1

Introduction

1.1 Interference

At the very heart of imperative programming lies the idea that the evaluation of one subterm of a program may later affect the evaluation of another subterm. We would have no wish to assign to a store if there existed no channel by which this assignment might affect the outcome of the program. We intend the term interference to take the meaning defined in [40]: a term M_1 is said to interfere with a term M_2 if exercising M_1 can have an effect on the result of later exercising M_2. Exercising is taken as a general term to mean evaluating, executing, calling, assigning to or dereferencing a subterm of our program — whichever action is appropriate for the type of the subterm. However, interference is commonly seen as a stumbling block when we wish to reason about programs. Consider the following program:

$$x := 5; !x$$

The variable x is first assigned the value 5 and then the variable is dereferenced. The two sequenced terms clearly interfere but we can reason about such stateful behaviour quite happily using conventional Hoare logic [26]. This logic provides a framework in which we can reason about simple imperative languages that do not possess procedures. Only when procedures are added to the language do we start to encounter problems: it is only in the presence of these two factors, procedures and state, that the particular form of interference that Reynolds describes in [47] can occur. Reynolds argues that reasoning about imperative languages only becomes really problematic when the language permits interference which is not syntactically obvious. This occurs when distinct identifiers in a term access the same resource in some way. The term *covert interference* is used to describe this phenomenon. The simplest example of covert interference is caused by aliasing. Aliasing occurs when distinct identifiers become bound to the same resource. We can demonstrate how such interference may create problems with the following example. Consider the term $y := 2 \times !x$. We might wish to use some axiomatic semantic theory to make a statement to the effect that after running this command the value stored in y is twice that stored in x. However,

this statement would be incorrect in the following program:

$$\mathsf{new}\ y := 0\ \mathsf{in}\ (\lambda x.x := 5; y := 2 \times !x)(y)$$

where x becomes bound to y. We should note here that interference can be more complex than this simple aliasing of variables. Subterms of product and procedure can also interfere if they side effect the store. The problems in formulating a Hoare style logic have been remedied somewhat by Reynolds in [48] where additional logical statements that insist upon non-interference are included in the language of assertions of the axiomatic semantics. However, it is vital that we are able to reason about the non-interference of subterms.

O'Hearn *et al.* [40] note that we may also imagine that we would wish to syntactically forbid *bad variables* from language. Bad variables are those terms of the type of storage variables that, when dereferenced, do not necessarily return the most recently assigned value. The classic example is the array variable $A[A[i]]$ which behaves badly when $A[i]$ is i. However, if we want to disallow such bad variables then we must also be able to tell that $A[j]$ is illegal when j aliases $A[i]$.

Reynolds [47] also notes that the situation becomes considerably more complex when union types are included in the language and provides the following example where x is a variable that holds a value of either an *integer* or a *character*.

$$\textbf{unioncase}\ x\ \textbf{of integer}\ :(y := \text{`}A\text{'}; n := !x + 1)\ \ \textbf{character}\ :\ \mathsf{skip}$$

The intended meaning of this statement is that the program branches according to the *type* of x. However, consider a situation where x and y are aliases. If x is an integer and the first branch is taken we run into a typing error when the expression $!x + 1$ is evaluated.

It should be noted that control of interference is desirable for more reasons than the axiomatic reasoning and type structure problems described above. It is suggested that interfering side effects are a common source of programming error. Perhaps it would be desirable for the programmer to be able to have at their disposal some syntactic decoration in order to specify whether or not certain identifiers may interfere. Furthermore, it may be desirable for a programming language to allow some limited parallelism in the composition of commands when the evaluation order has no effect on the outcome of the program. Of course, if we intend the language to be deterministic, we must certainly be careful about interference between commands thus composed. Reynolds again provides a simple example [47]: the term

$$x := !x + 1 || y := !y \times 2$$

should not be typeable if x and y interfere and we want our operational semantics to be deterministic.

1.2 SCI

In Syntactic Control of Interference [47] Reynolds suggests that the imperative language Idealized Algol [50, 42] could be extended to outlaw those terms in which covert interference is present. Reynolds' elegant idea is to propose a relation, $\sharp$, with the intention that the relation holds between program fragments if it is *syntactically* obvious that they do not interfere.

- If $M \sharp N$ and M and N are not procedures then exercising M will have no effect on the exercise of N.
- If M is a procedure, with arguments $A_1, \ldots, A_n$ such that $M \sharp N \wedge A_1 \sharp N \wedge \ldots \wedge A_n \sharp N$, then $M(A_1, \ldots, A_n) \sharp N$. That is, $M \sharp N$ implies that an application of M will not affect N if its arguments do not affect N.

Note that the above does not form a definition, but merely some conditions that the $\sharp$ relation is duty bound to fulfill. Given such a relation, Reynolds then proposes the following design principles for an SCI language.

SCI1 If $x \sharp y$ for all free identifiers x in M and y in N then $M \sharp N$. This simply states that all channels of interference are named by identifiers: there are no constants or programming constructs that interfere with each other.

SCI2 If x and y are distinct identifiers then $x \sharp y$.

SCI3 Terms of certain types are passive in that they do not assign to any global store. If M and N are both passive then $M \sharp N$.

Reynolds suggests a simple solution to principles **SCI1** and **SCI2** by defining the $\sharp$ relation as follows

$$M \sharp N \Leftrightarrow \mathrm{FV}(M) \cap \mathrm{FV}(N) = \emptyset.$$

That is, the relation holds between M and N if and only if they have no free identifiers in common.

Principle **SCI1** is already present in Idealized Algol-like languages. How can we then ensure that principle **SCI2** is satisfied? We must look at how identifiers are bound. In the illustrative language given in [47] the only two means of binding identifiers are with the **new** construct or by application of a λ expression. The **new** construct gives no problems as binding follows a strict nesting discipline. Function application, as already stated, is at the heart of the problem of covert interference. The solution proposed by Reynolds is simply to restrict the application $M(N)$ to only those phrases such that $M \sharp N$. By ensuring that a function and its argument do not interfere we can ensure that N will not become bound to any identifier that can interfere with other identifiers in the body of M. In modern parlance this is equivalent to using an affine type system. In Idealized Algol, the typing rule for function application is the same as that for the simply typed λ-calculus:

$$\frac{\Gamma \vdash M : \theta' \Rightarrow \theta \quad \Gamma \vdash N : \theta'}{\Gamma \vdash MN : \theta} \Rightarrow E$$

The typing rule for function application suggested by Reynolds is suggestive of the multiplicative form found in Girard's linear logic [23]:

$$\frac{\Gamma \vdash M : \theta' \multimap \theta \quad \Delta \vdash N : \theta'}{\Gamma, \Delta \vdash MN : \theta} \multimap E$$

where writing Γ, Δ implies that Γ and Δ are disjoint. Other language constructs are typed in the additive, or context sharing, form. for example sequential composition:

$$\frac{\Gamma \vdash M : \mathbf{com} \quad \Gamma \vdash N : \mathbf{com}}{\Gamma \vdash M; N : \mathbf{com}} \textit{ sequencing}$$

The variation of Idealized Algol with this multiplicative function application rule has been termed Basic SCI by O'Hearn in [43].

The third design principle, **SCI3**, has proved harder to satisfy. In [47] Reynolds suggests adding a form of contraction for passive types although the resultant language does not enjoy subject reduction. This is remedied by Reynolds in [49] using subtypes in a complicated type system. A later solution, **SCIR**, was proposed by O'Hearn *et al.* in [40] where typing judgements have split contexts and take the form:

$$\Gamma|\Delta \vdash M : \theta$$

with the intuition that identifiers in Γ are only *used* passively in M. Contraction is then permissible only in Γ.

1.3 Denotational Semantics

When we write a program we might expect to know what it does. However, even small programs written in a language with an unambiguous grammar and a well-defined operational semantics can leave us scratching our heads. Of course, well-known computability results imply that we cannot always expect to reason effectively about the exact behaviour of programs written in a Turing complete language, but even simple properties of trivial programs have been hard to prove formally, even when we have a clear intuition about them. Clearly, reasoning about syntax and operational semantics is difficult and there is a need to seek other reasoning methods.

Scott and Strachey's approach [52] is to seek to define a *denotational* model for a given language. An abstract mathematical object, termed the denotation, is assigned to each program; the intention being that it may prove easier to reason about a program's denotation than it is to reason directly about its syntax. The mapping from programs to denotations is usually compositional: it is defined inductively on the syntax or typing derivation, and thus denotations are normally defined for both open and closed terms.

Obviously care must be taken when we choose a denotational semantics for a language. The choice we make here will affect the kind of reasoning that we have at our disposal. Equality in the model of a programming language induces an equivalence relation on the language and it is important that this relation makes sense operationally. If we have two terms with equal denotations we can never use the model to reason about properties that distinguish the terms. Terms with equal denotations should, in some sense, have the same behaviour. Exactly what constitutes the "same behaviour"? The equality in which we will be interested is termed observational equivalence.

Definition 1 (Observational Equivalence) Observational equivalence is defined with respect to operational semantics. We write $M \Downarrow V$ to mean term M converges to value V and we write $M \Downarrow$ to indicate that M converges to some unspecified value. Two terms, M and N, are said to be observationally equivalent, written $M \simeq N$, if and only if, for all contexts of base type, $C[\_]$, we have

$$C[M] \Downarrow V \Leftrightarrow C[N] \Downarrow V$$

Sometimes we may wish to make use of a related ordering on terms: the observational preorder. We write $M \underset{\sim}{\sqsubseteq} N$ to indicate that for all contexts $C[\_]$ we have

$$C[M] \Downarrow V \Rightarrow C[N] \Downarrow V.$$

Obviously we then have

$$N \underset{\sim}{\sqsubseteq} M \wedge M \underset{\sim}{\sqsubseteq} N \Leftrightarrow M \simeq N.$$

We are now ready to consider some properties that a good denotational model might have.

Definition 2 (Syntax Independence) Any language models itself, but of course this achieves little. A model is syntax independent when it is constructed without reference to the syntax of the language, and a mapping is then shown to exist from the terms of the language to objects in the model. Of course the model may be an independent language, with an independently defined operational semantics, but the motivation behind denotational semantics is to transfer our reasoning to structures that are not syntactic and whose properties are not defined operationally.

Definition 3 (Coherence) Sometimes a typing judgement may have two or more derivations. If this is the case and the semantics is defined inductively on derivations then it may be the case that different derivations of a given judgement assign different semantics to the same term. A semantics is coherent if and only if the semantics assigned to a term is independent of the derivation of the judgement.

Definition 4 (Equational Soundness) We say that a denotational model is equationally sound if and only if

$$[\![M]\!] = [\![N]\!] \Rightarrow M \simeq N.$$

Equational soundness is fundamental if we wish to use the model to reason about contextual equivalence.

Definition 5 (Inequational Soundness) Sometimes it is possible to define a partial order, $\leq$, on denotations in our model. We normally insist that the partial order, $\leq$, is compositional; for any context $C[\_]$ we have

$$[\![M]\!] \leq [\![N]\!] \Rightarrow [\![C[M]]\!] \leq [\![C[N]]\!].$$

We say that our model is inequationally sound if and only if for terms M and N we have

$$[\![M]\!] \leq [\![N]\!] \Rightarrow M \underset{\sim}{\sqsubseteq} N.$$

It is then trivial to show that inequational soundness implies equational soundness.

Definition 6 (Equational Completeness) We say that a denotational model is equationally complete if and only if

$$M \simeq N \Rightarrow [\![M]\!] = [\![N]\!].$$

It may be desirable that a model is complete. It implies that observationally equivalent terms have equivalent denotations. We can thus reason about observational inequivalence. Equational completeness in the absence of equational soundness is rarely considered.

Definition 7 (Full Abstraction) We say that a semantic model is fully abstract if it is both equationally sound and equationally complete:

$$M \simeq N \Leftrightarrow [\![M]\!] = [\![N]\!].$$

A fully abstract model in some sense gives a full account of what the language means: it defines the language. Syntax independent, fully abstract models have proved to be rare animals. Sometimes we informally talk about degrees of abstraction: one model may be more abstract than another. A model is less abstract than another if it distinguishes between more programs that are observationally equivalent. It is often considered desirable to construct a model that is as abstract as possible.

Definition 8 (Operational Soundness) We say that a model is operationally sound if and only if

$$M \Downarrow V \Rightarrow [\![M]\!] = [\![V]\!].$$

Note that the direction of the implication contrasts with that in the definition of equational soundness.

Definition 9 (Computational Adequacy) Suppose that the partial order on denotations has a least element, written $\perp$.

We say that a model is computationally adequate if for any term M of base type we have

$$[\![M]\!] \neq \perp \Rightarrow M \Downarrow .$$

Operational soundness and computational adequacy together imply equational soundness. Given terms M and N such that $[\![M]\!] \leq [\![N]\!]$ and any context $C[\_]$ such that $C[M]\Downarrow$, we have $[\![C[M]]\!] \neq \perp$ by operational soundness. Compositionality ensures that $[\![M]\!] \leq [\![N]\!]$ implies $[\![C[M]]\!] \leq [\![C[N]]\!]$. Therefore $[\![C[N]]\!] \neq \perp$ and by adequacy $C[N]\Downarrow$ and hence $M \underset{\sim}{\sqsubset} N$.

Definition 10 (Universality) A model is universal if every element in our model is the denotation of a term in our language. Sometimes we might have a weaker property: every element in the model that satisfies a certain condition is the denotation of a term. Such a property is termed a definability property.

1.4 An Informal Introduction to Game Semantics

Game semantics is a branch of denotational semantics in that programs are mapped onto formally defined mathematical objects. The interpretation of a typed term is a set of sequences of symbols, possibly with additional structure added to the sequences and symbols. We will later see how the placing of various combinatorial constraints on these sets has been used to create definability results. Fully abstract models have been successfully formulated for a wide selection of programming languages using this approach [28, 39, 4, 5, 6, 1, 9].

We start this section with an informal introduction to game semantics of programming languages. For a considerably more in depth discussion see Abramsky and McCusker's tutorial [9]. The games that we describe here are after the style of Hyland and Ong [28] and Nickau [39] and are related to their fully abstract models of PCF. A contrasting style was pioneered by Abramsky, Jagadeesan and Malacaria [4] in their fully abstract model for PCF.

1.4.1 The Protagonists

At the heart of game semantics is the notion that the behaviour of a program fragment can be considered as the set of all possible interactions between the fragment and its environment. These sets are termed *strategies* in game semantics. A program may need to ask the environment about the value of certain identifiers or it may need to apply some higher order identifier to some arguments. The environment may need to ask the program about arguments to some of these higher order identifiers and so on. We should notice that the set of such traces gives the *intensional* meaning of a program.

To further extend the recreational metaphor we refer to the environment as *opponent* (O) and the program as *player* (P) - (or proponent in some presentations).

We are now going to give a flavour of the way in which terms of the call-by-name simply typed λ-calculus, enriched with a sprinkling of common programming constructs, might be interpreted.

1.4.2 Modelling Values

Consider what possible semantics we might have for a program of natural number type, $\mathbf{N}$. In the games that we shall consider opponent always asks the first question and we denote this as q. Player can then reply with its answer which will be one of the natural numbers. Consider the following typed term:

$$\vdash 5 : \mathbf{N}$$

The idea is that this program will respond to an initial opponent question q with an answer 5. As stated, we model programs as sets of possible traces, or *strategies*, and in the models we concern ourselves with these strategies are always closed with respect to even length prefixes of the sequences, so we can define the semantics of this term as $\{\varepsilon, \mathsf{q} \cdot 5\}$, where ε represents the empty sequence. A divergent closed program of type $\mathbf{N}$ would simply have the semantics $\{\varepsilon\}$.

1.4.3 Modelling Functions

Consider some function of type $\mathbf{N} \Rightarrow \mathbf{N}$, for example the successor function. In this case we imagine that opponent initially asks for the output to this program. Obviously output generally depends on input, so player asks for input. When opponent responds with a value, player can then respond with a value one greater. The semantics can therefore be given as

$$\{\mathsf{q} \cdot \mathsf{q} \cdot n \cdot n+1 | n \in \mathbf{N}\} \cup \{\varepsilon, \mathsf{q} \cdot \mathsf{q}\}$$

When we wish to depict traces of a program we often set them out in tabular format with the type of the program across the top and the moves placed underneath the atomic type to which they refer. The sequences in such a format grow downwards. This is done purely for reasons of readability. For example, one trace from the strategy modelling the successor function could be depicted as

follows.

N	⇒	**N**
		q
q		
5		
		6

No extra information is transmitted because, as we shall see later, the set of moves of a game is the *disjoint* union of the moves of its subgames.

How then, might we model constant functions? Consider the term $\lambda x.2$ that always returns 2. Its strategy will contain two sequences, ε and the following:

N	⇒	**N**
		q
		2

However, there are other functions that always return 2. Consider the term $\lambda x.\mathsf{if}\ x = 0\ \mathsf{then}\ 2\ \mathsf{else}\ 2$. This term is not observationally equivalent to $\lambda x.2$: it diverges if applied to a divergent argument. The strategy interpreting this term will comprise even-length prefixes of sequences of the form:

N	⇒	**N**
		q
q		
n		
		2

Functions that return function types are no problem. Consider this trace taken from a strategy for the curried form of addition.

N	⇒	**N**	⇒	**N**
				q
q				
3				
		q		
		5		
				8

We can also model functions that take functions as arguments. Consider a function that takes a function as its argument and applies it to the number 3.

(**N**	⇒	**N**)	⇒	**N**
				q
		q		
q				
3				
		8		
				8

When we have higher order arguments it is possible for runs to be interleaved. Consider the program $\lambda f.f(f3)$ with the following trace (in this sequence f is playing like the successor function):

$(\mathbf{exp}$	$\Rightarrow$	$\mathbf{exp})$	$\Rightarrow$	$\mathbf{exp}$
				q
		q		
q				
		q		
q				
3				
		4		
4				
		5		
				5

1.4.4 Modelling Products

A strategy for a term of product type is comprised of sequences that are either a sequence in one component or a sequence in the other. For example the term $\langle 2,5\rangle$ has a strategy containing ε and the following two traces:

$\mathbf{N}$	$\times$	$\mathbf{N}$
q		
2		

$\mathbf{N}$	$\times$	$\mathbf{N}$
		q
		5

All the terms we have considered so far have been closed, but the semantics can be extended to interpret open terms. A judgement

$$x_1 : A_1, \ldots, x_n : A_n \vdash M : B$$

can be interpreted in much the same way as the curried term

$$\vdash \lambda x_1 \ldots \lambda x_n.M : A_1 \Rightarrow \ldots \Rightarrow A_n \Rightarrow B$$

1.4.5 Justification

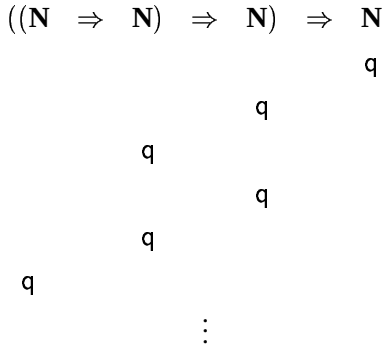
Unfortunately, to fully model the kinds of languages that we are interested in we need a little extra information to be included in the sequences. To demonstrate this we shall examine an example taken directly from [9], that was first described by Kirstead.

Let us consider the following two λ-terms:

$$\lambda f.f(\lambda x.f(\lambda y.y))$$

$$\lambda f.f(\lambda x.f(\lambda y.x))$$

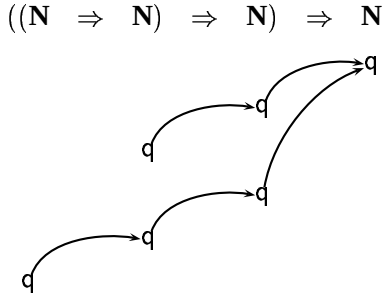
It transpires that the game semantics, as described so far, would equate these two terms. However they are *not* contextually equivalent in the sort of languages that we are interested in. The interpretation of either term contains sequences that are prefixes or extensions of the following form:



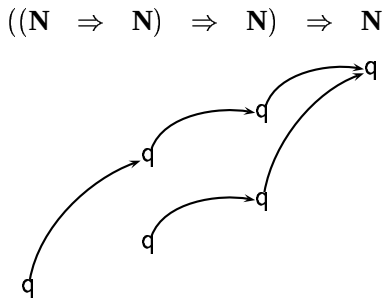
The problem lies with the last move. It queries the argument of the argument of f, but we cannot distinguish which occurrence of f in our term we are considering. We are, therefore, confused as to which subterm the argument indicated by the move belongs.

The confusion can be remedied by attaching, to each occurrence of a move in a sequence corresponding to a request for input, a pointer from the occurrence of the move that requires that input. We also include pointers to occurrences of answer moves from their corresponding occurrences of questions. We say that a move justifies those moves from which it has pointers.

Happily , the above λ-terms thus have the following distinguished plays. The term $\lambda f.f(\lambda x.f(\lambda y.y))$ plays like this:



whereas $\lambda f.f(\lambda x.f(\lambda y.x))$ has the following play:



1.5 Thesis Outline

The main contribution of this thesis is the construction of a games model that can be used to interpret three languages which conform to the SCI design principles of section 1.2.

In Chapter 2 we introduce a novel category of games, **G**, where moves are permitted to have more than one justifier. We show that the well known definitions of visibility and innocence can be naturally extended to this new setting where views are no longer sequences but partial orders of moves. We then show that visibility and innocence are preserved by composition and that identities in this category respect player visibility and innocence.

We then introduce a category of arenas, **A**, in Chapter 3. We demonstrate ways in which we can constrain the notion of strategy found in **A** and develop novel methods for proving properties about composition and identities for strategies that are thus constrained.

Chapter 4 introduces Reynolds' Idealized Algol [50] and recalls Abramsky and McCusker's games model for Idealized Algol [6].

In Chapter 5 we re-examine the design principles proposed by Reynolds in Syntactic Control of Interference [47] and introduce three languages that adhere to these principles: PCF, Basic SCI and SCIR. We examine the categorical structure that we might require of models of each of these languages and discuss some of the existing models.

In Chapters 6 and 7 we introduce the main contribution of this thesis: a games model for SCI languages. We show how our model tallies with our intuitions about the type systems and demonstrate that the model is sound and adequate. Definability results have been central to the success of game semantics as a discipline and we are able to prove definability and thus build a fully abstract model.

Finally, we present our conclusions in Chapter 8 and suggest further avenues for research.

Chapter 2

Games and Multiple Justifiers

2.1 Introduction

In this chapter we introduce a novel category of games, **G**, in which strategies are comprised of sequences in which any move occurrence may have more than one justifier. We find that the properties of visibility and innocence found in [28] can be extended in a natural way to fit this new setting and that these properties are possessed by the identity strategies and are preserved by composition. Although it would be possible to construct sound models for programming languages in **G**, we do not use this category as it is insufficiently constrained to get the definability results that we desire. There is just too much "junk" in **G**: too many morphisms that are not the denotations of any term. We will however use results obtained in this chapter to prove properties in the more constrained models of chapter 3 and 6. The use of **G** in chapter 3 may appear like taking a sledgehammer to a walnut; we will encounter a coconut in chapter 6.

2.1.1 Notation

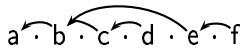
When we define our games models, we will be interested in sequences of moves from a particular set. We will tend to use m, m', n, n' for moves, and sometimes use q, q' or a, a' etc. when we wish to imply that a move is a *question*, or respectively an *answer*. We make no notational distinction between *occurrences* of moves in a sequence, singleton sequences and the moves themselves; no confusion should arise in practice. m^+ is the move immediately following m in a sequence; m^- is the move immediately preceding m. Sequences of moves will be notated s, s', t, t'. Concatenation of sequences is written $s \cdot t$; ε is the empty sequence. $m \in s$ means that move m occurs in s. We use $<, >, \leq, \geq$ when we wish to talk about the ordering of occurrences of moves within a sequence. $s_{<m}$ denotes the subsequence of those move occurrences of s that occur strictly before m. $s_{\leq m}$ is defined similarly.

When we have a sequence s of moves from a game $A + B$ we write $s \upharpoonright A$ for that subsequence of s containing only those moves from M_A. We define $s \upharpoonright B$ similarly.

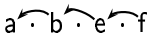
We use $\sqsubseteq$ to denote the usual prefix ordering on sequences and we write $s \sqsubseteq_{even} s'$ if and only if $s \sqsubseteq s'$ and the length of s is even.

2.1.2 Multiple Justifiers

In the familiar setting, where moves have at most one justifier, a view of a sequence is a subsequence. For example given the following justified sequence:

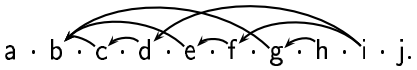


the player view would be as follows

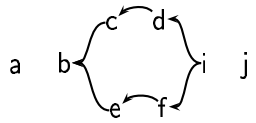


and an innocent strategy containing such a view must play according to this partial information about the original sequence.

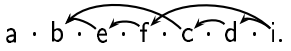
Unfortunately to appreciate how we construct views where moves may have more than one justifier we have to consider fairly complicated sequences such as the following sequence s:



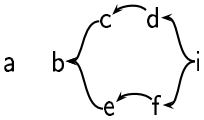
In this case our player view will be a *partial ordering* on a subset of the moves from the original sequence. Every player move in the partial order will be preceded immediately by its predecessor from the original sequence and every opponent move will be preceded immediately by *each* of its justifiers. We therefore arrive at the following player view of s:



Once again we can consider the player view to contain partial information about the original sequence. In this case we not only lose some of the moves from s but also we lose information about the ordering of some of the moves in the view. As before, any innocent strategy must play functionally according to the player view. Suppose that a strategy σ containing s also contains a sequence s' and that $s' \cdot i$ is a *valid* extension of s' (we will define validity later) and that $s' \cdot i$ has the following form:



When we construct the player view of $s' \cdot i$ we get the following:



and as σ is innocent player must respond according to this view and play the move j in response and hence we must have $s' \cdot i \cdot j \in \sigma$.

2.2 Games and Pre-arenas

Definition 11 (Pre-arena) A pre-arena A is a tuple $\langle \mathsf{M_A}, \lambda_\mathsf{A} \rangle$ where:

- $\mathsf{M_A}$ is a set of moves.
- $\lambda_\mathsf{A} : \mathsf{M_A} \to \{O, P\}$ is a labelling function that indicates whether a move is an opponent move or a player move.

We define $\overline{\lambda_\mathsf{A}} : \mathsf{M_A} \to \{O, P\}$ as follows:

$$\overline{\lambda_\mathsf{A}}\mathsf{m} = O \Leftrightarrow \lambda_\mathsf{A}\mathsf{m} = P.$$

Definition 12 (Justified Sequence) A justified sequence, s, for a pre-arena A is a sequence of moves from $\mathsf{M_A}$ together with a collection of pointers known as justifiers such that any two move occurrences $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ may be connected by a justifier, written $\mathsf{m} \curvearrowleft \mathsf{m}'$. Sometimes it will be useful for us to specify a justified sequence s as a tuple $\langle M, \lhd, \curvearrowleft \rangle$ where:

- M_s is a set of move occurrences.
- $\lhd_\mathsf{s} \subseteq M_\mathsf{s} \times M_\mathsf{s}$ is the successor relation of the sequence.
- $\curvearrowleft_\mathsf{s} \subseteq M_\mathsf{s} \times M_\mathsf{s}$ is arrow relation on M_s.

We write $\mathcal{X}_\mathsf{A}$ for that set of justified sequences for the pre-arena A such that each sequence $\mathsf{s} \in \mathcal{X}_\mathsf{A}$ satisfies the following:

- $\mathsf{s} = \mathsf{m} \cdot \mathsf{s}' \Rightarrow \lambda\mathsf{m} = O$
- $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}' \cdot \mathsf{s}'' \Rightarrow \lambda\mathsf{m} \neq \lambda\mathsf{m}'$. We term this property *alternation*.
- $\forall \mathsf{m}, \mathsf{m}' \in \mathsf{s}. \mathsf{m} \curvearrowleft \mathsf{m}' \Rightarrow \mathsf{m} \leq \mathsf{m}'$
- $\forall \mathsf{m}, \mathsf{m}' \in \mathsf{s}. \mathsf{m} \curvearrowleft \mathsf{m}' \Rightarrow \lambda\mathsf{m} \neq \lambda\mathsf{m}'$

Definition 13 (Games) A game A is a tuple $\langle \mathsf{M_A}, \lambda_\mathsf{A}, \mathcal{P}_\mathsf{A} \rangle$ where:

- $\langle \mathsf{M_A}, \lambda_\mathsf{A} \rangle$ is a pre-arena.
- $\mathcal{P}_\mathsf{A}$ is a subset of $\mathcal{X}_\mathsf{A}$ known as the valid plays.

Definition 14 Given games A and B we define the game $\mathsf{A} \to \mathsf{B} = \langle \mathsf{M_{A \to B}}, \lambda_{\mathsf{A} \to \mathsf{B}}, \mathcal{P}_{\mathsf{A} \to \mathsf{B}} \rangle$ as follows:

$$\begin{array}{rcl}
\mathsf{M_{A \to B}} & = & \mathsf{M_A} + \mathsf{M_B} \\
\lambda \mathsf{A} \to \mathsf{B} & = & [\overline{\lambda_\mathsf{A}}, \lambda_\mathsf{B}] \\
\mathsf{s} \in \mathcal{P}_\mathsf{A} & \Leftrightarrow & \mathsf{s} \in \mathcal{X}_{\mathsf{A} \to \mathsf{B}} \\
 & & \wedge\, \mathsf{s} \upharpoonright \mathsf{A} \in \mathcal{P}_\mathsf{A} \\
 & & \wedge\, \mathsf{s} \upharpoonright \mathsf{B} \in \mathcal{P}_\mathsf{B} \\
 & & \wedge\, \forall \mathsf{m}, \mathsf{m}' \in \mathsf{s}. \mathsf{m} \curvearrowleft_\mathsf{s} \mathsf{m}' \Leftrightarrow (\mathsf{m} \curvearrowleft_{\mathsf{s} \upharpoonright \mathsf{A}} \mathsf{m}' \vee \mathsf{m} \curvearrowleft_{\mathsf{s} \upharpoonright \mathsf{B}} \mathsf{m}')
\end{array}$$

Where we define the projection $s \upharpoonright A$ to be that subsequence of s containing exactly those moves from M_A. We define $s \upharpoonright B$ similarly. Implicit in our definition is that there is no pair of moves $m, m' \in s$ such that $m \in s \upharpoonright A$ and $m' \in s \upharpoonright B$ where $m \curvearrowleft m'$ or $m' \curvearrowleft m$.

Lemma 15 (The Switching Condition) Given games A and B and a sequence $s \in \mathcal{P}_{A \to B}$ with consecutive moves $m, m^+ \in s$ such that $m \in s \upharpoonright A$ (respectively $m \in s \upharpoonright B$) and $m^+ \in s \upharpoonright B$ (respectively $m^+ \in s \upharpoonright A$) we have $\lambda_{A \to B} m^+ = P$.

Proof We consider only the case where $m^+ \in s \upharpoonright B$ as the other case is similar. We construct a proof by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is trivial.

Inductive Step If $s = s' \cdot n$ then we can apply the inductive hypothesis if $m^+ \in s'$. Otherwise we have $m^+ = n$. Let the predecessor of m^+ in $s \upharpoonright B$ be m'. By assumption we do not have $m' = m$ so there must be some sequence of moves from M_A strictly between m' and m^+. By the inductive hypothesis we see that the first move of this sequence is a player move, hence m' is an opponent move. We know that $s \upharpoonright B \in \mathcal{P}_B$ and hence $s \upharpoonright B$ alternates therefore $\lambda_{A \to B} m^+ = P$. ■

2.3 Strategies, Composition and Identity

2.3.1 Strategies

Definition 16 (Strategy) A strategy σ for a game A, written $\sigma : A$, is a set of even length sequences from $\mathcal{P}_A$.

2.3.2 Composition

Definition 17 Given games A, B and C we define $\mathbf{int}(A, B, C)$ to be the set $\mathcal{P}_{(A \to B) \to C}$.

Similarly, if we are also given a game D we define

$$\mathbf{int}(A, B, C, D) = \mathcal{P}_{((A \to B) \to C) \to D}.$$

Given a sequence $s \in \mathbf{int}(A, B, C)$ we define the projections $s \upharpoonright A, s \upharpoonright A, C$ etc. in a similar fashion to the projections from sequences from $A \to B$. Given a sequence $s \in \mathbf{int}(A, B, C)$ we say that a move $m \in s$ is in a component A, B if $m \in s \upharpoonright A, B$ and similarly in component B, C if $m \in s \upharpoonright B, C$. In some of the ensuing proofs we refer to components X and Y with the understanding that these terms refer to these two different components but for the purposes of the proof it is not important which is which.

Definition 18 (Interaction Sequence) Given games A, B and C with strategies $\sigma : A \to B$ and $\tau : B \to C$ we define the set of *interaction sequences* as follows

$$\sigma \| \tau = \{ s \in \mathbf{int}(A, B, C) \mid s \upharpoonright A, B \in \sigma \wedge s \upharpoonright B, C \in \tau \}$$

Lemma 19 Given games A, B and C with strategies $\sigma : A \to B$ and $\tau : B \to C$ and sequence $s \cdot m \in \sigma \| \tau$ it follows that $m \notin s \cdot m \upharpoonright B$.

Proof As $s \cdot m \upharpoonright A, B$ and $s \cdot m \upharpoonright B, C$ both have even length and hence end with player moves with

respect to $\lambda_{A\to B}$ and $\lambda_{B\to C}$ respectively. However, any move from $s\cdot m \upharpoonright B$ will receive different labellings from $\lambda_{A\to B}$ and $\lambda_{B\to C}$. ■

Definition 20 (Composition) Given games A, B and C with strategies $\sigma : A \to B$ and $\tau : B \to C$ we define composition as follows

$$\sigma;\tau = \{s \upharpoonright A, C \mid s \in \sigma \| \tau\}$$

In this case we say that the sequence s is a *witness* to the sequence $s \upharpoonright A, C$.

Lemma 21 Given strategies $\sigma : A \to B$ and $\tau : B \to C$, an interaction sequence $s \in \sigma \| \tau$ and moves $m \in s \upharpoonright A$ and $m' \in s \upharpoonright C$, then m and m' are not consecutive in s.

Proof We prove only the case where $m < m'$ as the other case is similar. As $s \upharpoonright A, B \in \mathcal{P}_{A\to B}$ the first move in this sequence is from B. Let n be the greatest move in $s \upharpoonright B$ such that $n < m$. The switching condition of lemma 15 implies that $\lambda_{A\to B} n = O$ and hence $\lambda_{B\to C} n = P$. Lemma 15 then implies that there must exist some move in $s \upharpoonright B$ between n and m'. Hence m and m' are not consecutive. ■

Lemma 22 Given strategies $\sigma : A \to B$ and $\tau : B \to C$, an interaction sequence $s \in \sigma \| \tau$ and moves m and m' that are consecutive in $s \upharpoonright A, C$ then $m, m' \in s \upharpoonright A$ or $m, m' \in s \upharpoonright C$ if and only if that subsequence of s lying strictly between m and m' has even length.

Proof Let t be the subsequence of moves strictly between m and m'. We assume without loss of generality that $m \leq m'$.

case: Suppose $m, m' \in s \upharpoonright A$. If t is not empty then the switching condition assures us that the first move is a player move and the last an opponent move, with respect to $\lambda_{A\to B}$, and as $s \upharpoonright A, B \in \mathcal{X}_{A\to B}$ it follows that t has even length.

case: If $m, m' \in s \upharpoonright C$ we can apply a similar argument to the above.

case: Now suppose $m \in s \upharpoonright A$ and $m' \in s \upharpoonright C$. Lemma 21 assures us that there is at least one move between m and m'. We see that lemma 15 dictates that the first move in t is a player move when labelled by $\lambda_{A\to B}$ and the last move is an opponent move when labelled by $\lambda_{B\to C}$ hence the sequence t has odd length.

case: If $m \in u \upharpoonright C$ and $m' \in u \upharpoonright A$ we can apply a similar argument to the above. ■

Lemma 23 Given games A, B and C with strategies $\sigma : A \to B$ and $\tau : B \to C$, it is the case that $\sigma;\tau$ is a strategy for $A \to C$.

Proof Let s be any sequence in $\sigma;\tau$ and u be any witness to s. First we show that $\sigma;\tau \subseteq \mathcal{X}_{A\to C}$ as follows:

- s is a justified sequence by definition.
- We now show that if s is not empty then it starts with an opponent move. If s is not empty then the first move in $u \upharpoonright A$ must be preceded by some move in $u \upharpoonright B$ as $u \upharpoonright A, B \in \sigma$. Similarly any move in $u \upharpoonright B$ is preceded some move in $u \upharpoonright C$ and the first move in $u \upharpoonright C$ is an opponent move as $u \upharpoonright B, C \in \tau$.

- We now show that s obeys alternation. Consider any two consecutive moves $m, m' \in u \upharpoonright A, C$. If $m, m' \in u \upharpoonright A$ or $m, m' \in u \upharpoonright C$ then the labelling of the moves alternates by virtue of the fact that $u \upharpoonright A$ and $u \upharpoonright C$ both alternate. Otherwise we must have $m \in u \upharpoonright A$ and $m' \in u \upharpoonright C$ or vice versa, and we simply apply lemma 22.

We now show that $s \in \mathcal{P}_{A \to C}$. As $\sigma \subseteq \mathcal{P}_{A \to B}$ we have $u \upharpoonright A, B \in \mathcal{P}_{A \to B}$ and by the definition of $\mathcal{P}_{A \to C}$ we have $u \upharpoonright A \in \mathcal{P}_A$. Similarly we have $u \upharpoonright C \in \mathcal{P}_C$ and hence by definition $s \in \mathcal{P}_{A \to C}$.

To show that $\sigma;\tau$ is a strategy we are left to show that s has even length. We know that both $u \upharpoonright A, B$ and $u \upharpoonright B, C$ are of even length. We now have

$$|s| + 2 \times |u \upharpoonright B| = |u \upharpoonright A, B| + |u \upharpoonright B, C|$$

hence $|s|$ must be even. ■

Lemma 24 Composition is associative. Given games A, B C and D with strategies $\rho : A \to B$, $\sigma : B \to C$ and $\tau : C \to D$ we have

$$(\rho;\sigma);\tau = \rho;(\sigma;\tau)$$

Proof We first construct the following set:

$$S = \{s \in \mathbf{int}(A, B, C, D) | s \upharpoonright (A, B) \in \rho \wedge s \upharpoonright (B, C) \in \sigma \wedge s \upharpoonright (C, D) \in \tau\}.$$

It is fairly straightforward to show that $\{s \upharpoonright A, D | s \in S\} = (\rho;\sigma);\tau = \rho;(\sigma;\tau)$. ■

2.3.3 Identities

Definition 25 ($\mathbf{id}_A$) Given a game A we define $\mathbf{id}_A : A \to A$ to be the following set of sequences:

$$\{s \in \mathcal{P}_{A' \to A''} \mid \forall t \sqsubseteq_{\mathsf{even}} s.t \upharpoonright A' = t \upharpoonright A''\}$$

where A' and A'' are used to distinguish the different copies of game A. The strategy $\mathbf{id}_A$ is an example of a copycat strategy. For any sequence in the strategy, each move that opponent makes is copied by player in the other copy of A.

Lemma 26 The copycat strategy is the unit of the composition operation. Given games A and B and a strategy $\sigma : A \to B$ we have

$$\sigma = \mathbf{id}_A;\sigma = \sigma;\mathbf{id}_B$$

Proof By inspection of the definitions of composition and copycat. ■

Definition 27 A game $A = \langle M_A, \lambda_A, \mathcal{P}_A \rangle$ is a subgame of the game $A' = \langle M_A, \lambda_A, \mathcal{P}_{A'} \rangle$ if and only if $\mathcal{P}_A \subseteq \mathcal{P}_{A'}$.

Lemma 28 Given a strategy $\sigma : A \to B$ and games A' and B' such that A and B are subgames of A' and B' respectively it follows that $U\sigma : A' \to B'$ where U leaves the strategy itself unchanged but changes its domain and range accordingly.

Proof Proof is by inspection of the definitions. ■

Lemma 29 Given strategies $\sigma : A \to B$ and $\tau : B \to C$, games A', B' and C' such that A, B and C are subgames of A', B' and C' respectively it follows that $U(\sigma;\tau) = U\sigma;U\tau$ where U is defined as in lemma 28.

Proof The proof is by inspection of the definition of composition. ■

2.4 Categories of Games

2.4.1 The Underlying Category

Definition 30 (The Category G) We now define the category **G**: a category of games and strategies. Objects in **G** are games and a morphism from A to B is a strategy for the game $A \to B$. The composite of two morphisms is the composite of the strategies as given in definition 20 and the identity morphisms are the strategies of definition 25.

2.4.2 Prefix Closure

Definition 31 (Prefix Games) A game A is a prefix game if and only if

$$s \cdot m \in \mathcal{P}_A \Rightarrow s \in \mathcal{P}_A.$$

Lemma 32 Given prefix games A and B it follows that $A \to B$ is a prefix game.

Proof The follows directly from the definition of $A \to B$. ■

Definition 33 (Prefix Closure) We say that a strategy $\sigma : A$ is prefix closed if and only if

$$s \cdot m \cdot m' \in \sigma \Rightarrow s \in \sigma.$$

Lemma 34 Given a prefix game A we have $\mathbf{id}_A$ prefix closed.

Proof The follows directly from the definition of $\mathbf{id}_A$. ■

Lemma 35 Prefix closure is preserved under composition: given prefix closed strategies $\sigma : A \to B$ and $\tau : B \to C$, the strategy $\sigma;\tau$ is prefix closed.

Proof Let $s \cdot m \cdot m'$ be any sequence in $\sigma;\tau$ and let $u \cdot m \cdot u' \cdot m'$ be a witness to s. First we will show that $u \upharpoonright A, B \in \sigma$.

case: Suppose m is from game C. Lemma 15 implies that either $u \upharpoonright A, B$ is empty or it finishes with a player move with respect to $\lambda_{A \to B}$. Hence $u \upharpoonright A, B$ has even length and $u \upharpoonright A, B \in \sigma$ by the prefix closure of σ.

case: Suppose m is from game A. Lemma 15 implies that the ultimate move in $u \upharpoonright A, B$ must a player move with respect to $\lambda_{A \to B}$. Hence $u \upharpoonright A, B$ has even length and $u \upharpoonright A, B \in \sigma$ by prefix closure.

We can use similar reasoning to show that $u \upharpoonright B, C \in \tau$. By definition we now have $u \in \sigma \| \tau$ and thus $s \in \sigma;\tau$. ■

Definition 36 ($\mathbf{G}_p$) We define $\mathbf{G}_p$ to be the subcategory of **G** consisting of prefix games and prefix closed strategies.

2.4.3 Determinism

Definition 37 (Determinism) A strategy $\sigma : A$ is deterministic if and only if it is prefix closed and

$$\forall s \cdot m, s' \cdot m' \in \sigma.(s = s') \Rightarrow (s \cdot m = s' \cdot m').$$

Lemma 38 For any prefix game A we have $\mathbf{id}_A$ deterministic.

Proof Proof is by inspection of the definition of $\mathbf{id}_A$. ■

Definition 39 (Generalized Player and Opponent Moves) Given a sequence $u \in \mathbf{int}(A, B, C)$ we say that $m \in u$ is a generalized player move if and only if it occurs as a player move in either $u \upharpoonright A, B$ or $u \upharpoonright B, C$. Similarly, we say that $m \in u$ is a generalized opponent move if and only if it occurs as an opponent move in either $u \upharpoonright A, B$ or $u \upharpoonright B, C$.

Lemma 40 (Core Switching Condition) Given a sequence $u \in \mathbf{int}(A, B, C)$ and consecutive moves $m^-, m \in u$ then m is a player move in component X if and only if m^- is an opponent move in component X.

Proof By lemma 22 we know that m^- and m must be in the same component. We then simply observe that $u \upharpoonright X$ respects alternation. ■

Lemma 41 Given deterministic strategies $\sigma : A \to B$ and $\tau : B \to C$ and sequence $s \in \sigma;\tau$ the witness of s is unique.

Proof Suppose we have sequences u and u' that witness s. We will show that for any prefix $t \sqsubseteq u$ we also have $t \sqsubseteq u'$. We construct a proof by induction on the length of t.

Base Case If $t = \varepsilon$ then trivially $t \sqsubseteq u'$.

Inductive Step Suppose $t = t' \cdot m$.

case: If m is a generalised player move in component X then by lemma 40 we have m^- an opponent move in $u \upharpoonright X$. By inductive hypothesis we have $t' \sqsubseteq u'$ and inspection shows that we cannot have $t' = u'$ as a witness cannot end in a generalized opponent move. Let m' succeed m^- in u' and by lemma 40 m' must be a generalized player move in component X. As σ and τ are prefix closed strategies we have both $t' \cdot m \upharpoonright X$ and $t' \cdot m' \upharpoonright X$ in either σ or τ and hence by determinism we have $t' \cdot m = t' \cdot m'$.

case: Otherwise m is an opponent move from component X in $u \upharpoonright A, C$. By inductive hypothesis we have $t' \sqsubseteq u'$ and clearly $t' \neq u'$ as u' must witness s. Therefore let $t' \cdot m' \sqsubseteq u'$. Either, t' is empty or else by lemma 15 it ends with a player move in $u \upharpoonright A, C$ and by lemma 15 we have $m' \in u \upharpoonright A, C$. As both u and u' witness s it must therefore be the case that $t' \cdot m' = t' \cdot m$ and hence $t \cdot m \sqsubseteq u'$.

As $u \sqsubseteq u$ it follows that $u \sqsubseteq u'$ and similarly we can show $u' \sqsubseteq u$ and hence $u = u'$. ■

Lemma 42 Determinism is preserved under composition: given deterministic strategies $\sigma : A \to B$ and $\tau : B \to C$, it follows that $\sigma;\tau$ is deterministic.

Proof Suppose we have sequences $s \cdot m, s' \cdot m'$ in $\sigma;\tau$ such that $s = s'$. Let $u \cdot m$ witness $s \cdot m$

and let $u' \cdot m'$ witness $s' \cdot m'$. We construct a proof by induction on the length of a sequence $t \in \mathbf{int}(A, B, C)$ that $t \sqsubseteq u \cdot m$ implies $t \sqsubseteq u' \cdot m'$.

Base Case If $t = \varepsilon$ then we trivially have $t \sqsubseteq u' \cdot m'$.

Inductive Step Suppose $t = t' \cdot n$. By the inductive hypothesis we have $t' \sqsubseteq u' \cdot m'$.

case: If n is a generalised player move in component X then by lemma 40 it follows that t' ends in a generalized opponent move from component X. Therefore $t' \neq u' \cdot m'$ as no interaction sequence can end in a generalised opponent move. Let the move following the prefix t' in $u' \cdot m'$ be n'. n' must be a generalized player move in component X by lemma 40. As σ and τ are prefix closed we must have both $t' \cdot n \upharpoonright X$ and $t' \cdot n' \upharpoonright X$ in either σ or τ and by the determinism of these strategies we must have $t' \cdot n \upharpoonright X = t' \cdot n' \upharpoonright X$ and it is simple to check that $t' \cdot n = t' \cdot n'$.

case: If n is an opponent move in $u \upharpoonright A, C$ then by lemma 15 t' must be either empty or end in a player move from $u \upharpoonright A, C$. Clearly $t' \neq u' \cdot m'$ as $u \upharpoonright A, C = u' \upharpoonright A, C$. Let the move following the prefix t' in $u' \cdot m'$ be n'. n' must be an opponent move in $u' \upharpoonright A, C$ by lemma 15 and as $u \upharpoonright A, C = u' \upharpoonright A, C$ it must be that $t' \cdot n \upharpoonright A, C = t' \cdot n' \upharpoonright A, C$ and it is simple to check that $t' \cdot n = t' \cdot n'$.

As $u \cdot m \sqsubseteq u \cdot m$ we now have $u \cdot m \sqsubseteq u' \cdot m'$ and similarly we can show that $u' \cdot m' \sqsubseteq u \cdot m$ and thus $u \cdot m = u' \cdot m'$ and $s \cdot m = s' \cdot m'$ hence $\sigma;\tau$ is deterministic. ■

Definition 43 We define $\mathbf{G}_d$ to be the lluf (same objects fewer morphisms) subcategory of $\mathbf{G}_p$ consisting of prefix games and deterministic strategies.

2.4.4 Views

In contrast with the notions of view found in for example [28, 36] we define the player and opponent views of a sequence s to be not simply subsequences of s but directed acyclic graphs comprised of a subset of move occurrences from s.

Definition 44 (View) We define a view, v, of s to be a tuple $\langle M_v, \prec_v, \curvearrowleft_v \rangle$ where:

- $M_v \subseteq M_s$.
- $\prec_v \subseteq \lhd_s^*$, where $\lhd_s^*$ is the reflexive transitive closure of $\lhd_s$.
- $\curvearrowleft_v \subseteq M_v \times M_v$ is the justification relation $\curvearrowleft_s \cap (M_v \times M_v)$.

We write $m \in v$ as shorthand for $m \in M_v$.

Definition 45 (Equating Views) Given sequences $s, s' \in \mathcal{X}_A$ and views v of s and v' of s' we say that $v = v'$ if there exists some bijection $f : M_v \mapsto M_{v'}$ such that for all $m, m' \in M_v$:

Eq1 The bijection relates occurrences of the same move.

Eq2 $m \prec_v m' \Leftrightarrow fm \prec_{v'} fm'$. The bijection respects $\prec_v$ and $\prec_{v'}$.

Eq3 $m \curvearrowleft_v m' \Leftrightarrow fm \curvearrowleft_{v'} fm'$. The bijection respects $\curvearrowleft_v$ and $\curvearrowleft_{v'}$.

We say that a bijection obeying **Eq1-3** renders $v = v'$.

Player and opponent views will be familiar to those readers who have already encountered game semantics. Such a notion was first described by Hyland and Ong [28] and was originally defined for sequences where moves have at most one justifier, so that the player view of a sequence s, notated $\lceil s \rceil$, is a subsequence of s defined inductively as follows:

- $\lceil \varepsilon \rceil = \varepsilon$.
- $\lceil t \cdot m \rceil = \lceil t \rceil \cdot m$ m is a player move.
- $\lceil t \cdot m \rceil = m$ where m is an initial opponent move.
- $\lceil t \cdot j \overset{\frown}{\cdot t' \cdot} m \rceil = \lceil t \rceil \cdot j \overset{\frown}{\cdot} m$ where m is a non-initial opponent move.

As we can see, the predecessor of a given move m in $\lceil s \rceil$ is either its predecessor from s, if m is a player move, or else it is the justifier of m in s, if it exists.

Similarly the opponent view of a sequence s, notated $\lfloor s \rfloor$, is a subsequence of s defined inductively as follows:

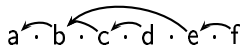
- $\lfloor \varepsilon \rfloor = \varepsilon$.
- $\lfloor t \cdot m \rfloor = \lfloor t \rfloor \cdot m$ where m is an opponent move.
- $\lfloor t \cdot m \rfloor = m$ where m is an initial player move.
- $\lfloor t \cdot j \overset{\frown}{\cdot t' \cdot} m \rfloor = \lfloor t \rfloor \cdot j \overset{\frown}{\cdot} m$ where m is a non-initial player move.

These definitions can be naturally extended to our new setting where a move may have more than one justifier, but the resultant structure of a player view is no longer merely a subsequence of our original but a DAG of move occurrences.

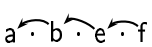
Definition 46 (Player View) Given a sequence $t \in \mathcal{X}_A$ we define the player view $\lceil t \rceil$ inductively as follows:

1. $\lceil \varepsilon \rceil = \langle \emptyset, \emptyset, \emptyset \rangle$.
2. $\lceil s \cdot m \cdot m' \rceil = \langle M_{\lceil s \cdot m \cdot m' \rceil}, \prec_{\lceil s \cdot m \cdot m' \rceil}, \curvearrowleft_{\lceil s \cdot m \cdot m' \rceil} \rangle$ when m' is a player move where:
   - $M_{\lceil s \cdot m \cdot m' \rceil} = M_{\lceil s \cdot m \rceil} \cup \{m'\}$.
   - $\prec_{\lceil s \cdot m \cdot m' \rceil} = \prec_{\lceil s \cdot m \rceil} \cup \{(m, m')\}$.
   - $\curvearrowleft_{\lceil s \cdot m \cdot m' \rceil} = \curvearrowleft_{s \cdot m \cdot m'} \cap (M_{\lceil s \cdot m \cdot m' \rceil} \times M_{\lceil s \cdot m \cdot m' \rceil})$.
3. $\lceil s \cdot m \rceil = \langle M_{s \cdot m}, \prec_{s \cdot m}, \curvearrowleft_{s \cdot m} \rangle$ when m is an opponent move where:
   - $M_{\lceil s \cdot m \rceil} = \bigcup \{ M_{\lceil s_{\leq j} \rceil} \mid j \curvearrowleft_s m \} \cup \{m\}$.
   - $\prec_{\lceil s \cdot m \rceil} = \bigcup \{ \prec_{\lceil s_{\leq j} \rceil} \cup \{(j, m)\} \mid j \curvearrowleft_{s \cdot m} m \}$.
   - $\curvearrowleft_{\lceil s \cdot m \rceil} = \curvearrowleft_{s \cdot m} \cap (M_{\lceil s \cdot m \rceil} \times M_{\lceil s \cdot m \rceil})$.

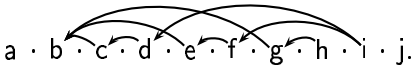
Hence, we see how this relates to the example at the beginning of this chapter when we considered the following sequence where each move has at most one justifier :



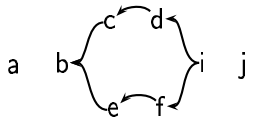
We can now inspect the definition of player view to see that it yields the following sequence which is identical to that view produced by Hyland and Ong's definition:



However, when we are given a sequence in which some moves have multiple justifiers



our player view will be the following DAG:



Definition 47 (Opponent View) Similarly, given a sequence $t \in \mathcal{X}_A$ we define the opponent view $\lfloor t \rfloor$ inductively as follows:

1. $\lfloor \varepsilon \rfloor = \langle \emptyset, \emptyset, \emptyset \rangle$.
2. $\lfloor s \cdot m \cdot m' \rfloor = \langle M_{\lfloor s \cdot m \cdot m' \rfloor}, \prec_{\lfloor s \cdot m \cdot m' \rfloor}, \curvearrowleft_{\lfloor} s \cdot m \cdot m' \rfloor \rangle$ when m' is an opponent move where:
   - $M_{\lceil s \cdot m \cdot m' \rceil} = M_{\lceil s \cdot m \rceil} \cup \{m'\}$.
   - $\prec_{\lceil s \cdot m \cdot m' \rceil} = \prec_{\lceil s \cdot m \rceil} \cup \{(m, m')\}$.
   - $\curvearrowleft_{\lceil s \cdot m \cdot m' \rceil} = \curvearrowleft_{s \cdot m \cdot m'} \cap (M_{\lceil s \cdot m \cdot m' \rceil} \times M_{\lceil s \cdot m \cdot m' \rceil})$.
3. $\lfloor s \cdot m \rfloor = \langle M_{\lfloor s \cdot m \rfloor}, \prec_{\lfloor s \cdot m \rfloor}, \curvearrowleft_{\lfloor s \cdot m \rfloor} \rangle$ when m is a player move where:
   - $M_{\lfloor s \cdot m \rfloor} = \bigcup \{ M_{\lfloor s_{\leq j} \rfloor} \mid j \curvearrowleft_s m \} \cup \{m\}$.
   - $\prec_{\lfloor s \cdot m \rfloor} = \bigcup \{ \prec_{\lfloor s_{\leq j} \rfloor} \cup \{(j, m)\} \mid j \curvearrowleft_{s \cdot m} m \}$.
   - $\curvearrowleft_{\lfloor s \cdot m \rfloor} = \curvearrowleft_{s \cdot m} \cap (M_{\lfloor s \cdot m \rfloor} \times M_{\lfloor s \cdot m \rfloor})$.

We now define some other views that we will only use in proofs in this chapter.

Definition 48 (Right View) Given a sequence $t \in \mathcal{X}_{A \to B}$ we define the right view $\lceil t \rceil^R$ inductively as follows:

1. $\lceil \varepsilon \rceil^R = \langle \emptyset, \emptyset, \emptyset \rangle$.
2. $\lceil s \cdot m \cdot m' \rceil^R = \langle M_{\lceil s \cdot m \cdot m' \rceil^R}, \prec_{\lceil s \cdot m \cdot m' \rceil^R}, \curvearrowleft_{\lceil s \cdot m \cdot m' \rceil^R} \rangle$ when m' is a player move or an opponent move from A where:
   - $M_{\lceil s \cdot m \cdot m' \rceil^R} = M_{\lceil s \cdot m \rceil^R} \cup \{m'\}$.
   - $\prec_{\lceil s \cdot m \cdot m' \rceil^R} = \prec_{\lceil s \cdot m \rceil^R} \cup \{(m, m')\}$.

- $\curvearrowleft_{\lceil s \cdot m \cdot m' \rceil^R} = \curvearrowleft_{s \cdot m \cdot m'} \cap (M_{\lceil s \cdot m \cdot m' \rceil^R} \times M_{\lceil s \cdot m \cdot m' \rceil^R})$.

3. $\lceil s \cdot m \rceil^R = \langle M_{s \cdot m}, \prec_{s \cdot m}, \curvearrowleft_{s \cdot m} \rangle$ when m is an opponent move from B where:

   - $M_{\lceil s \cdot m \rceil^R} = \bigcup \{ M_{\lceil s_{\leq j} \rceil^R} \mid j \curvearrowleft_s m \} \cup \{m\}$.
   - $\prec_{\lceil s \cdot m \rceil^R} = \bigcup \{ \prec_{\lceil s_{\leq j} \rceil^R} \cup \{(j, m)\} \mid j \curvearrowleft_{s \cdot m} m \}$.
   - $\curvearrowleft_{\lceil s \cdot m \rceil^R} = \curvearrowleft_{s \cdot m} \cap (M_{\lceil s \cdot m \rceil^R} \times M_{\lceil s \cdot m \rceil^R})$.

Definition 49 (Left View) Similarly, given a sequence $t \in \mathcal{X}_{A \to B}$ we define the right view $\lceil t \rceil^L$ inductively as follows:

1. $\lceil \varepsilon \rceil^L = \langle \emptyset, \emptyset, \emptyset \rangle$.
2. $\lceil s \cdot m \cdot m' \rceil^L = \langle M_{\lceil s \cdot m \cdot m' \rceil^L}, \prec_{\lceil s \cdot m \cdot m' \rceil^L}, \curvearrowleft_{\lceil s \cdot m \cdot m' \rceil^L} \rangle$ when m' is a player move or an opponent move from B where:

   - $M_{\lceil s \cdot m \cdot m' \rceil^L} = M_{\lceil s \cdot m \rceil^L} \cup \{m'\}$.
   - $\prec_{\lceil s \cdot m \cdot m' \rceil^L} = \prec_{\lceil s \cdot m \rceil^L} \cup \{(m, m')\}$.
   - $\curvearrowleft_{\lceil s \cdot m \cdot m' \rceil^L} = \curvearrowleft_{s \cdot m \cdot m'} \cap (M_{\lceil s \cdot m \cdot m' \rceil^L} \times M_{\lceil s \cdot m \cdot m' \rceil^L})$.

3. $\lceil s \cdot m \rceil^L = \langle M_{s \cdot m}, \prec_{s \cdot m}, \curvearrowleft_{s \cdot m} \rangle$ when m is an opponent move from A where:

   - $M_{\lceil s \cdot m \rceil^L} = \bigcup \{ M_{\lceil s_{\leq j} \rceil^L} \mid j \curvearrowleft_s m \} \cup \{m\}$.
   - $\prec_{\lceil s \cdot m \rceil^L} = \bigcup \{ \prec_{\lceil s_{\leq j} \rceil^L} \cup \{(j, m)\} \mid j \curvearrowleft_{s \cdot m} m \}$.
   - $\curvearrowleft_{\lceil s \cdot m \rceil^L} = \curvearrowleft_{s \cdot m} \cap (M_{\lceil s \cdot m \rceil^L} \times M_{\lceil s \cdot m \rceil^L})$.

The core view is the multiple justifier analog to that described by McCusker in [36].

Definition 50 (Core View) Given a sequence $t \in \mathcal{X}_{\mathbf{int}(A,B,C)}$ we define the core view $\overline{\lceil t \rceil}$ inductively as follows:

1. $\overline{\lceil \varepsilon \rceil} = \langle \emptyset, \emptyset, \emptyset \rangle$.
2. $\overline{\lceil s \cdot m \cdot m' \rceil} = \langle M_{\overline{\lceil s \cdot m \cdot m' \rceil}}, \prec_{\overline{\lceil s \cdot m \cdot m' \rceil}}, \curvearrowleft_{\overline{\lceil s \cdot m \cdot m' \rceil}} \rangle$ when m' is a player move or any move from B where:

   - $M_{\overline{\lceil s \cdot m \cdot m' \rceil}} = M_{\overline{\lceil s \cdot m \rceil}} \cup \{m'\}$.
   - $\prec_{\overline{\lceil s \cdot m \cdot m' \rceil}} = \prec_{\overline{\lceil s \cdot m \rceil}} \cup \{(m, m')\}$.
   - $\curvearrowleft_{\overline{\lceil s \cdot m \cdot m' \rceil}} = \curvearrowleft_{s \cdot m \cdot m'} \cap (M_{\overline{\lceil s \cdot m \cdot m' \rceil}} \times M_{\overline{\lceil s \cdot m \cdot m' \rceil}})$.

3. $\overline{\lceil s \cdot m \rceil} = \langle M_{s \cdot m}, \prec_{s \cdot m}, \curvearrowleft_{s \cdot m} \rangle$ when m is an opponent move from A or C where:

   - $M_{\overline{\lceil s \cdot m \rceil}} = \bigcup \{ M_{\overline{\lceil s_{\leq j} \rceil}} \mid j \curvearrowleft_s m \} \cup \{m\}$.
   - $\prec_{\overline{\lceil s \cdot m \rceil}} = \bigcup \{ \prec_{\overline{\lceil s_{\leq j} \rceil}} \cup \{(j, m)\} \mid j \curvearrowleft_{s \cdot m} m \}$.
   - $\curvearrowleft_{\overline{\lceil s \cdot m \rceil}} = \curvearrowleft_{s \cdot m} \cap (M_{\overline{\lceil s \cdot m \rceil}} \times M_{\overline{\lceil s \cdot m \rceil}})$.

Lemma 51 Given a sequence $s \in \mathcal{X}_A$ we have $\lceil s \rceil$ has a maximum element with respect to $\prec_{\lceil s \rceil}$ and $\lfloor s \rfloor$ has a maximum element with respect to $\prec_{\lfloor s \rfloor}$. Similarly, given a sequence $s' \in \mathcal{X}_{A \to B}$ we have $\lceil s' \rceil^L$ has a maximum element with respect to $\prec_{\lceil s' \rceil^L}$ and $\lceil s' \rceil^R$ has a maximum element with respect to $\prec_{\lceil s' \rceil^R}$. Similarly, given a sequence $s'' \in \mathbf{int}(A, B, C)$ we have $\overline{\lceil s'' \rceil}$ has a maximum element with respect to $\prec_{\overline{\lceil s'' \rceil}}$.

Proof Proof is by inspection of the definitions of the views in question. ■

Lemma 52 Suppose we are given sequences $s, s' \in A$, a move $m \in \lceil s \rceil$ and a bijection, f, that renders $\lceil s \rceil = \lceil s' \rceil$. Then $\lceil s_{\leq m} \rceil = \lceil s'_{\leq fm} \rceil$, rendered by a subset of f.

Similarly, given sequences $s, s' \in \mathbf{int}(A, B, C)$, a move $m \in \overline{\lceil s \rceil}$ and a bijection, f, that renders $\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}$ then $\overline{\lceil s_{\leq m} \rceil} = \overline{\lceil s'_{\leq fm} \rceil}$ rendered by a subset of f.

Proof Proof is by inspection of the definition of equality of views. ■

Lemma 53 Given a sequence $s \in A$ with moves $m, m', m'' \in s$ then view membership is transitive in the following sense:

$$m \in \lceil s_{\leq m'} \rceil \wedge m' \in \lceil s_{\leq m''} \rceil \Rightarrow m \in \lceil s_{\leq m''} \rceil$$

and

$$m \in \lfloor s_{\leq m'} \rfloor \wedge m' \in \lfloor s_{\leq m''} \rfloor \Rightarrow m \in \lfloor s_{\leq m''} \rfloor.$$

Similarly given a sequence $s \in A \to B$ and moves $m, m', m'' \in s$ we have

$$m \in \lceil s_{\leq m'} \rceil^R \wedge m' \in \lceil s_{\leq m''} \rceil^R \Rightarrow m \in \lceil s_{\leq m''} \rceil^R$$

and

$$m \in \lceil s_{\leq m'} \rceil^L \wedge m' \in \lceil s_{\leq m''} \rceil^L \Rightarrow m \in \lceil s_{\leq m''} \rceil^L.$$

Lastly given a sequence $u \in \mathbf{int}(A, B, C)$ with moves $m, m', m'' \in s$ we have

$$m \in \overline{\lceil s_{\leq m'} \rceil} \wedge m' \in \overline{\lceil s_{\leq m''} \rceil} \Rightarrow m \in \overline{\lceil s_{\leq m''} \rceil}.$$

Proof Proof is by inspection of the definitions of the views. ■

2.4.5 Visibility Conditions

Definition 54 (Visibility) We say that the *player visibility condition* is satisfied by a sequence $s \in \mathcal{X}_A$ if and only if, for every player move $m \in s$ we have $j \in \lceil s_{\leq} m \rceil$ for every move $j \in s$ such that $j \curvearrowleft m$.

Similarly, we say that the *opponent visibility condition* is satisfied by a sequence $s \in \mathcal{X}_A$ if and only if, for every opponent move $m \in s$ we have $j \in \lfloor s_{\leq} m \rfloor$ for every move $j \in s$ such that $j \curvearrowleft m$.

If a sequence $s \in \mathcal{X}_A$ satisfies both player and opponent visibility we say that it respects the *total visibility condition*.

Lemma 55 Given a sequence $s \cdot m \cdot t \cdot m' \cdot u \in \mathbf{int}(A, B, C)$ such that $t \cdot m'$ is made up from generalised player moves we have:

$$m \in \overline{\lceil s \cdot m \cdot t \cdot m' \cdot u \rceil} \Leftrightarrow m' \in \overline{\lceil s \cdot m \cdot t \cdot m' \cdot u \rceil}$$

Proof The proof is a simple induction on the length of t. ■

Lemma 56 Given sequences $s, s' \in \mathbf{int}(A, B, C)$ such that the bijection f renders $\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}$ and consecutive moves $m^-, m \in s$ with m a generalised player move we have

$$fm = (fm^-)^+.$$

Proof The proof follows from the fact that f obeys axiom **Eq2**. ■

Lemma 57 Given sequences $s, s' \in \mathbf{int}(A, B, C)$ such that bijection f renders $\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}$ and consecutive moves $m \cdot k_1 \cdots k_n$ in s with each k_i a generalised player move and consecutive moves $m' \cdot k'_1 \cdots k'_n$ in s', it follows that:

$$f m = m' \Rightarrow f k_i = k'_i$$

for all i such that $1 \leq i \leq n$.

Proof The proof is a simple induction on n using lemma 56 in the inductive step. ■

As a corollary to this lemma it follows that given moves $n, n' \in \overline{\lceil s \rceil}$ that are consecutive in $s \upharpoonright A, C$ with n' a player move we have fn and fn' consecutive in $s' \upharpoonright A, C$.

Lemma 58 Given a sequence $u \in \mathbf{int}(A, B, C)$ and moves $m, m' \in u \upharpoonright A, C$ we have

$$m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil \Leftrightarrow m \in \overline{\lceil u_{\leq m'} \rceil}.$$

Proof We carry out a proof by induction on the length of $u_{\leq m'}$.

Base Case If $u_{\leq m'} = \varepsilon$ then the lemma is trivially satisfied.

Inductive Step

case: Suppose m' is a player move and let m'^{-} be the immediate predecessor of m' in u and let m^* be the immediate predecessor of m' in $u \upharpoonright A, C$.

If $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ then either $m = m'$ and we trivially have $m \in \overline{\lceil u_{\leq m'} \rceil}$ or else $m \in \lceil u_{\leq m^*} \upharpoonright A, C \rceil$ and we can apply the inductive hypothesis to yield $m \in \overline{\lceil u_{\leq m^*} \rceil}$. All moves in u strictly between m^* and m' are from game B so we apply lemma 55 to yield $m \in \overline{\lceil u_{\leq m'} \rceil}$.

If $m \in \overline{\lceil u_{\leq m'} \rceil}$ then either $m = m'$ and we trivially have $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ or else $m \in \overline{\lceil u_{\leq m'^{-}} \rceil}$ and we apply lemma 55 to yield $m \in \overline{\lceil u_{\leq m^*} \rceil}$. We can now apply the inductive hypothesis to yield $m \in \lceil u_{\leq m^*} \upharpoonright A, C \rceil$ and hence $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ by the definition of $\lceil - \rceil$.

case: Now suppose m' is an opponent move.

If $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ then either $m = m'$ and we trivially have $m \in \overline{\lceil u_{\leq m'} \rceil}$ or else there exists some move $j \in u \upharpoonright A, C$ such that $j \curvearrowleft m'$ and $m \in \lceil u_{\leq j} \upharpoonright A, C \rceil$. We can apply the inductive hypothesis to yield $m \in \overline{\lceil u_{\leq j} \rceil}$ and hence $m \in \overline{\lceil u_{\leq m'} \rceil}$ by the definition of $\overline{\lceil - \rceil}$.

If $m \in \overline{\lceil u_{\leq m'} \rceil}$ then either $m = m'$ and we trivially have $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ or else there exists some move $j \in u \upharpoonright A, C$ such that $j \curvearrowleft m'$ and $m \in \overline{\lceil u_{\leq j} \rceil}$. By inductive hypothesis we have $m \in \lceil u_{\leq j} \upharpoonright A, C \rceil$ and hence $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ by the definition of $\lceil - \rceil$. ■

Lemma 59 Given a sequence $u \in \mathbf{int}(A, B, C)$, such that $u \upharpoonright A, B$ and $u \upharpoonright B, C$ respect player visibility, and moves $m, m' \in u \upharpoonright X$ we have

$$m \in \lceil u_{\leq m'} \upharpoonright X \rceil \Rightarrow m \in \overline{\lceil u_{\leq m'} \rceil}$$

where we let X range over components A, B and B, C.

Proof Let Y represent the component that is not X. We prove the lemma by induction on the

length of $u_{\leq m'}$.

Base Case If we have $u_{\leq m'} = \varepsilon$ then the lemma is trivially satisfied.

Inductive Step

case: Suppose m' is a player move in component X. Let m'^{-} be the predecessor of m' in u. By lemma 40 we know that $m'^{-} \in u \upharpoonright X$. If $m \in \lceil u_{\leq m'} \upharpoonright X \rceil$ then either $m = m'$ and we trivially have $m \in \overline{\lceil u_{\leq m'} \rceil}$ or else we have $m \in \lceil u_{\leq m'^{-}} \upharpoonright X \rceil$ and by inductive hypothesis $m \in \overline{\lceil u_{\leq m'^{-}} \rceil}$. The definition of $\overline{\lceil - \rceil}$ now yields $m \in \overline{\lceil u_{\leq m'} \rceil}$.

case: Now suppose m' is an opponent move in $u \upharpoonright A, C$. If $m \in \lceil u_{\leq m'} \upharpoonright X \rceil$ then either $m = m'$ and we trivially have $m \in \overline{\lceil u_{\leq m'} \rceil}$ or else we have some move $j \in u$ such that $j \curvearrowleft m'$ and $m \in \lceil u_{\leq j} \upharpoonright X \rceil$. By inductive hypothesis we have $m \in \overline{\lceil u_{\leq j} \rceil}$ and hence $m \in \overline{\lceil u_{\leq m'} \rceil}$ by the definition of $\overline{\lceil - \rceil}$.

case: Finally suppose m' is a move in $u \upharpoonright B$ and an opponent move in component X. If $m \in \lceil u_{\leq m'} \upharpoonright X \rceil$ then either $m = m'$ and we trivially have $m \in \overline{\lceil u_{\leq m'} \rceil}$ or else we have some move $j \in u$ such that $j \curvearrowleft m'$ and $m \in \lceil u_{\leq j} \upharpoonright X \rceil$. By inductive hypothesis we have $m \in \overline{\lceil u_{\leq j} \rceil}$ and by the player visibility of $u \upharpoonright Y$ we have $j \in \overline{\lceil u_{\leq m'} \rceil}$ and hence by lemma 53 we have $m \in \overline{\lceil u_{\leq m'} \rceil}$. ■

Lemma 60 Player visibility is preserved by composition: given strategies $\sigma : A \to B$ and $\tau : B \to C$ that respect player visibility it follows that $\sigma;\tau$ respects player visibility.

Proof Consider any sequence $s \in \sigma;\tau$ with moves $m, m' \in s \upharpoonright A, C$ such that $m \curvearrowleft m'$ with m' a player move in component X. Consider any witness $u \in \mathbf{int}(A, B, C)$ to s. By the player visibility of σ and τ we have $m \in \lceil u_{\leq m'} \upharpoonright X \rceil$ and hence $m \in \overline{\lceil u_{\leq m'} \rceil}$ by lemma 59 and by lemma 58 we have $m \in \lceil u_{\leq m'} \upharpoonright A, C \rceil$ ■

Definition 61 We can now define the subcategory $\mathbf{G}_v$ of $\mathbf{G}$ which has prefix games as objects and player visible strategies for the game $A \to B$ as morphisms from A to B.

Lemma 62 For any game A, $\mathbf{id}_A$ respects player visibility.

Proof We prove this by induction on the length of an arbitrary sequence $s \in \mathbf{id}_A$.

Base Case If $s = \varepsilon$ we have s respects player visibility.

Inductive Step Suppose $s = s' \cdot m \cdot m'$. By inductive hypothesis s' respects player visibility. We must check that for any move $j \in s' \cdot m \cdot m'$ such that $j \curvearrowleft m'$ we have $j \in \lceil s' \cdot m \cdot m' \rceil$. If $j = m$ then player visibility is clearly respected. Otherwise let j^+ immediately follow j in s and we have $j^+ \curvearrowleft m$ by inspection of the definition of $\mathbf{id}_A$ and hence $j \in \lceil s' \cdot m \cdot m' \rceil$ by inspection of the definition of $\lceil - \rceil$. ■

Lemma 63 Given a sequence $s \in \mathcal{X}_{A \to B}$ with moves $m, m' \in s$ such that $m' \in s \upharpoonright A$ (respectively $m' \in s \upharpoonright B$) we have

$$m \in \lfloor s_{\leq m'} \rfloor \Leftrightarrow m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor \text{ (respectively } m \in \lfloor s_{\leq m'} \upharpoonright B \rfloor)$$

Proof We only consider the case where $m \in s \upharpoonright A$ as the other case is similar.

We prove this by induction on the length of $s_{\leq m'}$.

Base Case If $s_{\leq m'} = \varepsilon$ then the lemma is trivially satisfied.

Inductive Step

case: Suppose m' is an opponent move. Let m'^{-} be the move immediately preceding m'. By lemma 15 we have $m'^{-} \in s \upharpoonright A$. If $m \in \lfloor s_{\leq m'} \rfloor$ then we either have $m = m'$ and $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ or else we have $m \in \lfloor s_{\leq m'^{-}} \rfloor$ and by inductive hypothesis $m \in \lfloor s_{\leq m'^{-}} \upharpoonright A \rfloor$ and hence $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ by the definition of $\lfloor - \rfloor$.

If $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ then we either have $m = m'$ and $m \in \lfloor s_{\leq m'} \rfloor$ or else we have $m \in \lfloor s_{\leq m'^{-}} \upharpoonright A \rfloor$ and by inductive hypothesis $m \in \lfloor s_{\leq m'^{-}} \rfloor$ and hence $m \in \lfloor s_{\leq m'} \rfloor$ by the definition of $\lfloor - \rfloor$.

case: Alternatively, suppose m' is a player move. If $m \in \lfloor s_{\leq m'} \rfloor$ then we either have $m = m'$ and $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ or else we have some move $j \in s$ such that $m \in \lfloor s_{\leq j} \rfloor$ and $j \curvearrowleft m'$. By inductive hypothesis we have $m \in \lfloor s_{\leq j} \upharpoonright A \rfloor$ and hence $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ by the definition of $\lfloor - \rfloor$.

If $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ then we either have $m = m'$ and $m \in \lfloor s_{\leq m'} \rfloor$ or else we have some move $j \in s$ such that If $m \in \lfloor s_{\leq j} \upharpoonright A \rfloor$ and $j \curvearrowleft m'$. By inductive hypothesis we have $m \in \lfloor s_{\leq j} \rfloor$ and $m \in \lfloor s_{\leq m'} \rfloor$ by the definition of $\lfloor - \rfloor$. ■

Lemma 64 Given a game A and a sequence $s \in \mathbf{id}_A$ such that $s \upharpoonright A$ respects total visibility it follows that s respects total visibility.

Proof Player visibility follows directly from lemma 62. Opponent visibility follows from lemma 63. ■

Lemma 65 Opponent visibility is preserved by composition: given strategies $\sigma : A \to B$ and $\tau : B \to C$ that respect opponent visibility it follows that $\sigma;\tau$ respects opponent visibility.

Proof Consider any sequence $s \in \sigma;\tau$ and moves $m, m' \in s$ such that $m \curvearrowleft m'$ with m' an opponent move. We will assume that $m, m' \in s \upharpoonright A$ as the proof when $m, m' \in s \upharpoonright C$ is similar. Let $u \in \mathbf{int}(A, B, C)$ be any witness to s. We know that σ obeys opponent visibility and therefore $m \in \lfloor u_{\leq m'} \upharpoonright A, B \rfloor$ and by lemma 63 we know that $m \in \lfloor u_{\leq m'} \upharpoonright A \rfloor$ and by the same lemma we also have $m \in \lfloor u_{\leq m'} \upharpoonright A, C \rfloor$. ■

Definition 66 (Total Visibility Game) A game A is a total visibility game if and only if each member of $\mathcal{P}_A$ respects total visibility.

Lemma 67 Given a total visibility game A we have $\mathbf{id}_A$ respects total visibility.

Proof We have noted in lemma 62 that $\mathbf{id}_A$ respects player visibility. Given any sequence $s \in \mathcal{P}_{A \to A}$ and moves $m, m' \in s \upharpoonright A$ such that $m \curvearrowleft m'$ and m' is an opponent move in s we have $m \in \lfloor s_{\leq m'} \upharpoonright A \rfloor$ by the total visibility of A and hence $m \in \lfloor s_{\leq m'} \rfloor$ by lemma 63. ■

Lemma 68 Given a sequence $s \in \mathcal{X}_{A \to B}$ and moves $m, m' \in s$ we have

V1 $m \in \lceil s_{\leq m'} \rceil \Rightarrow m \in \lceil s_{\leq m'} \rceil^{R}$.

V2 $(m \in \lfloor s_{\leq m'} \rfloor \wedge m' \in s \upharpoonright A) \Rightarrow m \in \lceil s_{\leq m'} \rceil^{R}$.

V3 $m \in \lceil s_{\leq m'} \rceil \Rightarrow m \in \lceil s_{\leq m'} \rceil^L$.

V4 $(m \in \lfloor s_{\leq m'} \rfloor \wedge m' \in s \upharpoonright B) \Rightarrow m \in \lceil s_{\leq m'} \rceil^L$.

Proof We prove only **V1** and **V2** as **V3** and **V4** are similar. Our proof is by induction on the length of $s_{\leq m'}$.

Base Case If $s_{\leq m'} = \varepsilon$ then the lemma is trivial.

Inductive Step

case: Suppose m' is a player move. First we prove **V1**. If $m \in \lceil s_{\leq m'} \rceil$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \rceil^R$ or else $m \in \lceil s_{\leq m'^-} \rceil$ in which case we apply inductive hypothesis **V1** to yield $m \in \lceil s_{\leq m'^-} \rceil^R$ and hence $m \in \lceil s_{\leq m'} \rceil^R$ by the definition of $\lceil - \rceil^R$. We now prove **V2**. If $m \in \lfloor s_{\leq m'} \rfloor$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \rceil^R$ or else there is some move j such that $j \curvearrowleft m'$ and $m \in \lfloor s_{\leq j} \rfloor$ in which case we apply inductive hypothesis **V2** to yield $m \in \lceil s_{\leq j} \rceil^R$. By total visibility we have $j \in \lceil s_{\leq m'} \rceil$ and, as $j \neq m'$ we have $j \in \lceil s_{\leq m'^-} \rceil$ and by inductive hypothesis **V1** we have $j \in \lceil s_{\leq m'^-} \rceil^R$ and hence $m \in \lceil s_{\leq m'^-} \rceil^R$ by lemma 53. We now have $m \in \lceil s_{\leq m'} \rceil^R$ by the definition of $\lceil - \rceil^R$.

case: Now suppose m' is an opponent move in $s \upharpoonright B$. First we prove **V1**. If $m \in \lceil s_{\leq m'} \rceil$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \rceil^R$ or else there is some move j such that $j \curvearrowleft m'$ and $m \in \lceil s_{\leq j} \rceil$ in which case we apply inductive hypothesis **V1** to yield $m \in \lceil s_{\leq j} \rceil^R$ and hence $m \in \lceil s_{\leq m'} \rceil^R$ by the definition of $\lceil - \rceil^R$. Statement **V2** is trivially satisfied.

case: Now suppose m' is an opponent move in $s \upharpoonright A$. First we prove **V1**. If $m \in \lceil s_{\leq m'} \rceil$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \rceil^R$ or else there is some move j such that $j \curvearrowleft m'$ and $m \in \lceil s_{\leq j} \rceil$ in which case we apply inductive hypothesis **V1** to yield $m \in \lceil s_{\leq j} \rceil^R$. By total visibility we have $j \in \lfloor s_{\leq m'} \rfloor$ and as $j \neq m'$ we have $j \in \lfloor s_{\leq m'^-} \rfloor$. Lemma 15 assures us that $m'^- \in s \upharpoonright A$ so by inductive hypothesis **V2** we have $j \in \lceil s_{\leq m'^-} \rceil^R$ and hence $j \in \lceil s_{\leq m'} \rceil^R$ by the definition of $\lceil - \rceil^R$. We can now apply lemma 53 to yield $m \in \lceil s_{\leq m'} \rceil^R$. Lastly we prove **V2**. If $m \in \lfloor s_{\leq m'} \rfloor$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \rceil^R$ or else we have $m \in \lfloor s_{\leq m'^-} \rfloor$. Lemma 15 assures us that $m'^- \in s \upharpoonright A$ so we can apply inductive hypothesis **V2** to yield $m \in \lceil s_{\leq m'^-} \rceil^R$ and hence $m \in \lceil s_{\leq m'} \rceil^R$ by the definition of $\lceil - \rceil^R$. ■

Lemma 69 Given a sequence $s \in \mathcal{X}_{A \to B}$ and moves $m, m' \in s$ we have

- $(m, m' \in s \upharpoonright A \wedge m \in \lceil s_{\leq m'} \rceil^L) \Rightarrow m \in \lceil s_{\leq m'} \upharpoonright A \rceil$.
- $(m, m' \in s \upharpoonright B \wedge m \in \lceil s_{\leq m'} \rceil^R) \Rightarrow m \in \lceil s_{\leq m'} \upharpoonright B \rceil$.

Proof We prove the first statement by induction on the length of $s_{\leq m'}$. Proof of the second statement is similar.

Base Case If $s_{\leq m'} = \varepsilon$ then the lemma is trivially satisfied.

Inductive Step

case: Suppose m' is an opponent move. If $m \in \lceil s_{\leq m'} \rceil^L$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \upharpoonright A \rceil$ or else there is some move $j \in s$ such that $j \curvearrowleft m'$ and $m \in \lceil s_{\leq j} \rceil^L$. We can apply the inductive hypothesis to yield $m \in \lceil s_{\leq j} \upharpoonright A \rceil$ and hence $m \in \lceil s_{\leq m'} \upharpoonright A \rceil$ by the definition of $\lceil - \rceil$.

case: Suppose m' is a player move. If $m \in \lceil s_{\leq m'} \rceil^L$ then either $m = m'$ and $m \in \lceil s_{\leq m'} \upharpoonright A \rceil$ otherwise $m \in \lceil s_{\leq m'^-} \rceil^L$ in which case let $m*$ be the predecessor of m' in $s \upharpoonright A$. Inspection of the definition of $\lceil - \rceil^L$ yields $m \in \lceil s_{\leq m*} \rceil^L$ and by inductive hypothesis $m \in \lceil s_{\leq m^*} \upharpoonright A \rceil$ and hence $m \in \lceil s_{\leq m'} \upharpoonright A \rceil$ by the definition of $\lceil - \rceil$. ■

Lemma 70 Given a sequence $s \in \mathcal{X}_{A \to B}$ such that $s \upharpoonright A$ and $s \upharpoonright B$ respect total visibility, it follows that s respects opponent visibility.

Proof Suppose we are given arbitrary moves $m, m' \in s \upharpoonright B$ such that $m \curvearrowleft m'$. If m' is an opponent move we have $m \in \lfloor s_{\leq m'} \upharpoonright B \rfloor$ by total visibility and $m \in \lfloor s_{\leq m'} \rfloor$ by lemma 63. The proof is similar if $m, m' \in s \upharpoonright A$. ■

Lemma 71 Given a sequence $s \in \mathcal{X}_{A \to B}$ that respects total visibility it follows that $s \upharpoonright A$ and $s \upharpoonright B$ respect total visibility.

Proof We only prove here that $s \upharpoonright B$ respects total visibility as the other proof is similar. Suppose we are given arbitrary moves $m, m' \in s \upharpoonright B$ such that $m \curvearrowleft m'$. If m' is an opponent move we have $m \in \lfloor s_{\leq m'} \rfloor$ by total visibility and $m \in \lfloor s_{\leq m'} \upharpoonright B \rfloor$ by lemma 63. Alternatively, if m' is a player move we have $m \in \lceil s_{\leq m'} \rceil$ by total visibility and $m \in \lceil s_{\leq m'} \rceil^R$ by lemma 68 and hence $m \in \lceil s_{\leq m'} \upharpoonright B \rceil$ by lemma 69. ■

Definition 72 We can now define a subcategory $\mathbf{G}_t$ of $\mathbf{G}$ comprising total visibility games as objects and strategies respecting total visibility as morphisms. By lemmas 65 and 60 we know that total visibility is preserved by composition and by lemma 67 we know that the copycat strategy for a total visibility game respects total visibility.

2.4.6 Innocence

Definition 73 (Innocence) A strategy $\sigma : A$ is innocent if and only if it is deterministic and for all sequences $s \cdot m \cdot n, s' \in \sigma$ such that there exists a sequence $s' \cdot m' \in \mathcal{P}_A$ for which $\lceil s \cdot m \rceil = \lceil s' \cdot m' \rceil$ then there must exist some extension $s' \cdot m' \cdot n' \in \sigma$ such that $\lceil s' \cdot m' \cdot n' \rceil = \lceil s \cdot m \cdot n \rceil$.

Lemma 74 For any game A the copycat strategy $\mathbf{id}_A$ is innocent.

Proof As noted in lemma 38 we have $\mathbf{id}_A$ deterministic, inspection of the definition of $\mathbf{id}_A$ shows that it is also innocent. ■

Lemma 75 Given games A, B and C, and sequences $s, s' \in \mathbf{int}(A, B, C)$, it follows that

$$(\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}) \Rightarrow (\lceil s \upharpoonright A, C \rceil = \lceil s' \upharpoonright A, C \rceil)$$

Proof Let f be a bijection that renders $\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}$. From lemma 58 we know that the moves in $\lceil s \upharpoonright A, C \rceil$ exactly coincide with the moves from $s \upharpoonright A, C$ in $\overline{\lceil s \rceil}$. Similarly, we know that the moves in $\lceil s' \upharpoonright A, C \rceil$ exactly coincide with the moves from $s' \upharpoonright A, C$ in $\overline{\lceil s' \rceil}$. We can therefore restrict f to the moves in $\lceil s \upharpoonright A, C \rceil$ and $\lceil s' \upharpoonright A, C \rceil$. This bijection, f', is between occurrences of the same move and respects justification as it is a subset of f. We therefore know that f' obeys **Eq1** and **Eq3** and

are only left to show that it obeys **Eq2**: we must show that if we have moves $m, n \in \lceil s \restriction A, C \rceil$ then we must also have

$$(m \prec_{\lceil s \restriction A,C \rceil} n) \Leftrightarrow (fm \prec_{\lceil s' \restriction A,C \rceil} fn).$$

case: If n is an opponent move then we have the following:

$$\begin{array}{lcll} m \prec_{\lceil s \restriction A,C \rceil} n & \Leftrightarrow & m \curvearrowleft n & \text{from the definition of } \lceil - \rceil. \\ & \Leftrightarrow & fm \curvearrowleft fn & \text{by } \mathbf{Eq3} \text{ applied to } f \\ & \Leftrightarrow & fm \prec_{\lceil s \restriction A,C \rceil} fn & \end{array}$$

case: If n is a player move and $m \prec_{\lceil s \restriction A,C \rceil} n$ then m immediately precedes n in $s \restriction A, B$ and all moves in s after m up to and including n are generalised player moves. We can therefore apply lemma 57 to show $fm \prec_{\lceil s' \restriction A,C \rceil} fn$. ■

Lemma 76 Given sequences $s, s' \in \mathbf{int}(A, B, C)$ that are either empty or end in a move from component X then if bijection f renders $\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}$ we have

$$m \in \lceil s \restriction X \rceil \Leftrightarrow f(m) \in \lceil s' \restriction X \rceil.$$

Proof We only need to prove

$$m \in \lceil s \restriction X \rceil \Rightarrow f(m) \in \lceil s' \restriction X \rceil$$

as f is bijective. The proof is by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is trivially satisfied.

Inductive Step Suppose $s = t \cdot n$.

case: If n is an opponent move in component X then by lemma 51 and inspection of the definition of equality of views reveals that s' must be of the form $t' \cdot fn$. Now consider any move $m \in \lceil t \cdot n \restriction X \rceil$. It must be that either $m = n$ and hence $f(m) \in \lceil s' \restriction X \rceil$ or else there is some move j such that $j \curvearrowleft n$ and $m \in \lceil t_{\leq j} \restriction X \rceil$. By lemma 52 we have $\overline{\lceil t_{\leq j} \rceil} = \overline{\lceil t'_{\leq fj} \rceil}$ and by inductive hypothesis $f(m) \in \lceil s'_{\leq fj} \restriction X \rceil$ and hence $f(m) \in \lceil s' \restriction X \rceil$ by the definition of player view.

case: If n is a player move then by lemma 51 and inspection of the definition of equality of views reveals that s' must be of the form $t' \cdot fn$. By lemma 52 we have $\overline{\lceil t \rceil} = \overline{\lceil t' \rceil}$ rendered by a subset of f. Now consider any move $m \in \lceil t \cdot n \restriction X \rceil$. It must be that either $m = n$ and hence $f(m) \in \lceil s' \restriction X \rceil$ or else $m \in \lceil t \restriction X \rceil$ and $f(m) \in \lceil t' \restriction X \rceil$ by inductive hypothesis hence $f(m) \in \lceil s' \restriction X \rceil$ by the definition of player view. ■

Lemma 77 Given sequences $s, s' \in \mathbf{int}(A, B, C)$. Let X be either component A, B or component B, C and let s be either empty or ending in a move from component X. Then we have

$$(\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}) \Rightarrow (\lceil s \restriction X \rceil = \lceil s' \restriction X \rceil).$$

Proof Let f be a bijection that renders $\overline{\lceil s \rceil} = \overline{\lceil s' \rceil}$. Lemma 76 assures us that the restriction of of

f to $M_{\lceil s \restriction X \rceil}$ has range $M_{\lceil s' \restriction X \rceil}$. As this relation is a subset of f, it satisfies **Eq1** and **Eq3**. We are left to check **Eq2**: that the bijection respects $\prec_{\lceil s \restriction X \rceil}$ and $\prec_{\lceil s' \restriction X \rceil}$. Let $m, m' \in \lceil s \restriction X \rceil$.

case: If m' is a player move in component X we have

$$\begin{array}{lll} m \prec_{\lceil s \restriction X \rceil} m' & \Leftrightarrow \quad m = m'^{-} & \text{by inspection of the definitions of } \lceil - \rceil \\ & \Leftrightarrow \quad m \prec_{\overline{\lceil s \rceil}} m' & \text{by inspection of the definitions of } \overline{\lceil - \rceil} \\ & \Leftrightarrow \quad fm \prec_{\overline{\lceil s' \rceil}} fm' & \text{by } \mathbf{Eq2} \text{ applied to } \overline{\lceil s \rceil} = \overline{\lceil s' \rceil} \\ & \Leftrightarrow \quad fm = (fm')^{-} & \text{by inspection of the definitions of } \overline{\lceil - \rceil} \\ & \Leftrightarrow \quad fm \prec_{\lceil s \restriction X \rceil} fm' & \text{by inspection of the definitions of } \lceil - \rceil \end{array}$$

case: If m' is an opponent move in component X we have

$$\begin{array}{lll} m \prec_{\lceil s \restriction X \rceil} m' & \Leftrightarrow \quad m \curvearrowleft m' & \text{by inspection of the definition of } \lceil - \rceil \\ & \Leftrightarrow \quad fm \prec_{\overline{\lceil s' \rceil}} fm' & \text{by } \mathbf{Eq3} \text{ applied to } \overline{\lceil s \rceil} = \overline{\lceil s' \rceil} \\ & \Leftrightarrow \quad fm \curvearrowleft fm' & \text{by inspection of the definition of } \overline{\lceil - \rceil} \\ & \Leftrightarrow \quad fm \prec_{\lceil s \restriction X \rceil} fm' & \end{array}$$

One intuition we might have about games of this kind is that they might be used to model languages in which certain subcomputations behave independently of each other. Indeed the ideas presented in this chapter bear a close resemblance to those that we will introduce when we define the category of arenas that we use to model interference controlled languages in chapter 6.

■

Lemma 78 Given innocent strategies $\sigma : A \to B$ and $\tau : B \to C$, sequences $u, u' \in \sigma \| \tau$ and consecutive moves m, n in u and m', n' in u' such that m and m' are generalised opponent moves we have

$$\overline{\lceil u_{\leq m} \rceil} = \overline{\lceil u'_{\leq m'} \rceil} \Rightarrow \overline{\lceil u_{\leq n} \rceil} = \overline{\lceil u'_{\leq n'} \rceil}$$

Proof Let m be an opponent move in component X. We know that $\overline{\lceil u_{\leq m} \rceil} = \overline{\lceil u'_{\leq m'} \rceil}$ implies $\overline{\lceil u_{\leq m} \restriction X \rceil} = \overline{\lceil u'_{\leq m'} \restriction X \rceil}$ by lemma 77. Note that both n and n' are player moves in component X therefore, by the innocence of σ and τ, we must have $\overline{\lceil u_{\leq n} \restriction X \rceil} = \overline{\lceil u'_{\leq n'} \restriction X \rceil}$. We can therefore extend the bijection that rendered $\overline{\lceil u_{\leq m} \rceil} = \overline{\lceil u'_{\leq m'} \rceil}$ with (n, n') and it is straightforward to show that the resultant bijection renders $\overline{\lceil u_{\leq n} \rceil} = \overline{\lceil u'_{\leq n'} \rceil}$. ■

Lemma 79 Given innocent strategies $\sigma : A \to B$ and $\tau : B \to C$ a sequence $s \cdot m \cdot t \cdot n \in \sigma \| \tau$ where m and n are from A or C and t is a sequence of moves from B and a sequence $s' \cdot m' \in \mathbf{int}(A, B, C)$ such that $\overline{\lceil s' \cdot m' \rceil} = \overline{\lceil s \cdot m \rceil}$ and $s' \in \sigma \| \tau$ then there exists a sequence $s' \cdot m' \cdot t' \cdot n' \in \sigma \| \tau$ such that

$$\overline{\lceil s' \cdot m' \cdot t' \cdot n' \rceil} = \overline{\lceil s \cdot m \cdot t \cdot n \rceil}.$$

Proof Suppose that we take any non-empty prefix $s \cdot u \cdot l \sqsubseteq s \cdot m \cdot t$. If l is a generalised player move, then let component X be the component where l is a player move, otherwise let component X be the component that does not contain l. Let component Y be the component that is not X. Let α and β range over σ and τ with α corresponding to component X and β to component Y.

We prove the statement that there exists a sequence $s' \cdot u' \cdot l' \in \mathcal{X}_{\mathbf{int}(A,B,C)}$ such that we have the following:

U1 $\overline{\lceil s' \cdot u' \cdot l' \rceil} = \overline{\lceil s' \cdot u \cdot l \rceil}$.

U2 $s' \cdot u' \upharpoonright Y \in \beta$.

U3 $s' \cdot u' \cdot l' \upharpoonright X \in \alpha$.

Our proof is by induction on the length of the sequence u.

Base Case If $u = \varepsilon$ then $l = m$ and we set $s' \cdot u' \cdot l' = s' \cdot m'$.

First we see that **U1** is simply a restatement of our original assumption that $\overline{\lceil s' \cdot m' \rceil} = \overline{\lceil s \cdot m \rceil}$. To prove **U2** we note that as $s' \in \sigma \| \tau$ by our original assumptions we have $s' \cdot u' \upharpoonright Y \in \beta$. Similarly, condition **U3** also follows from the fact that $s' \in \sigma \| \tau$ and therefore $s' \cdot u' \cdot l' \upharpoonright X \in \alpha$ as l' is not in component X.

Inductive Step If u is not empty then l is a generalised player move in component X. By inductive hypothesis **U1** we have some sequence $s' \cdot u'$ such that $\overline{\lceil s' \cdot u' \rceil} = \overline{\lceil s \cdot u \rceil}$. It is simple then to extend $s' \cdot u'$ with a matching generalised player move to yield a sequence $s' \cdot u' \cdot l' \in \mathbf{int}(A, B, C)$ with appropriate justifiers attached such that $\overline{\lceil s' \cdot u' \cdot l' \rceil} = \overline{\lceil s \cdot u \cdot l \rceil}$.

To prove **U2** we see that we have $s' \cdot u' \upharpoonright Y \in \beta$ by inductive hypothesis **U3**.

We now prove **U3**. As noted, inductive hypothesis **U1** implies $\overline{\lceil s' \cdot u' \rceil} = \overline{\lceil s \cdot u \rceil}$ hence by lemma 77 we have $\lceil s' \cdot u' \upharpoonright X \rceil = \lceil s \cdot u \upharpoonright X \rceil$ and u' must end in an opponent move in X. From the innocence of β and the fact that we have $s \cdot u \cdot l \upharpoonright X \in \alpha$, we must have sequence $s' \cdot u' \cdot l' \upharpoonright X \in \alpha$.

To prove the lemma we consider the special case when $s \cdot u \cdot l = s \cdot m \cdot t \cdot n$ and see that there exists a $s' \cdot m' \cdot t' \cdot n' \in \mathbf{int}(A, B, C)$ such that we have the following:

U1 $\overline{\lceil s' \cdot m \cdot t' \cdot n' \rceil} = \overline{\lceil s \cdot m \cdot t \cdot n \rceil}$.

U2 $s' \cdot m' \cdot t' \upharpoonright Y \in \beta$.

U3 $s' \cdot m' \cdot t' \cdot n' \upharpoonright X \in \alpha$.

As n' is not in component Y we also have $s' \cdot m' \cdot t' \cdot n' \upharpoonright Y \in \beta$. We therefore have a sequence $s' \cdot m' \cdot t' \cdot n' \in \sigma \| \tau$ such that $\overline{\lceil s' \cdot m \cdot t' \cdot n' \rceil} = \overline{\lceil s \cdot m \cdot t \cdot n \rceil}$. ■

Lemma 80 Given innocent strategies $\sigma : A \to B$ and $\tau : B \to C$ and sequences $s \cdot m, s' \cdot m' \in \mathbf{int}(A, B, C)$ such that $s, s' \in \sigma \| \tau$ we have:

$$(\lceil s \upharpoonright A, C \rceil = \lceil s' \upharpoonright A, C \rceil) \Rightarrow (\overline{\lceil s \rceil} = \overline{\lceil s' \rceil})$$

and also

$$(\lceil s \cdot m \upharpoonright A, C \rceil = \lceil s' \cdot m' \upharpoonright A, C \rceil) \Rightarrow (\overline{\lceil s \cdot m \rceil} = \overline{\lceil s' \cdot m' \rceil}).$$

Proof We construct a proof by induction on the length of $s \cdot m$. We strengthen our inductive

hypotheses so that $(\overline{\lceil s \rceil} = \overline{\lceil s' \rceil})$ is rendered by a superset of the bijection that renders $(\lceil s \upharpoonright A, C \rceil = \lceil s' \upharpoonright A, C \rceil)$ and similarly $(\overline{\lceil s \cdot m \rceil} = \overline{\lceil s' \cdot m' \rceil})$ is rendered by a superset of the bijection that renders $(\lceil s \cdot m \upharpoonright A, C \rceil = \lceil s' \cdot m' \upharpoonright A, C \rceil)$.

Base Case It cannot be the case that $s \cdot m = \varepsilon$ so the lemma is trivially satisfied.

Inductive Step Suppose $(\lceil s \upharpoonright A, C \rceil = \lceil s' \upharpoonright A, C \rceil)$ is rendered by bijection f. Let k be the last opponent move in s from game A or C. Clearly fk is the last opponent move in s' from game A or C. From the definition of $\lceil - \rceil$ we have $(\lceil s_{\leq k} \upharpoonright A, C \rceil = \lceil s'_{\leq fk} \upharpoonright A, C \rceil)$ and by inductive hypothesis we have $(\overline{\lceil s_{\leq k} \rceil} = \overline{\lceil s'_{\leq fk} \rceil})$ and hence $(\overline{\lceil s \rceil} = \overline{\lceil s' \rceil})$ by lemma 79.

Suppose $(\lceil s \cdot m \upharpoonright A, C \rceil = \lceil s' \cdot m' \upharpoonright A, C \rceil)$ is rendered by bijection f. For all moves j such that $j \curvearrowleft m$ we have $(\lceil s \cdot m_{\leq j} \upharpoonright A, C \rceil = \lceil s' \cdot m'_{\leq fj} \upharpoonright A, C \rceil)$ by lemma 52. We can apply the inductive hypothesis to yield $(\overline{\lceil s \cdot m_{\leq j} \rceil} = \overline{\lceil s' \cdot m'_{\leq fj} \rceil})$. It is now trivial to check that $(\overline{\lceil s \cdot m \rceil} = \overline{\lceil s' \cdot m' \rceil})$.

■

Lemma 81 Innocence is preserved by composition: given innocent strategies $\sigma : A \to B$ and $\tau : B \to C$ then the composite $\sigma;\tau : A \to C$ is also innocent.

Proof We know that $\sigma;\tau$ is deterministic and respects player visibility by lemmas 42 and 60.

Finally we must show for all sequences $s \cdot m \cdot n, s' \in \sigma;\tau$, given any $s' \cdot m' \in \mathcal{P}_{A \to C}$ such that $\lceil s \cdot m \rceil = \lceil s' \cdot m' \rceil$ we must also have a sequence $s' \cdot m' \cdot n' \in \sigma;\tau$.

Let $u \cdot m \cdot t \cdot n$ be the witness to $s \cdot m \cdot n$ and u' be any witness to s'. By lemma 80 we have $\overline{\lceil u \cdot m \rceil} = \overline{\lceil u' \cdot m \rceil}$ and by lemma 79 we must have a sequence $u' \cdot m' \cdot t' \cdot n' \in \sigma \| \tau$ such that $\overline{\lceil u \cdot m \cdot t \cdot n \rceil} = \overline{\lceil u' \cdot m \cdot t' \cdot n' \rceil}'$ and hence some sequence $s' \cdot m' \cdot n' \in \sigma;\tau$ such that $s \cdot m \cdot n = s' \cdot m' \cdot n'$ by lemma 75.

■

Chapter 3

QA Arenas

3.1 Introduction

In this chapter we introduce the familiar notion of arenas. Unfortunately one can find many subtle variations in definition and presentation of game semantics in the literature. We present definitions that we hope will be readily recognizable and we indicate, where necessary, changes that have been made to familiar definitions. The most important distinction between games and arenas in the literature is that an arena does not have a set of valid plays directly associated with it. The arenas presented in this chapter have extra structure in the form of an enabling relation $\vdash$, which specifies which move may justify which, and extra labelling on the moves themselves: moves are either labelled as questions are answers. It is important also to note that the strategies for the arenas in this chapter are comprised of sequences in which any move occurrence may have at most one justifier.

Placing constraints on strategies is central to game semantics. It is often a complicated task to demonstrate that a category is formed when these constraints are in place: the proof that constrained strategies yield a similarly constrained composite is often arduous and it is sometimes hard to show that a suitably constrained strategy is available to act as an identity. We have already encountered this headache in Chapter 2, when we showed that visibility, innocence, determinism and so on were preserved under composition and were present in certain copycat strategies. Proofs of analogous results can be used to establish categories of arenas and strategies under similar constraints but it would be desirable if we did not have to repeat such work too often. To this end we introduce the novel concepts of *referees* and *umpires* and use these, and the results of Chapter 2, to identify certain properties of strategies that are preserved under composition and properties that are possessed by strategies that act as the unit under composition. We then show how the familiar definitions of visibility, innocence, bracketing, rigidity and thread-independence may be reformulated in this framework and this in turn yields straightforward proofs that these constraints are indeed preserved under composition and are possessed by strategies that can act as identities. We will use these methods again in Chapter 6 when we introduce our games model of SCI.

3.2 Arenas, Strategies and Composition

3.2.1 Arenas

Definition 82 (Question Answer Arenas) A question-answer arena (from now on a QA arena or simply an arena) A is defined by a tuple $\langle \mathsf{M_A}, \vdash_{\mathsf{A}}, \lambda_{\mathsf{A}} \rangle$ where

- $\mathsf{M_A}$ is a set of moves.
- $\lambda_{\mathsf{A}} : \mathsf{M_A} \to \{O,P\} \times \{Q,A\}$ is a labelling function that indicates whether a member of $\mathsf{M_A}$ is an opponent move or a player move and whether it is a *question* or an *answer*. We sometimes write λ^{QA} if we are only concerned whether a given move is a question or an answer. We define λ^{OP} similarly.
- $\vdash_{\mathsf{A}} \subseteq (\mathsf{M_A} + \{\star\}) \times \mathsf{M_A}$, where $\star$ is just some dummy symbol, is known as the enabling relation and it must satisfy the following:
  1. $\star \vdash_{\mathsf{A}} \mathsf{m} \Rightarrow \lambda^{QA}_{\mathsf{A}}\mathsf{m} = Q \wedge (\mathsf{m}' \vdash \mathsf{m} \Leftrightarrow \mathsf{m}' = \star)$. We call such a move m an *initial move*.
  2. If $\mathsf{m} \vdash_{\mathsf{A}} \mathsf{m}'$ and $\mathsf{m} \neq \star$ then
  $$\lambda^{OP}_{\mathsf{A}}\mathsf{m} \neq \lambda^{OP}_{\mathsf{A}}\mathsf{m}'$$
  and
  $$\lambda^{QA}_{\mathsf{A}}\mathsf{m}' = A \Rightarrow \lambda^{QA}_{\mathsf{A}}\mathsf{m} = Q.$$

We must note here that we are going to allow initial player moves. This is in contrast with many of the models found in the literature and will be purely cosmetic until we define our games model for interference controlled languages in chapter 6. This restriction is also relaxed, for entirely different reasons in [32].

Definition 83 (Negative Arenas) Following the convention of [32] we say that a negative QA arena is one for which all initial moves are opponent moves.

Definition 84 Given a labelling function λ_{A} we define $\overline{\lambda_{\mathsf{A}}}$ to be the labelling function such that for all $\mathsf{m} \in \mathsf{M_A}$ we have

- $\overline{\lambda_{\mathsf{A}}}^{QA}\mathsf{m} = \lambda^{QA}_{\mathsf{A}}\mathsf{m}$.
- $\overline{\lambda_{\mathsf{A}}}^{OP}\mathsf{m} \neq \lambda^{OP}_{\mathsf{A}}\mathsf{m}$.

Definition 85 (Justified Sequences) A justified sequence for an arena A is a sequence s of moves from $\mathsf{M_A}$ such that each non-initial move m has a pointer to *exactly one* previous move $\mathsf{j} \in \mathsf{s}$ such that $\mathsf{j} \vdash \mathsf{m}$. We call such a move the *justifier* of m and we write $\mathsf{j} \curvearrowleft \mathsf{m}$ to indicate this. If we annotate a sequence as
$$\mathsf{s} \cdot \mathsf{j} \overset{\frown}{\cdot \mathsf{s}' \cdot} \mathsf{m} \cdot \mathsf{s}''$$
it means that $\mathsf{j} \curvearrowleft \mathsf{m}$. We use the term hereditary justification to mean the reflexive, transitive closure of the justification relation. We write $\mathcal{J}_{\mathsf{A}}$ for the set of justified sequences for arena A in which opponent and player play alternate moves.

Definition 86 (Arrow Arena) Given negative arenas A and B we define arena

$$\mathsf{A} \to \mathsf{B} = \langle \mathsf{M}_{\mathsf{A}\to\mathsf{B}}, \vdash_{\mathsf{A}\to\mathsf{B}}, \lambda_{\mathsf{A}\to\mathsf{B}} \rangle$$

as follows:

$$\begin{aligned} \mathsf{M}_{\mathsf{A}\to\mathsf{B}} &= \mathsf{M}_{\mathsf{A}} + \mathsf{M}_{\mathsf{B}} \\ \vdash_{\mathsf{A}\to\mathsf{B}} &= \vdash_{\mathsf{A}} + \vdash_{\mathsf{B}} \\ \lambda_{\mathsf{A}\to\mathsf{B}} &= [\overline{\lambda_{\mathsf{A}}}, \lambda_{\mathsf{B}}] \end{aligned}$$

Note that this definition differs from previous definitions of the arrow arena. The approach taken in, for example, [28, 39, 4, 5, 6, 1, 9, 31] when forming the analog of our arrow arena $\mathsf{A} \to \mathsf{B}$ is to have initial moves from A enabled by initial moves from B, and thus $\mathsf{A} \to \mathsf{B}$ is also a negative arena. Given a sequence $\mathsf{s} \in \mathcal{I}_{\mathsf{A}\to\mathsf{B}}$ we write $\mathsf{s} \upharpoonright \mathsf{A}$ for that subsequence of moves in s comprising exactly those moves from M_{A}; we define $\mathsf{s} \upharpoonright \mathsf{B}$ similarly.

Definition 87 We define the empty arena:

$$\mathbf{null} = \langle \emptyset, \emptyset, \emptyset \rangle$$

Lemma 88 Given a negative arena A it follows that $\mathsf{A} = \mathbf{null} \to \mathsf{A}$. This follows directly from the definitions; the tagging of moves to form disjoint unions is not important here.

3.2.2 Strategies

Definition 89 (The Switching Condition) Given negative QA arenas A and B we say that a sequence $\mathsf{s} \in \mathcal{I}_{\mathsf{A}\to\mathsf{B}}$ respects the switching condition if and only if:

- The sequence s does not commence with a player move.
- For any consecutive moves $\mathsf{m}, \mathsf{m}^+ \in \mathsf{s}$ such that either $\mathsf{m} \in \mathsf{s} \upharpoonright \mathsf{A}$ and $\mathsf{m}^+ \in \mathsf{s} \upharpoonright \mathsf{B}$, or $\mathsf{m} \in \mathsf{s} \upharpoonright \mathsf{B}$ and $\mathsf{m}^+ \in \mathsf{s} \upharpoonright \mathsf{A}$, it follows that m^+ is a player move. We write $\mathcal{S}_{\mathsf{A}\to\mathsf{B}}$ for the set of sequences in $\mathcal{I}_{\mathsf{A}\to\mathsf{B}}$ that obey the switching condition.

Lemma 90 Given an arena $\mathsf{A} \to \mathsf{B}$ and a sequence $\mathsf{s} \in \mathcal{I}_{\mathsf{A}\to\mathsf{B}}$ that does not start with a player move it follows that:

$$\mathsf{s} \in \mathcal{S}_{\mathsf{A}\to\mathsf{B}} \Leftrightarrow (\mathsf{s} \upharpoonright \mathsf{A} \in \mathcal{S}_{\mathsf{A}} \wedge \mathsf{s} \upharpoonright \mathsf{B} \in \mathcal{S}_{\mathsf{B}}).$$

Proof First we assume $\mathsf{s} \in \mathcal{S}_{\mathsf{A}\to\mathsf{B}}$ and prove that $\mathsf{s} \upharpoonright \mathsf{A} \in \mathcal{S}_{\mathsf{A}}$ and $\mathsf{s} \upharpoonright \mathsf{B} \in \mathcal{S}_{\mathsf{B}}$. Note that for any moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that $\mathsf{m} \curvearrowleft \mathsf{m}'$ we have either $\mathsf{m}, \mathsf{m}' \in \mathsf{s} \upharpoonright \mathsf{A} \in \mathcal{S}_{\mathsf{A}}$ or $\mathsf{m}, \mathsf{m}' \in \mathsf{s} \upharpoonright \mathsf{B} \in \mathcal{S}_{\mathsf{B}}$ and we are simply left to check that $\mathsf{s} \upharpoonright \mathsf{A}$ and $\mathsf{s} \upharpoonright \mathsf{B}$ obey alternation by induction on the length of s.

Base Case If $\mathsf{s} = \varepsilon$ then $\mathsf{s} \upharpoonright \mathsf{A}$ and $\mathsf{s} \upharpoonright \mathsf{B}$ trivially alternate.

Inductive Step Suppose $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$. Let $\mathsf{m} \in \mathsf{s} \upharpoonright \mathsf{A}$. If s' is empty or ends in a move from A then the proof follows from the inductive hypothesis and the fact that s alternates. Now suppose s' ends in a move from B. By the switching condition we know that m is a player move and by the switching

condition we also know that the last move in $s' \upharpoonright A$ must be an opponent move and as $s' \upharpoonright A$ and $s' \upharpoonright B$ alternate by inductive hypothesis it follows that $s \upharpoonright A$ and $s \upharpoonright B$ obey alternation.

The proof is similar when $m \in s \upharpoonright B$.

We now assume that $s \upharpoonright A \in \mathcal{S}_A$ and $s \upharpoonright A \in \mathcal{S}_B$ and prove that $s \in \mathcal{S}_{A \to B}$ by induction on the length of s.

Base Case If $s = \varepsilon$ then we trivially have $s \in \mathcal{S}_{A \to B}$.

Inductive Step Suppose $s = s' \cdot m$. Let $m \in s \upharpoonright A$. The sequence s' obeys the switching condition by inductive hypothesis. If s' is empty or ends in a move from A then it follows trivially that s also obeys the switching condition. Now suppose s' ends in a move from B. If there is no move in s' from A then it must be that m is a player move, as there are no initial opponent moves is A, and hence s obeys the switching condition. Otherwise it must be that the last move in $s' \upharpoonright A$ is an opponent move by the inductive hypothesis and as $s' \upharpoonright A$ alternates it follows that m is a player move and hence $s \in \mathcal{S}_{A \to B}$.

The proof is similar when $m \in s \upharpoonright B$. ■

Definition 91 A strategy, σ, for the arena $A \to B$ is a subset of $\mathcal{S}_{A \to B}$. We write $\sigma : A \to B$ to denote this.

Definition 92 Given a negative arena $A = \langle M_A, \vdash_A, \lambda_A \rangle$ we define the game $U_{\mathcal{S}} A = \langle M_A, \lambda_A^{OP}, \mathcal{S}_A \rangle$.

We overload the notation so that for any sequence $s \in \mathcal{S}_A$ we have $U_{\mathcal{S}} s = s$ and for any set $S \subseteq \mathcal{S}_A$ we have $U_{\mathcal{S}} S = S$. The reasons for doing this will become clear presently.

Lemma 93 Given a sequence $s \in \mathcal{J}_{A \to B}$ it follows that:

$$s \in \mathcal{S}_{A \to B} \Leftrightarrow s \in \mathcal{P}_{U_{\mathcal{S}} A \to U_{\mathcal{S}} B}.$$

Proof By lemma 90 we have $s \upharpoonright A \in \mathcal{S}_A$ and $s \upharpoonright B \in \mathcal{S}_B$ and s cannot start with a move from A as there are no initial opponent moves in A. ■

As a corollary for any set $\sigma \subseteq \mathcal{J}_{A \to B}$ it follows that:

$$\sigma : A \to B \Leftrightarrow U_{\mathcal{S}} \sigma : U_{\mathcal{S}} A \to U_{\mathcal{S}} B.$$

3.2.3 Composition

Definition 94 Given arenas $A \to B$ and $B \to C$ and strategies $\sigma : A \to B$ and $\tau : B \to C$ we define the composite exactly as in category **G**: $\sigma;\tau$ is the unique set such that

$$U_{\mathcal{S}}(\sigma;\tau) = U_{\mathcal{S}} \sigma; U_{\mathcal{S}} \tau.$$

Lemma 95 Given arenas $A \to B$ and $B \to C$ and strategies $\sigma : A \to B$ and $\tau : B \to C$ then the composite $\sigma;\tau$ is a strategy for $A \to C$.

Proof The proof follows from lemma 93. ■

Lemma 96 Composition of strategies for QA arenas is associative: given arenas $A \to B$, $B \to C$ and $C \to D$ and strategies $\rho : A \to B$, $\sigma : B \to C$ and $\tau : C \to D$ it follows that $(\rho;\sigma);\tau = \rho;(\sigma;\tau)$.

Proof This follows directly from lemma 24. ■

Definition 97 Given an arena A we define the following strategy:

$$\mathbf{copy}^{\mathcal{S}}_{A} = \{s \in \mathcal{S}_{A' \to A''} \mid \forall s' \sqsubseteq_{\mathsf{even}} s.s' \upharpoonright A' = s' \upharpoonright A''\}$$

where the arenas A' and A'' are simply used to distinguish the different copies of A.

Lemma 98 Given a negative arena A it follows that $\mathbf{id}_{UA} = \mathbf{copy}^{\mathcal{S}}_{A}$.

Proof This follows from lemma 93 and by inspection of the definition of $\mathbf{id}_{UA}$. ■

As a corollary it follows from lemma 26 that given a negative arena A the strategy $\mathbf{copy}^{\mathcal{S}}_{A} : A \to A$ acts as the identity for the composition operation.

Proposition 99 We define a category $\mathbf{A}_{\mathcal{S}}$ of arenas for which the objects are negative arenas and each morphism from A to B is a subset of $\mathcal{S}_{A \to B}$. The category $\mathbf{A}_{\mathcal{S}}$ is well-defined.

Proof Lemma 95 assures us that the composite of two strategies is also a strategy. Lemma 96 assures us that composition is associative and lemma 98 assures us that for each game, the strategy $\mathbf{copy}^{\mathcal{S}}_{A}$ acts as the identity morphism. ■

3.3 Referees

We have now defined the category $\mathbf{A}$ of arenas and strategies. However although we could soundly model programming languages in this category it would most likely be insufficient for our purposes. There is too much junk in the category for us to achieve a suitable definability result; too many strategies that are not the denotation of any program. We would therefore like to constrain the model so that we can characterize exactly those strategies that are the denotations of programs. We might like to constrain our strategies to contain only special kinds of sequences. To achieve this we might wish to return to a category of games where each object in the category has associated with it a set of valid sequences as in chapter 2. However, we have chosen arenas as our objects and place the constraints globally. To this end we introduce the notion of a referee.

Definition 100 (Referees) A referee $\mathcal{R}$ is a collection of sets indexed by arrow arenas so that for each arrow arena A, $\mathcal{R}_A$ is a prefix-closed, non-empty subset of $\mathcal{S}_A$.

We have already spent some time in this chapter proving that composition in $\mathbf{A}$ is well-defined. We want to be sure that our refereed strategies comprise a category and hence we will be interested in referees that have the following property.

Definition 101 (Compositionality) A referee $\mathcal{R}$ is compositional if and only if for any strategies $\sigma \subseteq \mathcal{R}_{A \to B}$ and $\tau \subseteq \mathcal{R}_{B \to C}$ we have $\sigma;\tau \in \mathcal{R}_{A \to C}$.

We must also take care with identity strategies. It may be that the identity strategies in **A** do not respect a particular referee. When we construct a refereed category we will restrict the copycat strategies as follows.

Definition 102 Given a referee $\mathcal{R}$ and an arena A we define the following strategy:

$$\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}} = \mathbf{copy}^{\mathcal{S}}_{\mathsf{A}} \cap \mathcal{R}_{\mathsf{A}\to\mathsf{A}}.$$

If we are to construct a refereed category we must be sure that our constrained copycat strategies contain sufficient plays to act as identities.

Definition 103 Given a referee $\mathcal{R}$ and a negative arena A we define the set $\overline{\mathcal{R}_{\mathsf{A}}}$ to be that subset of $\mathcal{S}_{\mathsf{A}}$ such that $\mathsf{s} \in \overline{\mathcal{R}_{\mathsf{A}}}$ if and only if there exists some arena $\mathsf{A} \to \mathsf{B}$ and some sequence $\mathsf{s}' \in \mathcal{R}_{\mathsf{A}\to\mathsf{B}}$ such that $\mathsf{s} = \mathsf{s}' \upharpoonright \mathsf{A}$ or else there exists some arena $\mathsf{B} \to \mathsf{A}$ and some sequence $\mathsf{s}' \in \mathcal{R}_{\mathsf{B}\to\mathsf{A}}$ such that $\mathsf{s} = \mathsf{s}' \upharpoonright \mathsf{A}$.

Definition 104 (Copy Completeness) A referee $\mathcal{R}$ is copy complete if and only if for any negative arena A and any sequence $\mathsf{s} \in \overline{\mathcal{R}_{\mathsf{A}}}$ it follows that there exists a sequence $\mathsf{s}' \in \mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$ such that $\mathsf{s}' \upharpoonright \mathsf{A} = \mathsf{s}$.

Lemma 105 Given a copy complete referee and a strategy $\sigma \subseteq A \to B$ it it is easy to show that:

$$\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}; \sigma = \sigma = \sigma; \mathbf{copy}^{\mathcal{R}}_{\mathsf{B}}.$$

Theorem 106 For every compositional, copy complete, referee $\mathcal{R}$ there is a category $\mathbf{A}_{\mathcal{R}}$ in which the objects are negative arenas and each morphism from A to B is a subset of $\mathcal{R}_{\mathsf{A}\to\mathsf{B}}$.

Proof Compositionality assures us that for any strategies $\sigma \subseteq \mathcal{R}_{\mathsf{A}\to\mathsf{B}}$ and $\tau \subseteq \mathcal{R}_{\mathsf{B}\to\mathsf{C}}$ we have $\sigma;\tau \in \mathcal{R}_{\mathsf{A}\to\mathsf{C}}$. Lemma 96 assures us that composition is associative and by lemma 105 for each arena, A, the strategy $\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$ acts as the identity morphism. ■

Lemma 107 Given a referee $\mathcal{R}$ and any arena A we have $\varepsilon \in \mathcal{R}_{\mathsf{A}}$.

Proof By the definition of referee we have $\mathcal{R}_{\mathsf{A}}$ non-empty therefore we have $\varepsilon \in \mathcal{R}_{\mathsf{A}}$ by prefix-closure. ■

It is important to note that we can combine referees to place stronger constraints on strategies and that this affords us a modular approach to proving compositionality and copy completeness.

Definition 108 Given referees $\mathcal{R}$ and $\mathcal{R}'$ we define $(\mathcal{R} \cap \mathcal{R}')$ to be that referee for which

$$(\mathcal{R} \cap \mathcal{R}')_{\mathsf{A}} = \mathcal{R}_{\mathsf{A}} \cap \mathcal{R}'_{\mathsf{A}}$$

for each arrow arena A.

Lemma 109 Given compositional, copy complete referees $\mathcal{R}$ and $\mathcal{R}'$ it follows that $\mathcal{R} \cap \mathcal{R}'$ is a compositional and copy complete referee.

Proof It is simple to show that, for each arrow arena A, it follows that $(\mathcal{R} \cap \mathcal{R}')_{\mathsf{A}}$ is a prefix-

closed, non-empty subset of $\mathcal{S}_{\mathsf{A}}$.

Suppose we have some sequence $\mathsf{s} \in \overline{(\mathcal{R} \cap \mathcal{R}')}_{\mathsf{A}}$. This implies that one of the following is true:

- There exists some arena $\mathsf{A} \to \mathsf{B}$ and some sequence $\mathsf{t} \in (\mathcal{R} \cap \mathcal{R}')_{\mathsf{A} \to \mathsf{B}}$ such that $\mathsf{s} = \mathsf{t} \upharpoonright \mathsf{A}$. By the definition of intersection of referees this implies that $\mathsf{t} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$ and $\mathsf{t} \in \mathcal{R}'_{\mathsf{A} \to \mathsf{B}}$ hence by copy completeness we have a sequence s' in both $\mathbf{copy}_{\mathsf{A}}^{\mathcal{R}}$ and $\mathbf{copy}_{\mathsf{A}}^{\mathcal{R}'}$ such that $\mathsf{s}' \upharpoonright \mathsf{A} = \mathsf{s}$. By the definition of intersection of referees this implies that $\mathsf{s}' \in \mathbf{copy}_{\mathsf{A}}^{(\mathcal{R} \cap \mathcal{R}')}$.
- There exists some arena $\mathsf{B} \to \mathsf{A}$ and some sequence $\mathsf{t} \in (\mathcal{R} \cap \mathcal{R}')_{\mathsf{B} \to \mathsf{A}}$ such that $\mathsf{s} = \mathsf{t} \upharpoonright \mathsf{A}$ and our reasoning is similar.

Hence intersection preserves copy completeness.

Given strategies $\sigma \subseteq (\mathcal{R} \cap \mathcal{R}')_{\mathsf{A} \to \mathsf{B}}$ and $\tau \subseteq (\mathcal{R} \cap \mathcal{R}')_{\mathsf{B} \to \mathsf{C}}$ we have $\sigma \subseteq \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$ and $\tau \subseteq \mathcal{R}_{\mathsf{B} \to \mathsf{C}}$ and hence $\sigma ; \tau \subseteq \mathcal{R}_{\mathsf{A} \to \mathsf{C}}$ by the compositionality of $\mathcal{R}$. Similarly it follows that $\sigma ; \tau \subseteq \mathcal{R}'_{\mathsf{A} \to \mathsf{C}}$ and hence $\sigma ; \tau \subseteq (\mathcal{R} \cap \mathcal{R}')_{\mathsf{A} \to \mathsf{C}}$ by the definition of $(\mathcal{R} \cap \mathcal{R}')_{\mathsf{A} \to \mathsf{C}}$. ■

Definition 110 (Bias) A referee $\mathcal{R}$ is biased if and only if for any odd-length sequence $\mathsf{s} \cdot \mathsf{m} \in \mathcal{S}_{\mathsf{A} \to \mathsf{B}}$ such that

$$\mathsf{s} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}} \wedge \mathsf{s} \cdot \mathsf{m} \upharpoonright \mathsf{A} \in \overline{\mathcal{R}_{\mathsf{A}}} \wedge \mathsf{s} \cdot \mathsf{m} \upharpoonright \mathsf{B} \in \overline{\mathcal{R}_{\mathsf{B}}}$$

it follows that $\mathsf{s} \cdot \mathsf{m} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$.

Definition 111 (Admissibility) We say that a referee is admissible if and only if it is compositional, copy complete and biased.

Lemma 112 Given admissible referees $\mathcal{R}$ and $\mathcal{R}'$ it follows that $\mathcal{R} \cap \mathcal{R}'$ is also admissible.

Proof The proof is straightforward and follows simply from the definitions of compositionality, copy completeness and bias. ■

We now consider a class of referee for which it is very simple to demonstrate admissibility.

Definition 113 (Separability) A referee $\mathcal{R}$ is separable if and only if for any sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A} \to \mathsf{B}}$ we have:

$$(\mathsf{s} \upharpoonright \mathsf{A} \in \mathcal{R}_{\mathsf{A}} \wedge \mathsf{s} \upharpoonright \mathsf{B} \in \mathcal{R}_{\mathsf{B}}) \Leftrightarrow (\mathsf{s} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}}).$$

Lemma 114 Any separable referee $\mathcal{R}$ is admissible.

Proof We check each item in the definition of admissibility.

Compositionality Given strategies $\sigma \subseteq \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$ and $\tau \subseteq \mathcal{R}_{\mathsf{B} \to \mathsf{C}}$ and $\mathsf{s} \in \sigma ; \tau$ we have $\mathsf{s} \upharpoonright \mathsf{A} \in \mathcal{R}_{\mathsf{A}}$ and $\mathsf{s} \upharpoonright \mathsf{C} \in \mathcal{R}_{\mathsf{C}}$ by locality and hence $\mathsf{s} \in \mathcal{R}_{\mathsf{A} \to \mathsf{C}}$ by separability and therefore $\mathcal{R}$ is compositional.

Bias For any odd-length sequence $\mathsf{s} \cdot \mathsf{m} \in \mathcal{S}_{\mathsf{A} \to \mathsf{B}}$ such that

$$(\mathsf{s} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}} \wedge \mathsf{s} \cdot \mathsf{m} \upharpoonright \mathsf{A} \in \mathcal{R}_{\mathsf{A}} \wedge \mathsf{s} \cdot \mathsf{m} \upharpoonright \mathsf{B} \in \mathcal{R}_{\mathsf{B}})$$

it follows that $\mathsf{s} \cdot \mathsf{m} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$ directly from the definition of separability.

Copy Completeness Given any negative arena A and any sequence $\mathsf{s} \in \mathcal{R}_{\mathsf{A}}$ it follows that there exists a sequence $\mathsf{t} \in \mathbf{copy}^{\mathcal{S}}_{\mathsf{A}}$ such that $\mathsf{t} \upharpoonright \mathsf{A}' = \mathsf{t} \upharpoonright A'' = \mathsf{s}$ (where once again A' and A'' are used to distinguish different copies of arena A). From the definition of separability it therefore follows that $\mathsf{t} \in \mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$.

■

Definition 115 Given referees $\mathcal{R}$ and $\mathcal{R}'$ we write $\mathcal{R} \subseteq \mathcal{R}'$ if and only if

$$\mathcal{R}_{\mathsf{A}} \subseteq \mathcal{R}'_{\mathsf{A}}$$

for each arrow arena A.

Definition 116 Given a referee $\mathcal{R}$ and an arrow arena

$$\mathsf{A} = \langle \mathsf{M}_{\mathsf{A}}, \vdash_{\mathsf{A}}, \lambda_{\mathsf{A}} \rangle$$

we define the following game:

$$\mathsf{U}_{\mathcal{R}}\mathsf{A} = \langle \mathsf{M}_{\mathsf{A}}, \lambda^{OP}_{\mathsf{A}}, \overline{\mathcal{R}_{\mathsf{A}}} \rangle.$$

We again overload our definition so that for any sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ we have $\mathsf{U}_{\mathcal{R}}\mathsf{s} = \mathsf{s}$ and for any set $S \subseteq \mathcal{S}_{\mathsf{A}}$ we have $\mathsf{U}_{\mathcal{R}}S = S$.

Lemma 117 Given a referee $\mathcal{R}$ and a sequence $\mathsf{s} \in \mathcal{R}_{\mathsf{A}\to\mathsf{B}}$ we have $\mathsf{U}_{\mathcal{R}}\mathsf{s} \in \mathcal{P}_{\mathsf{U}_{\mathcal{R}}\mathsf{A}\to\mathsf{U}_{\mathcal{R}}\mathsf{B}}$.

Proof The proof follows from the definition $\overline{\mathcal{R}_{\mathsf{A}}}$ and $\overline{\mathcal{R}_{\mathsf{B}}}$ and by inspection of the definition of $\mathcal{P}_{\mathsf{U}_{\mathcal{R}}\mathsf{A}\to\mathsf{U}_{\mathcal{R}}\mathsf{B}}$. ■

As a corollary it follows that for any strategy $\sigma \subseteq \mathcal{R}_{\mathsf{A}\to\mathsf{B}}$ we have

$$\mathsf{U}_{\mathcal{R}}\sigma : \mathsf{U}_{\mathcal{R}}\mathsf{A} \to \mathsf{U}_{\mathcal{R}}\mathsf{B}.$$

Lemma 118 Given a referee $\mathcal{R}$ and strategies $\sigma \subseteq \mathcal{R}_{\mathsf{A}\to\mathsf{B}}$ and $\tau \subseteq \mathcal{R}_{\mathsf{B}\to\mathsf{C}}$ it follows that:

1. $\mathsf{U}_{\mathcal{R}}(\sigma;\tau) = \mathsf{U}_{\mathcal{R}}\sigma;\mathsf{U}_{\mathcal{R}}\tau$.
2. $\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}} \subseteq \mathbf{id}_{\mathsf{U}_{\mathcal{R}}\mathsf{A}}$.
3. $\mathbf{id}_{\mathsf{U}_{\mathcal{R}}\mathsf{A}} = \mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$ for every negative arena A if and only if $\mathcal{R}$ is admissible.

Proof The first statement follows from lemma 29 and the fact that $\mathsf{U}_{\mathcal{R}}\mathsf{A}$, $\mathsf{U}_{\mathcal{R}}\mathsf{B}$ and $\mathsf{U}_{\mathcal{R}}\mathsf{C}$ are subgames of $\mathsf{U}_{\mathcal{S}}\mathsf{A}$, $\mathsf{U}_{\mathcal{S}}\mathsf{B}$ and $\mathsf{U}_{\mathcal{S}}\mathsf{C}$ respectively. It is straightforward to check the second statement. The third statement follows from the copy completeness of $\mathcal{R}$. ■

3.4 Sequence Transformers

In this section we examine a novel way of showing the admissibility of referees by reducing referee membership to visibility in **G**.

Definition 119 Given a referee $\mathcal{R}$ we define a sequence transformer $K : \mathcal{R}$ to be a family of functions indexed by negative arenas such that for each negative arena A we have a function $K_{\mathsf{A}} : \overline{\mathcal{R}_{\mathsf{A}}} \mapsto \mathcal{X}_{\mathsf{A}}$ that satisfies the following:

Sequence Preservation For each sequence $\mathsf{s} \in \mathcal{R}_{\mathsf{A}}$ where

$$\mathsf{s} = \langle \mathsf{M}_{\mathsf{s}}, \lhd_{\mathsf{s}}, \curvearrowleft_{\mathsf{s}} \rangle \text{ and } K_{\mathsf{A}}\mathsf{s} = \langle \mathsf{M}_{K_{\mathsf{A}}\mathsf{s}}, \lhd_{K_{\mathsf{A}}\mathsf{s}}, \curvearrowleft_{K_{\mathsf{A}}\mathsf{s}} \rangle$$

it follows that:

$$\mathsf{M}_{\mathsf{s}} = \mathsf{M}_{K_{\mathsf{A}}\mathsf{s}} \text{ and } \lhd_{\mathsf{s}} = \lhd_{K_{\mathsf{A}}\mathsf{s}}.$$

In other words the transformer leaves underlying sequences of moves unchanged and only adds or removes justification pointers.

Prefix Respect For any arena A and sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{R}_{\mathsf{A}}$ we have

$$\mathsf{s} \sqsubseteq \mathsf{s}' \Rightarrow K\mathsf{s} \sqsubseteq K\mathsf{s}'.$$

We may omit subscripts from our transformers when the index can be readily inferred from the context.

We lift such a K to sequences $\mathsf{s} \in \mathcal{P}_{\mathsf{U}_{\mathcal{R}}\mathsf{A} \to \mathsf{U}_{\mathcal{R}}\mathsf{B}}$ where $\mathsf{s} = \langle \mathsf{M}_{\mathsf{s}}, \lhd_{\mathsf{s}}, \curvearrowleft_{\mathsf{s}} \rangle$ as follows:

$$\mathsf{M}_{\mathsf{s}} = \mathsf{M}_{K\mathsf{s}} \text{ and } \lhd_{\mathsf{s}} = \lhd_{K\mathsf{s}}$$

and also for any moves m, m' it follows that $\mathsf{m} \curvearrowleft_{K\mathsf{s}} \mathsf{m}'$ if and only if

$$\mathsf{m} \curvearrowleft_{K(\mathsf{s}\restriction \mathsf{U}_{\mathcal{R}}A)} \mathsf{m}' \text{ or } \mathsf{m} \curvearrowleft_{K(\mathsf{s}\restriction \mathsf{U}_{\mathcal{R}}B)} \mathsf{m}'$$

Similarly, given a sequence $\mathsf{s} \in \mathbf{int}(\mathsf{U}_{\mathcal{R}}\mathsf{A}, \mathsf{U}_{\mathcal{R}}\mathsf{B}, \mathsf{U}_{\mathcal{R}}\mathsf{C})$ we define $K\mathsf{s}$ as follows:

$$\mathsf{M}_{\mathsf{s}} = \mathsf{M}_{K\mathsf{s}} \text{ and } \lhd_{\mathsf{s}} = \lhd_{K\mathsf{s}}$$

and also for any moves m, m' it follows that $\mathsf{m} \curvearrowleft_{K\mathsf{s}} \mathsf{m}'$ if and only if

$$\mathsf{m} \curvearrowleft_{K(\mathsf{s}\restriction \mathsf{U}_{\mathcal{R}}\mathsf{A})} \mathsf{m}' \text{ or } \mathsf{m} \curvearrowleft_{K(\mathsf{s}\restriction \mathsf{U}_{\mathcal{R}}\mathsf{B})} \mathsf{m}' \text{ or } \mathsf{m} \curvearrowleft_{K(\mathsf{s}\restriction \mathsf{U}_{\mathcal{R}}\mathsf{C})} \mathsf{m}'.$$

We will further overload our definition to sets of sequences. Given a sequence transformer K and a set or sequences S we define:

$$KS = \{K\mathsf{s} \mid \mathsf{s} \in S\}.$$

Finally, we overload the definition to map negative arenas onto games. Given an arena $\mathsf{A} = \langle \mathsf{M}_{\mathsf{A}}, \vdash_{\mathsf{A}}, \lambda_{\mathsf{A}} \rangle$ we define:

$$K\mathsf{A} = \langle \mathsf{M}_{\mathsf{A}}, \lambda^{OP}_{\mathsf{A}}, K\overline{\mathcal{R}_{\mathsf{A}}} \rangle.$$

Lemma 120 Given a referee $\mathcal{R}$, a sequence $\mathsf{s} \in \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$ and a sequence transformer $K : \mathcal{R}$ it is straightforward to show that $K\mathsf{s} \in \mathcal{P}_{K\mathsf{A} \to K\mathsf{B}}$. As a corollary it follows that given a strategy $\sigma \subseteq \mathcal{R}_{\mathsf{A} \to \mathsf{B}}$ it follows that $K\sigma$ is a strategy for the game $K\mathsf{A} \to K\mathsf{B}$.

Lemma 121 Given negative arenas A, B and C, a sequence transformer $K : \mathcal{R}$ and strategies $\sigma : \mathsf{A} \to \mathsf{B}$ and $\tau : \mathsf{B} \to \mathsf{C}$ it follows that:

$$K(\sigma;\tau) \subseteq K\sigma;K\tau.$$

Proof Consider any sequence $Ks \in K(\sigma;\tau)$ and let $u \in \mathbf{int}(\mathsf{U}_{\mathcal{R}}\mathsf{A}, \mathsf{U}_{\mathcal{R}}\mathsf{B}, \mathsf{U}_{\mathcal{R}}\mathsf{C})$ witness s. We have $K(u \upharpoonright \mathsf{U}_{\mathcal{R}}A, \mathsf{U}_{\mathcal{R}}B) \in K\sigma$ and $K(u \upharpoonright \mathsf{U}_{\mathcal{R}}B, \mathsf{U}_{\mathcal{R}}C) \in K\tau$ and hence $(Ku) \upharpoonright K\mathsf{A}, K\mathsf{B} \in K\sigma$ and $(Ku) \upharpoonright K\mathsf{B}, K\mathsf{C} \in K\tau$. Hence $Ku \in K\sigma \| K\tau$ and $(Ku) \upharpoonright K\mathsf{A}, K\mathsf{C} \in K\sigma;K\tau$ and therefore $s \in K\sigma;K\tau$.

We should note here that it will not in general be the case that

$$K\sigma;K\tau \subseteq K(\sigma;\tau)$$

as it may be the case that we have sequences $s \in \sigma$ and $s' \in \tau$ such that $s \upharpoonright \mathsf{B} \neq s' \upharpoonright \mathsf{B}$ but $Ks \upharpoonright K\mathsf{B} = Ks' \upharpoonright K\mathsf{B}$. ■

Lemma 122 Given a negative arena A and a sequence transformer $K : \mathcal{R}$ it follows that:

- $K\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}} \subseteq \mathbf{id}_{K\mathsf{A}}$.
- $\mathbf{id}_{K\mathsf{A}} \subseteq K\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$ for every arena A if and only if $\mathcal{R}$ is copy complete.

Proof We prove the first statement as follows. Given any sequence $s \in \mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$ we have $s \upharpoonright \mathsf{A}' = s \upharpoonright \mathsf{A}''$ and hence by the definition of sequence transformers we know that $Ks \upharpoonright K\mathsf{A}' = Ks \upharpoonright K\mathsf{A}''$ and hence $s \in \mathbf{id}_{K\mathsf{A}}$.

Proof of the second statement involves two parts. First we assume that $\mathcal{R}$ is copy complete and show that $\mathbf{id}_{K\mathsf{A}} \subseteq K\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$. Given a sequence $s \in \mathbf{id}_{K\mathsf{A}}$ it of course follows that $s \upharpoonright K\mathsf{A}' = s \upharpoonright K\mathsf{A}'' = Kt$ for some $t \in \overline{\mathcal{R}_{\mathsf{A}}}$. If $\mathcal{R}$ is copy complete then there exists some sequence $u \in \mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$ such that $u \upharpoonright \mathsf{A} = t$ where $Ku = s$ and hence $\mathbf{id}_{K\mathsf{A}} \subseteq K\mathbf{copy}^{\mathcal{R}}_{\mathsf{A}}$. ■

Definition 123 We say that a sequence transformer K is injective if and only if for each arena A it follows that K_{A} is injective.

Lemma 124 Given negative arenas A, B and C, an injective sequence transformer $K : \mathcal{R}$ and strategies $\sigma : \mathsf{A} \to \mathsf{B}$ and $\tau : \mathsf{B} \to \mathsf{C}$ it follows that:

$$K(\sigma;\tau) = K\sigma;K\tau.$$

Proof We already know that $K(\sigma;\tau) \subseteq K\sigma;K\tau$ from lemma 121. Consider any sequence $s \in K\sigma;K\tau$ and let $u \in \mathbf{int}(\mathsf{KA}, \mathsf{KB}, \mathsf{KC})$ witness s. By injectivity we know that $u = Kt$ for exactly one sequence t. Let $Kt' = Kt \upharpoonright K\mathsf{A}, K\mathsf{B}$ and we know that $Kt' \in K\sigma$ and let $Kt'' = Kt \upharpoonright K\mathsf{B}, K\mathsf{C}$ and similarly we have $Kt'' \in K\tau$. By the injectivity of K we know that $t' \upharpoonright \mathsf{B} = t'' \upharpoonright \mathsf{B}$ and it is straightforward to show that t witnesses a sequence $t \upharpoonright \mathsf{U}_{\mathcal{A}}, \mathsf{U}_{\mathcal{B}} \in \sigma;\tau$ and hence $s \in K(\sigma;\tau)$. ■

Lemma 125 Given an admissible referee $\mathcal{R}'$, another referee $\mathcal{R}$ such that $\mathcal{R} \subseteq \mathcal{R}'$ and a sequence transformer $K : \mathcal{R}'$ such that for any arrow A and any sequence $s \in \mathcal{R}'_{\mathsf{A}}$ it is the case that:

$$s \in \mathcal{R}_{\mathsf{A}} \text{ if and only if } Ks \text{ is player visible.}$$

then it follows that $\mathcal{R}$ is admissible.

Proof

Compositionality Suppose we are given strategies $\sigma \subseteq \mathcal{R}_{A \to B}$ and $\tau \subseteq \mathcal{R}_{A \to B}$.

We know that $K\sigma$ and $K\tau$ are player visible and by lemma 60 $K\sigma;K\tau$ is player visible. We know that $\sigma;\tau \in \mathcal{R}'_{A \to C}$ by the compositionality of $\mathcal{R}'$ and by lemma 121 it follows that $K(\sigma;\tau)$ is player visible and hence $\sigma;\tau \subseteq \mathcal{R}_{A \to C}$ and therefore $\mathcal{R}$ is compositional.

Bias For any odd-length sequence $s \cdot m \in \mathcal{S}_{A \to B}$ such that

$$(s \in \mathcal{R}_{A \to B} \wedge s \cdot m \upharpoonright A \in \overline{\mathcal{R}_A} \wedge s \cdot m \upharpoonright B \in \overline{\mathcal{R}_B})$$

it follows that Ks is player visible and hence $Ks \cdot m$ is player visible and $s \cdot m \in \mathcal{R}_{A \to B}$.

Copy Completeness By lemma 62 we know that the strategy $\mathbf{id}_{KA}$ is player visible and from lemma 122 we therefore know that $K\mathbf{copy}^{\mathcal{R}'}_A$ is player visible. Hence $\mathbf{copy}^{\mathcal{R}'}_A \subseteq \mathcal{R}_{A \to A}$ and therefore $\mathbf{copy}^{\mathcal{R}'}_A = \mathbf{copy}^{\mathcal{R}}_A$. From the definition of admissibility we know that given any sequence $s \in \mathcal{R}'_A$ it follows that there exists a sequence $t \in \mathbf{copy}^{\mathcal{R}'}_A$ such that $t \upharpoonright A' = t \upharpoonright A'' = s$ (where once again A' and A'' are used to distinguish different copies of arena A). So it follows that copy completeness is satisfied.

■

Lemma 126 Given an admissible referee $\mathcal{R}'$, another referee $\mathcal{R}$ such that $\mathcal{R} \subseteq \mathcal{R}'$ and a sequence transformer $K : \mathcal{R}'$ such that :

$$s \in \mathcal{R}_A \text{ if and only if } Ks \text{ is totally visible.}$$

then it follows that $\mathcal{R}$ is admissible.

Proof

Compositionality Given strategies $\sigma \subseteq \mathcal{R}_{A \to B}$ and $\tau \subseteq \mathcal{R}_{B \to C}$ it follows that $K\sigma$ and $K\tau$ respect total visibility and hence by lemmas 65 and 60 we know that $K\sigma;K\tau$ respects total visibility and by lemma 121 we therefore know that $K(\sigma;\tau)$ respects total visibility and by the compositionality of $\mathcal{R}'$ we know $\sigma;\tau \in \mathcal{R}'_{A \to C}$ and hence $\sigma;\tau \in \mathcal{R}_{A \to C}$.

Bias For any odd-length sequence $s \cdot m \in \mathcal{S}_{A \to B}$ such that

$$(s \in \mathcal{R}_{A \to B} \wedge s \cdot m \upharpoonright A \in \overline{\mathcal{R}_A} \wedge s \cdot m \upharpoonright B \in \overline{\mathcal{R}_B})$$

It follows that Ks, $Ks \cdot m \upharpoonright KA$ and $Ks \cdot m \upharpoonright KB$ are totally visible and hence by lemma 70 we know that $Ks \cdot m$ is totally visible. By the bias of $\mathcal{R}'$ we know $s \cdot m \in \mathcal{R}'_{A \to B}$ and hence $s \cdot m \in \mathcal{R}_{A \to B}$.

Copy Completeness Given any negative arena A and any sequence $s \in \mathcal{R}_A$ it follows from the copy completeness of $\mathcal{R}'$ that there exists a sequence $t \in \mathbf{copy}^{\mathcal{R}'}_A$ such that

$$t \upharpoonright A' = t \upharpoonright A'' = s$$

(where once again A' and A'' are used to distinguish different copies of arena A). We know that Ks respects total visibility and from lemma 122 that $Kt \in \mathbf{id}_{KA}$ hence by lemma 64 we know that Kt respects total visibility and thus $t \in \mathcal{R}_{A \to A}$. We therefore have $t \in \mathbf{copy}^{\mathcal{R}}_A$ and $\mathcal{R}$ is therefore copy complete.

■

3.5 Safety Constraints on Strategies

We will now examine some constraints on strategies that are present in models in the game semantics literature. We will show that these constraints can be reformulated as admissible referees. We will therefore have demonstrated a fairly robust and uniform way of proving that certain constraints placed on strategies are preserved by composition and that similarly constraining the copycat strategies does not destroy their ability to act as the unit for composition.

3.5.1 Unrestrained Strategies

Lemma 127 The referee $\mathcal{S}$ is separable, and hence admissible by lemma 114.

Proof This follows from lemma 90. ■

3.5.2 Visibility

The visibility conditions that we will shortly be examining were present in the games model for PCF in [28]. The repercussions of relaxing these conditions were studied by Abramsky, Honda and McCusker in [1] and the relaxation was shown to allow the interpretation of higher order store.

Definition 128 (Views) The player view (p-view) of a justified sequence s, notated $\lceil s \rceil$, is a subsequence of s defined inductively as follows:

- $\lceil \varepsilon \rceil = \varepsilon$.
- $\lceil t \cdot m \rceil = \lceil t \rceil \cdot m$ m is a player move.
- $\lceil t \cdot m \rceil = m$ where m is an initial opponent move.
- $\lceil t \cdot j \overset{\frown}{\cdot t' \cdot} m \rceil = \lceil t \rceil \cdot j \overset{\frown}{\cdot} m$ where m is a non-initial opponent move.

Similarly, the opponent view (o-view) of a justified sequence s, notated $\lfloor s \rfloor$, is a subsequence of s defined inductively as follows:

- $\lfloor \varepsilon \rfloor = \varepsilon$.
- $\lfloor t \cdot m \rfloor = \lfloor t \rfloor \cdot m$ where m is an opponent move.
- $\lfloor t \cdot m \rfloor = m$ where m is an initial player move.
- $\lfloor t \cdot j \overset{\frown}{\cdot t' \cdot} m \rfloor = \lfloor t \rfloor \cdot j \overset{\frown}{\cdot} m$ where m is a non-initial player move.

Definition 129 (Visibility) Given an arena A and a strategy $\sigma : A$ we say that σ obeys player visibility if and only if for all $s \in \sigma$ and moves $j, m \in s$ such that $j \curvearrowleft m$ we have $j \in \lceil s_{\leq m} \rceil$. Similarly, we say that σ obeys opponent visibility if and only if for all $s \in \sigma$ and moves $j, m \in s$ such that $j \curvearrowleft m$ we have $j \in \lfloor s_{\leq m} \rfloor$ and of course σ obeys total visibility if and only if it obeys both player and opponent visibility.

Lemma 130 Given an arena A it is straightforward to show that the definitions of views and visibility for a sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ are equivalent to those in **G**. For all sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{J}_{\mathsf{A}}$ it follows that $(\lceil \mathsf{s} \rceil = \lceil \mathsf{s}' \rceil) \Leftrightarrow (\lceil \mathsf{U}_{\mathcal{S}} \mathsf{s} \rceil = \lceil \mathsf{U}_{\mathcal{S}} \mathsf{s}' \rceil)$ and for an move $\mathsf{m} \in \mathsf{s}$ it follows that $(\mathsf{m} \in \lceil \mathsf{s} \rceil) \Leftrightarrow (\mathsf{m} \in \lceil \mathsf{U}_{\mathcal{S}} \mathsf{s} \rceil)$.

Definition 131 We define the referee $\mathcal{V}^p$ so that for any arena A we have $\mathsf{s} \in \mathcal{V}^p_{\mathsf{A}}$ if and only if s respects player visibility.

We define the referee $\mathcal{V}^o$ so that for any arena A we have $\mathsf{s} \in \mathcal{V}^o_{\mathsf{A}}$ if and only if s respects opponent visibility.

We define the referee $\mathcal{V}$ so that for any arena A we have $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ if and only if s respects total visibility.

Definition 132 We define the sequence transformer $I : \mathcal{S}$ such that for each arrow arena A and each sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ we have $I_{\mathsf{A}} \mathsf{s} = \mathsf{s}$.

Lemma 133 Given a sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ it follows that $\mathsf{s} \in \mathcal{V}^p_{\mathsf{A}}$ if and only if $I\mathsf{s}$ is player visible, $\mathsf{s} \in \mathcal{V}^o_{\mathsf{A}}$ if and only if $I\mathsf{s}$ is opponent visible and also $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ if and only if $I\mathsf{s}$ is totally visible.

Proof The proof follows from lemma 130. ■

Lemma 134 The referees $\mathcal{V}$ and $\mathcal{V}^p$ are admissible.

Proof The referee $\mathcal{V}$ is admissible by lemmas 133 and 126. The admissibility of referee $\mathcal{V}^p$ follows from lemmas 133 and 125. ■

3.5.3 Bracketing

Bracketing conditions are present in the model of PCF in [28]. A thorough examination of the implications of relaxing these conditions are studied by Jim Laird in [31] and are found to coincide with the addition of control features to the language.

Definition 135 (Well-Bracketing) When an answer m' is justified by a question m we say that m' answers m. We say that a sequence $\mathsf{s} \in \mathcal{J}_{\mathsf{A}}$ is well bracketed if and only if all answers in s are justified by the most recent unanswered question.

Definition 136 (Player Bracketing and Opponent Bracketing) We say that a sequence s is player bracketed if and only if, for every player answer $\mathsf{a} \in \mathsf{s}$, a answers the most recent unanswered question in $\lceil \mathsf{s}_{\leq \mathsf{a}} \rceil$.

Similarly, we say that a sequence s is opponent bracketed if and only if, for every opponent answer $\mathsf{a} \in \mathsf{s}$, a answers the most recent unanswered question in $\lfloor \mathsf{s}_{\leq \mathsf{a}} \rfloor$.

Definition 137 We define the referee $\mathcal{B}^p$ so that for any arena A we have $\mathsf{s} \in \mathcal{B}^p_{\mathsf{A}}$ if and only if s is player bracketed.

We define the referee $\mathcal{B}^o$ so that for any arena A we have $\mathsf{s} \in \mathcal{B}^o_{\mathsf{A}}$ if and only if s is opponent bracketed.

We define the referee $\mathcal{B}$ so that for any arena A we have $\mathsf{s} \in \mathcal{B}_{\mathsf{A}}$ if and only if s is well-bracketed.

We now prove that a sequence $s \in \mathcal{J}_A$ is well-bracketed if and only if it is both player bracketed and opponent bracketed.

Lemma 138 Given a well bracketed sequence $s \in \mathcal{J}_A$, both $\lceil s \rceil$ and $\lfloor s \rfloor$ are well bracketed.

Proof We carry out induction on the length of s.

Base Case If $s = \varepsilon$ then the proof is trivial.

Inductive Step Suppose $s = s' \cdot m$.

case: We first consider the case when $\lambda m = OQ$, and omit the case when $\lambda m = PQ$ as it is similar. First we show that $\lfloor s' \cdot m \rfloor$ is well bracketed. By definition we know that $\lfloor s \rfloor = \lfloor s' \rfloor \cdot m$ and $\lfloor s' \rfloor$ is well bracketed by inductive hypothesis hence $\lfloor s' \rfloor \cdot m$ is well bracketed. We now show that $\lceil s' \cdot m \rceil$ is well bracketed. If m is initial then $\lceil s \rceil = m$, which is well bracketed. Otherwise let $j \curvearrowleft m$ and we have $\lceil s \rceil = \lceil s_{\leq j} \rceil \cdot m$, and by inductive hypothesis $\lceil s_{\leq j} \rceil$ is well bracketed and hence $\lceil s_{\leq j} \rceil \cdot m$ is well bracketed.

case: Now suppose m is the answer to j. We consider only the case when $\lambda m = OA$, as the case when $\lambda m = PA$ is similar. We have $\lceil s \rceil = \lceil s_{\leq j} \rceil \cdot m$ where $\lceil s_{\leq j} \rceil$ is well bracketed by inductive hypothesis and hence so is $\lceil s_{\leq j} \rceil \cdot m$ as m is an answer to the most recent unanswered question in the view.

We must now show that $\lfloor s \rfloor$ is well bracketed. As $\lfloor s \rfloor = \lfloor s' \rfloor \cdot m$ we have $\lfloor s' \rfloor$ well bracketed by inductive hypothesis. It is now left for us to show that j is the most recent unanswered question in $\lfloor s' \rfloor$. If j immediately precedes m then this is assured, otherwise let j^+ immediately succeed j. It follows from the well-bracketing of s that j^+ must be a question and must be answered in s' by some move a. It also follows from the definition of opponent view that j is the most recent unanswered question in $\lfloor s' \cdot m_{\leq a} \rfloor$. Let a' be the greatest player answer in $s' \cdot m$ such that j is the most recent unanswered question in $\lfloor s' \cdot m_{\leq a'} \rfloor$. Now suppose the successor of a', which we call a'^+, is also in s'. It follows that a'^+ would be a question as, by inductive hypothesis $\lfloor s' \cdot m_{\leq a'^+} \rfloor$ is well-bracketed. However, it also follows that a'^+ would be answered in s' by some move a'' and by inspection of the definition of opponent view we would have

$$\lfloor s' \cdot m_{\leq a''} \rfloor = \lfloor s' \cdot m_{\leq a'} \rfloor \cdot a'^+ \cdot a''$$

which would conflict with our definition of a' as the greatest player answer in $s' \cdot m$ such that j is the most recent unanswered question in $\lfloor s' \cdot m_{\leq a'} \rfloor$. We therefore conclude that no such move a'^+ exists and a' is the last move in s' and hence m answers the most recent unanswered question in $\lfloor s' \cdot m \rfloor$. ■

Lemma 139 Given a sequence s such that for every prefix $t \sqsubseteq s$ we have $\lceil t \rceil$ and $\lfloor t \rfloor$ well bracketed, s is well bracketed.

Proof We construct a proof by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is trivial.

Inductive Step Suppose $s = s' \cdot m$. If m is a question then we merely need to appeal to the inductive hypothesis. Let us consider the case where $\lambda m = PA$, as the case when $\lambda m = OA$ is similar.

We first check that the most recent unanswered question, q in $\lceil s' \rceil$ is also the most recent in s'. As $\lceil s \rceil$ is well bracketed it must be of the form:

$$\cdots q \cdot \overbrace{q^1 \cdot a^1 \ldots q^n \cdot a^n}^{} \cdot m.$$

We can apply our inductive hypothesis to s' so the only way for s to violate well bracketing is for there to be an unanswered question between some pair $q^i \cdot a^i$ but by inductive hypothesis $s_{\leq a^i}$ is well bracketed hence all questions played between q^i and i^i must be answered before a^i is played. ■

Lemma 140 A sequence $s \in \mathcal{J}_A$ is well-bracketed if and only if it is both player bracketed and opponent bracketed.

Proof This follows from lemmas 138 and 139. ■

Definition 141 We now define a sequence transformer, $B : \mathcal{S}$, as follows. Give an arrow arena A and a sequence $s = \langle M_s, \prec_s, \curvearrowleft_s \rangle$ in $\mathcal{S}_A$ we define:

$$B_A s = \langle M_s, \prec_s, \{(m, m') \in \curvearrowleft_s | \ \lambda_A^{QA} m' = A\} \rangle.$$

In other words we have the same sequence of moves but only include the justifiers between questions and their answers.

Lemma 142 Given a sequence $s \in \mathcal{S}_A$ it follows that:

- $s \in \mathcal{B}_A^p$ if and only if Bs is player visible.
- $s \in \mathcal{B}_A^o$ if and only if Bs is opponent visible.
- $s \in \mathcal{B}_A$ if and only if Bs is totally visible.

Proof We prove the first of these statements by induction on an arbitrary sequence $s \in \sigma$. Proof of the second statement is similar. Proof of the third statement follows from the first two by an application of lemma 140.

Base Case If $s = \varepsilon$ then then s is player bracketed and Bs is player visible.

Inductive Step Suppose $s = s' \cdot m$.

case: If m is an opponent move or a player question then we have $s' \cdot m$ is player bracketed if and only if s' is player bracketed and $Bs' \cdot m$ is player visible if and only if Bs' is player visible. We can now simply apply the inductive hypothesis to s'.

case: If m is a player answer to a move q then $s' \cdot m$ is player bracketed if and only if s' is player bracketed and q is the most recent unanswered question in $\lceil s' \rceil$. Inspection of the definition of B shows us that q is the most recent unanswered question in $\lceil s' \rceil$ if and only if q is in $\lceil Bs' \rceil$. ■

Lemma 143 The referees $\mathcal{B}$ and $\mathcal{B}^p$ are admissible.

Proof The referee $\mathcal{B}$ is compositional by lemmas 142 and 126. The referee $\mathcal{B}^p$ is admissible by lemmas 142 and 125. ■

3.5.4 Rigidity

In [17] Danos and Harmer further restrict the Hyland-Ong/Nickau games model for PCF and propose the notion of rigidity. The restricted model is fully abstract for a language without sequencing or case constructs.

Definition 144 (Rigid View) The rigid player view of a justified sequence s, notated $\lceil \mathsf{s} \rceil^{\mathsf{rig}}$, is a subsequence of s defined inductively as follows:

- $\lceil \varepsilon \rceil^{\mathsf{rig}} = \varepsilon$.
- $\lceil \mathsf{t} \cdot \mathsf{m} \rceil^{\mathsf{rig}} = \lceil \mathsf{t} \rceil^{\mathsf{rig}} \cdot \mathsf{m}$ where $\lambda^{OP}\mathsf{m} = P$.
- $\lceil \mathsf{t} \cdot \mathsf{m} \rceil^{\mathsf{rig}} = \mathsf{m}$ where m is an initial opponent move or an opponent question.
- $\lceil \mathsf{t} \cdot \mathsf{j} \overset{\frown}{\cdot \mathsf{t}' \cdot \mathsf{m}} \rceil^{\mathsf{rig}} = \lceil \mathsf{t} \rceil^{\mathsf{rig}} \cdot \mathsf{j} \overset{\frown}{\cdot \mathsf{m}}$ where m is a non-initial opponent question.

Similarly, the rigid opponent view of a justified sequence s, notated $\lfloor \mathsf{s} \rfloor^{\mathsf{rig}}$, is a subsequence of s defined inductively as follows:

- $\lfloor \varepsilon \rfloor^{\mathsf{rig}} = \varepsilon$.
- $\lfloor \mathsf{t} \cdot \mathsf{m} \rfloor^{\mathsf{rig}} = \lfloor \mathsf{t} \rfloor^{\mathsf{rig}} \cdot \mathsf{m}$ where $\lambda^{OP}\mathsf{m} = O$.
- $\lfloor \mathsf{t} \cdot \mathsf{m} \rfloor^{\mathsf{rig}} = \mathsf{m}$ where m is an initial player move or a player question.
- $\lfloor \mathsf{t} \cdot \mathsf{j} \overset{\frown}{\cdot \mathsf{t}' \cdot \mathsf{m}} \rfloor^{\mathsf{rig}} = \lfloor \mathsf{t} \rfloor^{\mathsf{rig}} \cdot \mathsf{j} \overset{\frown}{\cdot \mathsf{m}}$ where m is a non-initial player question.

Definition 145 A sequence s is player rigid if and only if for every player question $\mathsf{q} \in \mathsf{s}$ justified by move j we have $\mathsf{j} \in \lceil \mathsf{s}_{\leq \mathsf{q}} \rceil^{\mathsf{rig}}$. Similarly, a sequence s is opponent rigid if and only if for every opponent question $\mathsf{q} \in \mathsf{s}$ justified by move j we have $\mathsf{j} \in \lfloor \mathsf{s}_{\leq \mathsf{q}} \rfloor^{\mathsf{rig}}$. A sequence is totally rigid if it is both player and opponent rigid.

Definition 146 We define the referee $\mathcal{R}^p$ so that for any arena A we have $\mathsf{s} \in \mathcal{R}^p_{\mathsf{A}}$ if and only if s is player rigid.

We define the referee $\mathcal{R}^o$ so that for any arena A we have $\mathsf{s} \in \mathcal{R}^o_{\mathsf{A}}$ if and only if s is opponent rigid.

We define the referee $\mathcal{R}$ so that for any arena A we have $\mathsf{s} \in \mathcal{R}_{\mathsf{A}}$ if and only if s is totally rigid.

Definition 147 We define the sequence transformer $R : \mathcal{S}$ such that for each arrow arena A and each sequence $\mathsf{s} = \langle \mathsf{M}_{\mathsf{s}}, \prec_{\mathsf{s}}, \curvearrowright_{\mathsf{s}} \rangle$ in $\mathcal{S}_{\mathsf{A}}$ we define:

$$B_{\mathsf{A}}\mathsf{s} = \langle \mathsf{M}_{\mathsf{s}}, \prec_{\mathsf{s}}, \{(\mathsf{m}, \mathsf{m}') \in \curvearrowright_{\mathsf{s}} |\ \lambda^{QA}_{\mathsf{A}}\mathsf{s} = Q\} \rangle.$$

In other words we have the same sequence of moves but only include the justifiers of questions.

Lemma 148 Given a sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ it follows that $\mathsf{s} \in \mathcal{R}^p_{\mathsf{A}}$ if and only if $R\mathsf{s}$ is player visible and also $\mathsf{s} \in \mathcal{R}_{\mathsf{A}}$ if and only if $R\mathsf{s}$ is totally visible.

Proof The proof follows directly from inspection of the definition of R. ■

Lemma 149 The referees $\mathcal{V}$ and $\mathcal{V}^p$ are admissible.

Proof The referee $\mathcal{V}$ is admissible by lemmas 148 and 125. The referee $\mathcal{V}^p$ is admissible by lemmas 148 and 126. ■

3.6 Coherence Constraints on Strategies

We will now study some further constraints on strategies. Unlike those constraints placed by referees, which determine which sequences may exist in a strategy, the constraints in this section determine which sequences may, or must, coexist in a strategy. Such constraints have been termed coherence constraints in personal communication from Russ Harmer via Guy McCusker.

3.6.1 Prefix Closure and Determinism

The determinism constraint is present in the PCF model of [28]. The relaxation of this constraint was studied in depth by Russ Harmer in [24] and by Harmer and McCusker [25] and was found to coincide with the inclusion of a nondeterministic choice operator into the language.

Definition 150 As with strategies for games, a strategy σ for arena A is prefix-closed if and only if it is not empty and for any sequence $s \cdot m \cdot m' \in \sigma$ we have $s \in \sigma$. All prefix-closed strategies contain the empty sequence ε.

Definition 151 As with strategies for games, a prefix-closed strategy σ is deterministic if and only if for all sequences $s \cdot m, s' \cdot m' \in \sigma$ we have $s = s' \Rightarrow s \cdot m = s' \cdot m'$.

Lemma 152 Given a compositional, copy complete referee $\mathcal{R}$ then any subcategory $\mathbf{C}$ of $\mathbf{A}_{\mathcal{R}}$ can be further constrained to contain only prefix closed strategies. The resultant collection of arenas and strategies constitute a lluf subcategory of $\mathbf{C}$.

Proof Lemma 35 ensures that prefix-closure is preserved by composition. For any negative arena A the underlying game $U_{\mathcal{R}}A$ is a prefix game as $\mathcal{R}_A$ is prefix closed. Hence, by lemma 34, the identity $\mathbf{id}_{U_{\mathcal{R}}A}$ is prefix-closed. ■

Lemma 153 Given a compositional, copy complete referee $\mathcal{R}$ and any subcategory $\mathbf{C}$ of $\mathbf{A}_{\mathcal{R}}$ containing only prefix-closed strategies it follows that $\mathbf{C}$ can be further constrained to contain only the deterministic strategies. The resultant collection of arenas and strategies constitute a lluf subcategory of $\mathbf{C}$.

Proof Lemma 42 ensures that determinism is preserved under composition. For any negative arena A the underlying game $U_{\mathcal{R}}A$ is a prefix game as $\mathcal{R}_A$ is prefix closed. Hence, by lemma 38, the identity $\mathbf{id}_{U_{\mathcal{R}}A}$ is deterministic. ■

3.7 Umpires

3.7.1 Umpire Observance

Definition 154 Given a referee $\mathcal{R}$ an umpire $\simeq$: $\mathcal{R}$ is a family of equivalence relations, such that for each arrow arena A we have an equivalence relation

$$\simeq_A \subseteq \mathcal{R}_A \times \mathcal{R}_A.$$

We will omit subscripts from our umpires when they can be readily inferred from the context.

Definition 155 Given a referee $\mathcal{R}$ and an umpire $\simeq$: $\mathcal{R}$ we say that a deterministic strategy $\sigma \subset \mathcal{R}_A$ is $\simeq$ observant with respect to $\mathcal{R}$ if and only if for all sequences $s \cdot m \cdot n, s' \in \sigma$ such that there exists a sequence $s' \cdot m' \in \mathcal{R}_A$ for which $s \cdot m \simeq s' \cdot m'$ then there must exist some extension $s' \cdot m' \cdot n' \in \sigma$ such that $s' \cdot m' \cdot n' \simeq s \cdot m \cdot n$.

Definition 156 We say that an umpire $\simeq$: $\mathcal{R}$ is compositional if and only if for any $\simeq$ observant strategies $\sigma \subseteq \mathcal{R}_{A \to B}$ and $\tau \subseteq \mathcal{R}_{B \to C}$ it follows that $\sigma;\tau$ is also $\simeq$ observant.

Definition 157 We say that an umpire $\simeq$: $\mathcal{R}$ is copy sufficient if and only if for every arena A it follows that $\mathbf{copy}_A^{\mathcal{R}}$ is $\simeq$ observant.

Theorem 158 Given a compositional, copy complete referee $\mathcal{R}$ and a compositional, copy sufficient umpire $\simeq$: $\mathcal{R}$ it is straightforward to show that the $\simeq$ observant strategies constitute a lluf subcategory of $\mathbf{A}_{\mathcal{R}}$.

3.7.2 Umpires and Sequence Transformers

We will now show how we can sometimes use sequence transformers to reduce the proof that a particular umpire is compositional and copy sufficient to the proofs in **G** that innocence is preserved by composition and that identities are innocent.

Lemma 159 Given a biased referee $\mathcal{R}$, an umpire $\simeq$: $\mathcal{R}$ and an injective sequence transformer $K : \mathcal{R}$ such that for any arrow arena $A \to B$ and sequences $s, s' \in U_{\mathcal{R}}A \to U_{\mathcal{R}}B$ it follows that:

- Ks is player visible if $s \in \mathcal{R}_{A \to B}$.
- $s \simeq s' \Leftrightarrow \lceil Ks \rceil = \lceil Ks' \rceil$.

then for any strategy $\sigma \subseteq \mathcal{R}_{A \to B}$ it follows that σ is $\simeq$ observant if and only if $K\sigma$ is innocent.

Proof First let us suppose that σ is $\simeq$ observant. Suppose we have sequences $K(s \cdot m \cdot n), Ks' \in K\sigma$ and suppose there exists an extension $K(s' \cdot m') \in \mathcal{P}_{KA \to KB}$ therefore by bias and injectivity we have the extension $s' \cdot m' \in \mathcal{R}_{A \to B}$ and if $\lceil K(s' \cdot m') \rceil = \lceil K(s \cdot m) \rceil$ then $s' \cdot m' \simeq s \cdot m$ and hence by $\simeq$ observance we have a sequence $s' \cdot m' \cdot n' \in \sigma$ such that $s' \cdot m' \cdot n' \simeq s \cdot m \cdot n$ and hence there exists a $K(s' \cdot m' \cdot n') \in K\sigma$ such that $\lceil K(s' \cdot m' \cdot n') \rceil = \lceil K(s \cdot m \cdot n) \rceil$ and hence $K\sigma$ is innocent.

Now let us suppose that $K\sigma$ is innocent. Suppose we have sequences $s \cdot m \cdot n, s' \in \sigma$ and hence $K(s \cdot m \cdot n), Ks' \in K\sigma$. Now suppose there exists a sequence $s' \cdot m' \in \mathcal{R}_{A \to B}$ such that $s \cdot m \simeq s' \cdot m'$. Therefore, as K respects prefixes it must be that $K(s' \cdot m')$ is an extension of Ks' and $\lceil K(s \cdot m) \rceil = \lceil K(s' \cdot m') \rceil$ hence by the innocence of $K\sigma$ we must have some extension $K(s' \cdot m') \cdot n' \in K\sigma$ such that $\lceil K(s' \cdot m') \cdot n' \rceil = \lceil K(s \cdot m \cdot n) \rceil$. By injectivity and prefix respect it follows that $K(s' \cdot m') \cdot n' = K(s' \cdot m' \cdot n')$ for some extension of $s' \cdot m'$. Hence $s' \cdot m' \cdot n' \in \sigma$ and σ is $\simeq$ observant. ■

Lemma 160 Given an admissible referee $\mathcal{R}$, an umpire $\simeq$: $\mathcal{R}$ and an injective sequence transformer $K : \mathcal{R}$ such that for any arena A and sequences $s, s' \in U_{\mathcal{R}}A \to U_{\mathcal{R}}B$ it follows that

- Ks is player visible if $s \in \mathcal{R}_{A \to B}$.

- $s \simeq s' \Leftrightarrow \lceil Ks \rceil = \lceil Ks' \rceil$

then $\simeq$ is compositional and copy sufficient.

Proof Suppose we are given $\simeq$ observant strategies $\sigma \subseteq \mathcal{R}_{A \to B}$ and $\tau \subseteq \mathcal{R}_{B \to C}$. Lemma 159 guarantees that $K\sigma$ and $K\tau$ are innocent and hence by lemma 81 it follows that $K\sigma; K\tau$ is innocent. We know from the compositionality of $\mathcal{R}$ that $\sigma;\tau \in \mathcal{R}_{A \to C}$ so we can now apply lemma 124 to see that $K(\sigma;\tau)$ is innocent and hence $\sigma;\tau$ is $\simeq$ observant and thus $\simeq$ is compositional.

Furthermore, we know from lemma 74 that $\mathbf{id}_{KA}$ is innocent and by lemma 122 we know that $\mathbf{id}_{KA} = K\mathbf{copy}^{\mathcal{R}}_{A}$ and hence by lemma 159 we know that $\mathbf{copy}^{\mathcal{R}}_{A}$ is $\simeq$ observant. ■

3.7.3 Innocence

The property of innocence is present in the PCF games model of [28]. A study of the consequences of its relaxation is studied by Abramsky and McCusker [6] and is shown to be equivalent to the introduction of references of base type. We will study their model more closely in the next chapter.

Our definition of innocence for arenas is dependent on what moves it is possible for opponent to make: innocence is defined with respect to a referee. We therefore give the following definition.

Definition 161 Given a referee $\mathcal{R}$ we say that a deterministic strategy $\sigma \subset \mathcal{R}_A$ is innocent with respect to $\mathcal{R}$ if and only if for all sequences $s \cdot m \cdot n, s' \in \sigma$ such that there exists a sequence $s' \cdot m' \in \mathcal{R}_A$ for which $\lceil s \cdot m \rceil = \lceil s' \cdot m' \rceil$ it follows that there must exist some extension $s' \cdot m' \cdot n' \in \sigma$ such that $\lceil s' \cdot m' \cdot n' \rceil = \lceil s \cdot m \cdot n \rceil$.

Definition 162 Given a referee $\mathcal{R}$ we define the umpire $\simeq_p$: $\mathcal{R}$ such that $s \simeq_p s'$ if and only if $\lceil s \rceil = \lceil s' \rceil$. Clearly innocence is equivalent to $\simeq_p$ observance.

Theorem 163 The given an admissible referee $\mathcal{R}$ such that $\mathcal{R} \subseteq \mathcal{V}_p$ it follows that the umpire $\simeq_p$: $\mathcal{R}$ is compositional and copy sufficient.

Proof Recall the identity sequence transformer of definition 132. Clearly it is injective. For any arena A and sequences $s, s' \in A$ we clearly have:

$$(s \simeq_p s') \Leftrightarrow (\lceil Is \rceil = \lceil Is' \rceil).$$

Hence by lemma 160 it follows that $\simeq_p$: $\mathcal{R}$ is compositional and copy sufficient. ■

As a corollary we know by proposition 158 that the innocent strategies form a lluf subcategory of $\mathbf{A}_{\mathcal{R}}$.

3.7.4 Proper Stranding and Thread Independence

Proper stranding is implicit in the games models in [28, 39, 4, 5, 6, 1, 9, 31] and as we will later show is implied by total visibility.

Definition 164 (Strands) Given a sequence $s \in \mathcal{S}_A$ we say that a *strand* of s is a subsequence of s containing exactly those moves that are hereditarily justified by a particular initial move.

Given a justified sequence $s \cdot m \in \mathcal{S}_A$, we write $\mathbf{strand}(s \cdot m)$ as that strand of $s \cdot m$ whose moves are hereditarily justified by the same initial move as m. For the sake of completeness we write $\mathbf{strand}(\varepsilon) = \varepsilon$.

Definition 165 (Proper Stranding) We say that s is *properly stranded* if all its constituent strands obey alternation.

Definition 166 We can define the referee $\mathcal{Z}$ such that for each arrow arena A and sequence $s \in \mathcal{S}_A$ it follows that

$$s \in \mathcal{Z}_A \text{ if and only if } s \text{ is properly stranded.}$$

Lemma 167 The referee $\mathcal{Z}$ is separable, hence admissible.

Proof Proof is by inspection of the definition. ■

We will now examine proper stranding a little more thoroughly and will eventually prove that it is implicitly satisfied by sequences that respect total visibility.

Lemma 168 (Switching Conditions) A sequence $s \in \mathcal{S}_{A \to B}$ is properly stranded if and only if for any moves $m, m' \in s$ such that m and m' are in different strands we have:

- If m and m' are consecutive in $s \upharpoonright B$ then m' is an opponent move.
- If m and m' are consecutive in $s \upharpoonright A$ then m' is a player move.

Proof We first assume that s is properly stranded and that m and m' are consecutive in $s \upharpoonright B$. We construct a proof that m' is an opponent move by induction on the length of $s_{\leq m'}$.

Base Case If $s_{\leq m'} = \varepsilon$ then the lemma is vacuously satisfied.

Inductive Step If m' is initial then it must be an opponent move and we have nothing more to prove. Otherwise, let the greatest (with respect to $<$) move in $s_{\leq m}$ that is in $\mathbf{strand}(s_{\leq m'})$ be n. We do not have $n = m$ and therefore know that the move following n in $s \upharpoonright B$ is in a different strand and hence, by inductive hypothesis, is an opponent move and thus n must be a player move. As s is properly stranded we therefore have m' an opponent move.

If we assume that s is properly stranded and that m and m' are consecutive in $s \upharpoonright A$ we can use a similar argument to show that m' is a player move.

We now prove the implication in the other direction:

- Let m and m' be opponent moves from the same strand in $s \upharpoonright B$. We will show that they are not consecutive in that strand. Let m^+ immediately follow m in $s \upharpoonright B$ and we know this is a player move as $s \upharpoonright B$ alternates and hence by hypothesis it follows that m^+ is in the same strand as m.

- Now m and m' be player moves from the same strand in $s \upharpoonright B$. We will show that they are not consecutive in that strand. Let m'^{-} immediately precede m' in $s \upharpoonright B$ and we know this is an opponent move as $s \upharpoonright B$ alternates and hence by hypothesis it follows that $m'-$ is in the same strand as m'.

Our proof in this direction is similar when we consider moves from A. ■

Lemma 169 Given a player visible sequence $s \in \mathcal{S}_{A \to B}$ with moves $m, m' \in s \upharpoonright B$ such that $\lceil s_{\leq m} \rceil$ and $\lceil s_{\leq m'} \rceil$ commence with the same move we have m and m' in the same strand.

Proof Exactly one initial opponent move hereditarily justifies m, by player visibility that move must be in $\lceil s_{\leq m} \rceil$ and by the definition of $\lceil - \rceil$ it must be the first move in the view. Similarly $\lceil s_{\leq m'} \rceil$ commences with the initial opponent move that hereditarily justifies m'. If these moves are the same then m and m' are in the same strand by definition. ■

Lemma 170 Given a player visible sequence $s \in \mathcal{S}_{A \to B}$ and moves $m, m' \in s$ such that m and m' are in the same strand it follows that $\lceil s_{\leq m} \rceil$ and $\lceil s_{\leq m'} \rceil$ commence with the same move.

Proof Let move k be the initial move that hereditarily justifies m and m'. By player visibility we have both $k \in \lceil s_{\leq m} \rceil$ and $k \in \lceil s_{\leq m'} \rceil$ and hence by inspection of the definition of $\lceil - \rceil$ it must be that $\lceil s_{\leq m} \rceil$ and $\lceil s_{\leq m'} \rceil$ commence with the same move. ■

Lemma 171 Given a player visible, properly stranded sequence $s \in \mathcal{S}_{A \to B}$ and moves $m, m' \in s$ such that all moves strictly between m and m' are in $s \upharpoonright A$ it follows that $\lceil s_{\leq m} \rceil$ and $\lceil s_{\leq m'} \rceil$ commence with the same move.

Proof We will simply show that for any consecutive pair of moves $n, n^+ \in s$ such that $m \leq n$ and $n^+ \leq m'$ it must be that $\lceil s_{\leq n} \rceil$ and $\lceil s_{\leq n^+} \rceil$ commence with the same move. If n^+ is a player move then this follows directly from the definition of $\lceil - \rceil$. Otherwise, if n^+ is an opponent move, by the switching condition we must have $n, n^+ \in s \upharpoonright A$. By lemma 168 we have n and n^+ in the same strand and by lemma 170 we know that $\lceil s_{\leq n} \rceil$ and $\lceil s_{\leq n^+} \rceil$ commence with the same move. ■

Lemma 172 Given a sequence $s \in \mathcal{V}_{A \to B}$ we have s properly stranded.

Proof Our proof is by induction on the length of s.

Base Case If $s = \varepsilon$ then s the lemma is trivial.

Inductive Step Suppose $s = s' \cdot m$. We have s' properly stranded by inductive hypothesis. If m is initial then it follows that $s' \cdot m$ is properly stranded. Otherwise, let n be the greatest move in s' from the same strand as m and we must simply show that $\lambda^{OP} n \neq \lambda^{OP} m$.

case: Suppose m is a player move in $s \upharpoonright A$. If n is the last move in $s' \upharpoonright A$ then n must be an opponent move. Otherwise let n' immediately follow n in $s' \upharpoonright A$. By inductive hypothesis we have $s'_{\leq n'}$ properly stranded and by lemma 168 we know that n' is a player move hence n must be an opponent move.

case: Suppose m is an opponent move in $s \upharpoonright A$. Consider the form of $\lfloor s' \rfloor$ as follows:

$$p^0 \cdot o^1 \overset{\curvearrowleft}{\cdot} p^1 \cdots o^n \overset{\curvearrowleft}{\cdot} p^n$$

All moves in this view are from $s \upharpoonright A$ by the switching condition and by inductive hypothesis we have s' properly stranded hence by lemma 168 we have every move in this sequence from the same strand. By opponent visibility we have the justifier of m in this sequence, hence m is in the same strand as the last move in s' and we have proper stranding.

case: If m is an opponent move in $s \upharpoonright B$ then we can simply apply lemma 168 to show proper stranding.

case: Finally, suppose m is a player move in $s \upharpoonright B$. If n' is the last move in $s' \upharpoonright B$ then by lemma 171 we have $\lceil s'_{\leq n} \rceil$ and $\lceil s' \rceil$ start with the same opponent move and by lemma 169 we have n' and m in the same strand and hence we have proper stranding. ■

Thread independence, or single threading as it is sometimes known, is often explicitly required in games models where the objects are arenas, for example in [1, 24]. Consideration of thread independence may be avoided in categories of games, for example in[28, 39, 4, 5, 6] by utilising the linear nature of the models and interpreting the languages in Kleisli categories of games where plays may contain only a single thread.

We will now prove that thread-independence is preserved by composition. This proof is considerably more involved and we only prove it for strategies that respect total visibility. It is not clear how a proof of this nature can be extended to a more general form; for example the thread-independence for the model of a language with general references in [1] requires a different kind of proof.

Definition 173 (Threads) Given a sequence $s \cdot m$, we define **thread**$(s \cdot m)$ to be that subsequence of s consisting only of those moves m' such that $\lceil s_{\leq m'} \rceil$ commences with the same move as does $\lceil s \cdot m \rceil$. For completeness' sake we define **thread**$(\varepsilon) = \varepsilon$. We say that a sequence is single-threaded if and only if it contains only one initial opponent move.

Definition 174 (Thread Independence) Given a referee $\mathcal{R}$ we say that a deterministic strategy $\sigma \subset \mathcal{R}_A$ is thread independent with respect to $\mathcal{R}$ if and only if for all sequences $s \cdot m \cdot n, s' \in \sigma$ such that there exists a sequence $s' \cdot m' \in \mathcal{R}_A$ for which **thread**$(s \cdot m) =$ **thread**$(s' \cdot m')$ then there must exist some extension $s' \cdot m' \cdot n' \in \sigma$ such that **thread**$(s' \cdot m' \cdot n') =$ **thread**$(s \cdot m \cdot n)$.

Definition 175 Given a sequence $s \in \mathcal{V}_A$ we define the relation $\ll_s$ between moves in s as follows. Given $m, m' \in s$ we say $m \ll_s m'$ if and only if m is the greatest move in $s_{<m'}$ from the same strand as m'.

It is immediately apparent that for any non-initial move $n' \in s$ there is exactly one move $n \in s$ such that $n \ll_s n'$.

We know from lemma 172 that s is properly stranded and hence for any $m, m' \in s$ such that $m \ll_s m'$ we have $\lambda^{OP} n \neq \lambda^{OP} n'$.

Definition 176 We define a sequence transformer $F : \mathcal{V}$ as follows. Given arrow arena A and sequence $s = \langle M_s, \lhd_s, \curvearrowleft_s \rangle$ in $\mathcal{V}_A$ then

$$Fs = \langle M_s, \lhd_s, \ll_s \rangle.$$

Lemma 177 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ such that $F\mathsf{s}$ respects opponent visibility and moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s} \upharpoonright \mathsf{B}$ where $\mathsf{m} \ll \mathsf{m}'$ and m is a player move in s we have all moves inclusively between m and m' are from $\mathsf{s} \upharpoonright \mathsf{B}$.

Proof As $F\mathsf{s}$ respects opponent visibility it must be that $\lfloor F\mathsf{s}_{\leq \mathsf{m}'} \rfloor$ has the following form:

$$\mathsf{m} \cdot \mathsf{o}^1 \overset{\frown}{\cdot} \mathsf{p}^1 \cdots \mathsf{o}^n \overset{\frown}{\cdot} \mathsf{p}^n \cdot \mathsf{m}'.$$

The switching condition implies that for all $1 \leq i \leq n$ we have o_i and p_i in $\mathsf{s} \upharpoonright \mathsf{B}$ and lemma 168 implies that o_i and p_i are consecutive moves. Hence all moves inclusively between m and m' are from $\mathsf{s} \upharpoonright \mathsf{B}$. ■

Lemma 178 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ such that $F\mathsf{s}$ respects total visibility then for any moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ we have

$$\mathsf{m} \in \mathbf{strand}(\mathsf{s}_{\leq \mathsf{m}'}) \Rightarrow \mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{m}'} \rceil \wedge \mathsf{m} \in \lfloor F\mathsf{s}_{\leq \mathsf{m}'} \rfloor.$$

Proof We carry out an induction on the length of s. We consider here only the case where m' is a player move as the other case is similar.

Base Case If $\mathsf{s} = \varepsilon$ then the lemma is trivial.

Inductive Step First note that $\mathsf{m} \in \mathbf{strand}(\mathsf{s}_{\leq \mathsf{m}'})$ implies that either $\mathsf{m} = \mathsf{m}'$ and the lemma is trivial or else there exists a move $\mathsf{k} \in \mathsf{s}$ such that $\mathsf{m} \in \mathbf{strand}(\mathsf{s}_{\leq \mathsf{k}})$ and $\mathsf{k} \ll \mathsf{m}'$. By inductive hypothesis we have $\mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{k}} \rceil$ and $\mathsf{m} \in \lfloor F\mathsf{s}_{\leq \mathsf{k}} \rfloor$ and by visibility we have $\mathsf{k} \in \lceil F\mathsf{s}_{\leq \mathsf{m}'} \rceil$ and by construction we have $\mathsf{k} \in \lfloor F\mathsf{s}_{\leq \mathsf{m}'} \rfloor$. Therefore by lemma 53 we have $\mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{m}'} \rceil$ and $\mathsf{m} \in \lfloor F\mathsf{s}_{\leq \mathsf{m}'} \rfloor$. ■

As a corollary we have

$$\mathsf{m} \curvearrowright_{\mathsf{s}} \mathsf{m}' \Rightarrow \mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{m}'} \rceil \wedge \mathsf{m} \in \lfloor F\mathsf{s}_{\leq \mathsf{m}'} \rfloor$$

and consequently $\forall \mathsf{k} \in \lceil \mathsf{s}_{\mathsf{m}'} \rceil . \mathsf{k} \in \lceil F\mathsf{s}_{\leq \mathsf{m}'} \rceil$ and $\forall \mathsf{k} \in \lfloor \mathsf{s}_{\mathsf{m}'} \rfloor . \mathsf{k} \in \lfloor F\mathsf{s}_{\leq \mathsf{m}'} \rfloor$.

Lemma 179 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ it follows that $F\mathsf{s}$ respects total visibility.

Proof Our proof is my induction on the length of s.

Base Case If $\mathsf{s} = \varepsilon$ then $F\mathsf{s}$ trivially respects total visibility.

Inductive Step Suppose $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$. By inductive hypothesis we assume $F\mathsf{s}'$ respects total visibility. Consider any move j in $F\mathsf{s}$ such that $\mathsf{j} \curvearrowright_{F\mathsf{s}} \mathsf{m}$. If we also have $\mathsf{j} \curvearrowright_{\mathsf{s}} \mathsf{m}$ then we have $\mathsf{j} \in \lceil \mathsf{s} \rceil$ from the definition of F we must have $\mathsf{j} \ll_{\mathsf{s}} \mathsf{m}$ and we consider the following cases.

case: Suppose m is an opponent move in $\mathsf{s} \upharpoonright \mathsf{A}$. By lemma 168 we must have j the immediate predecessor of m hence $\mathsf{j} \in \lfloor F\mathsf{s} \rfloor$.

case: Now suppose m is an opponent move in $\mathsf{s} \upharpoonright \mathsf{B}$. We note that for any pair of moves n, n' that are consecutive in $F\mathsf{s} \upharpoonright \mathsf{B}$ we have $\mathsf{j} \in \lfloor F\mathsf{s}_{\leq \mathsf{n}} \rfloor \Rightarrow \mathsf{j} \in \lfloor F\mathsf{s}_{\leq \mathsf{n}'} \rfloor$ by the definition of $\lfloor - \rfloor$ when n' is an opponent move and from lemma 172 when n' is an player move. We therefore have $\mathsf{j} \in \lfloor F\mathsf{s} \rfloor$.

case: Now suppose m is a player move in $\mathsf{s} \upharpoonright \mathsf{A}$. We know that $\lceil F\mathsf{s} \rceil$ is of the following from:

$$\mathsf{o}^0 \cdot \mathsf{p}^1 \overset{\frown}{\cdot} \mathsf{o}^1 \cdots \mathsf{p}^n \overset{\frown}{\cdot} \mathsf{o}^n \cdot \mathsf{m}$$

where o^0 is initial. Consider any pair of moves p_i and o_i. By lemma 168 we know that if p_i and o_i are from $s \upharpoonright A$ then they must be consecutive. By inductive hypothesis Fs' respects total visibility so by lemma 177 we know that if p_i and o_i are from $s \upharpoonright B$ then all moves between p_i and o_i are also from $s \upharpoonright B$. We therefore know that j cannot fall exclusively between any moves in $\lceil Fs \rceil$. By lemma 178 we know that the unique initial opponent move in $\lceil s' \rceil$ must also be the unique initial opponent move in $\lceil Fs' \rceil$ and by the visibility of s we have $o^0 \leq j$ and hence $j \in \lceil Fs \rceil$.

case: Finally suppose m is a player move in $s \upharpoonright B$. By lemma 172 we know that j is the immediate predecessor of m in $s \upharpoonright B$. If there are no moves from A between j and m then we trivially have $j \in \lceil Fs \rceil$. Otherwise we note that for any adjacent pair of moves n, n^+ such that $j \leq n$ and $n^+ \leq m$ we have have $j \in \lceil Fs_{\leq n} \rceil \Rightarrow j \in \lceil Fs_{\leq n^+} \rceil$ this follows from the definition of $\lceil - \rceil$ when n^+ is a player move and from lemma 172 when n^+ is an opponent move. We therefore have $j \in \lceil Fs \rceil$. ■

Lemma 180 Given a non-empty total visibility sequence $s \in \mathcal{V}_{A \to B}$ there is exactly one initial opponent move in $\lceil Fs \rceil$.

Proof The proof is a simple induction on the length of s. ■

Lemma 181 Given a sequence $s \in \mathcal{V}_{A \to B}$ and moves $m, m', m'' \in s$ we have

$$m \in \lceil Fs_{\leq m''} \rceil \wedge m' \in \lceil Fs_{\leq m''} \rceil \Rightarrow m \in \lceil Fs_{\leq m'} \rceil \vee m' \in \lceil Fs_{\leq m} \rceil.$$

Proof We assume without loss of generality that $m \leq m'$ and prove that $m \in \lceil Fs_{\leq m''} \rceil$ by induction on the length of $s_{\leq m''}$.

Base Case If $s_{\leq m''} = \varepsilon$ then the proof is trivial.

Inductive Step

case: Suppose m'' is a player move. If $m' \in \lceil Fs_{\leq m''} \rceil$ then either $m' = m''$ and we have nothing to prove or else $m' \in \lceil Fs_{< m''} \rceil$ in which case we must also have $m \in \lceil Fs_{< m''} \rceil$ as $m \leq m'$ and we can apply our inductive hypothesis to give $m \in \lceil Fs_{\leq m''} \rceil$.

case: Now suppose m'' is an opponent move. If $m' \in \lceil Fs_{\leq m''} \rceil$ then either $m' = m''$ and again we have nothing to prove or else there is some move j such that $j \curvearrowleft_{Fs} m''$ and $m' \in \lceil Fs_{\leq j} \rceil$ in which case we must also have $m \in \lceil Fs_{< j} \rceil$ as $m \leq m'$ and moves in Fs have at most one justifier. Hence by inductive hypothesis we have $m \in \lceil Fs_{\leq m''} \rceil$. ■

Definition 182 We now define a similar sequence transformer $T : \mathcal{V}$ such that for any arena A and any sequence $s = \langle M_s, \lhd_s, \curvearrowleft_s \rangle$ in $\mathcal{V}_A$ we have:

$$Ts = \langle M_s, \lhd_s, \curvearrowleft_s \cup \ll_s \rangle.$$

Lemma 183 The sequence transformer $T : \mathcal{V}$ is injective.

Proof Given any sequence $s \in T\mathcal{V}_A$ we can create a sequence $T^{-1}s \in \mathcal{V}_A$ by removing any

justifier between moves $\mathsf{j}, \mathsf{m} \in \mathsf{s}$ when there exists another justifier for m in $\mathsf{s}_{<\mathsf{j}}$. It is simple to check that $T^{-1}\mathsf{s}$ is well-defined and is indeed the inverse. ■

As a corollary for justified sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{V}_{\mathsf{A}}$ we have:

$$\mathsf{s} = \mathsf{s}' \Leftrightarrow T\mathsf{s} = T\mathsf{s}'.$$

Lemma 184 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ for all $\mathsf{m}, \in \mathsf{s}$ we have:

- $\mathsf{m} \in \lceil \mathsf{s} \rceil \Rightarrow \mathsf{m} \in \lceil T\mathsf{s} \rceil$.
- $\mathsf{m} \in \lceil F\mathsf{s} \rceil \Leftrightarrow \mathsf{m} \in \lceil T\mathsf{s} \rceil$.

Proof The proof is by induction on the length of s. The proof of

$$\mathsf{m} \in \lceil \mathsf{s} \rceil \Rightarrow \mathsf{m} \in \lceil T\mathsf{s} \rceil \text{ and } \mathsf{m} \in \lceil F\mathsf{s} \rceil \Rightarrow \mathsf{m} \in \lceil T\mathsf{s} \rceil$$

is very straightforward. We consider here only the proof of $\mathsf{m} \in \lceil T\mathsf{s} \rceil \Rightarrow \mathsf{m} \in \lceil F\mathsf{s} \rceil$.

Base Case If $\mathsf{s} = \varepsilon$ then the proof is trivial.

Inductive Step Suppose $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}'$.

case: If m' is a player move then $\mathsf{m} \in \lceil T\mathsf{s} \rceil$ implies either $\mathsf{m} = \mathsf{m}'$ and therefore $\mathsf{m} \in \lceil F\mathsf{s} \rceil$ or else $\mathsf{m} \in \lceil T\mathsf{s}' \rceil$ and by inductive hypothesis $\mathsf{m} \in \lceil F\mathsf{s}' \rceil$ and $\mathsf{m} \in \lceil F\mathsf{s} \rceil$ by the definition of $\lceil - \rceil$.

case: If m' is an opponent move then $\mathsf{m} \in \lceil T\mathsf{s} \rceil$ implies either $\mathsf{m} = \mathsf{m}'$ and once again $\mathsf{m} \in \lceil F\mathsf{s} \rceil$ or else there exists some move $\mathsf{j} \in \mathsf{s}'$ such that $\mathsf{j} \curvearrowleft_{T\mathsf{s}} \mathsf{m}'$ and $\mathsf{m} \in \lceil T\mathsf{s}_{\leq \mathsf{j}} \rceil$. We can apply the inductive hypothesis to yield $\mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{j}} \rceil$ and by the definition of T we either have $\mathsf{j} \curvearrowleft_{F\mathsf{s}} \mathsf{m}'$ and hence $\mathsf{m} \in \lceil F\mathsf{s} \rceil$ by the definition of $\lceil - \rceil$ or else $\mathsf{j} \curvearrowleft_{\mathsf{s}} \mathsf{m}'$ in which case we have $\mathsf{j} \in \lceil F\mathsf{s} \rceil$ by lemma 178 and $\mathsf{m} \in \lceil F\mathsf{s} \rceil$ by lemma 53. ■

Lemma 185 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ it follows that $T\mathsf{s}$ respects total visibility.

Proof Given any moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that $\mathsf{m} \curvearrowleft_{T\mathsf{s}} \mathsf{m}'$ we have either $\mathsf{m} \curvearrowleft_{\mathsf{s}} \mathsf{m}'$ or else $\mathsf{m} \curvearrowleft_{F\mathsf{s}} \mathsf{m}'$ by the definition of T and hence $\mathsf{m} \in \lceil T\mathsf{s}_{\leq \mathsf{m}'} \rceil$ by lemma 184. We lso have $\mathsf{m} \in \lfloor T\mathsf{s}_{\leq \mathsf{m}'} \rfloor$ as either m is the predecessor of m' or else $\mathsf{m} \curvearrowleft_{\mathsf{s}} \mathsf{m}'$. ■

Lemma 186 Given a non-empty sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ there is exactly one initial opponent move in $\lceil T\mathsf{s} \rceil$.

Proof The proof is by induction on the length of s. First we note that an opponent move in s is initial if and only if it is initial in $T\mathsf{s}$.

Base Case If $\mathsf{s} = \varepsilon$ then the proof is trivial.

Inductive Step Suppose $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$.

case: If m is a player move then given any initial opponent moves $\mathsf{n}, \mathsf{n}' \in \mathsf{s}$ such that $\mathsf{n}, \mathsf{n}' \in \lceil T\mathsf{s}' \cdot \mathsf{m} \rceil$ we must have $\mathsf{n}, \mathsf{n}' \in \lceil T\mathsf{s}' \rceil$ and we can apply the inductive hypothesis to yield $\mathsf{n} = \mathsf{n}'$.

case: If m is an initial opponent move then given any initial opponent moves $\mathsf{n}, \mathsf{n}' \in \mathsf{s}$ such that $\mathsf{n}, \mathsf{n}' \in \lceil T\mathsf{s}' \cdot \mathsf{m} \rceil$ we must have $\mathsf{m} = \mathsf{n} = \mathsf{n}'$ by the definition of $\lceil - \rceil$.

case: If m is a non-initial opponent move then given any initial opponent moves $\mathsf{n}, \mathsf{n}' \in \mathsf{s}$ such that $\mathsf{n}, \mathsf{n}' \in \lceil T\mathsf{s}' \cdot \mathsf{m} \rceil$ we must have some move j such that $\mathsf{j} \curvearrowleft_{T\mathsf{s}} \mathsf{m}$ and $\mathsf{n}, \mathsf{n}' \in \lceil T\mathsf{s}'_{\leq \mathsf{j}} \rceil$ in which case we apply the inductive hypothesis to $\mathsf{s}'_{\leq \mathsf{j}}$ to yield $\mathsf{n} = \mathsf{n}'$. ■

Lemma 187 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ and moves $\mathsf{m}, \mathsf{m}', \mathsf{m}'' \in \mathsf{s}$ we have

$$\mathsf{m} \in \lceil T\mathsf{s}_{\leq \mathsf{m}''} \rceil \wedge \mathsf{m}' \in \lceil T\mathsf{s}_{\leq \mathsf{m}''} \rceil \Rightarrow \mathsf{m} \in \lceil T\mathsf{s}_{\leq \mathsf{m}'} \rceil \vee \mathsf{m}' \in \lceil T\mathsf{s}_{\leq \mathsf{m}} \rceil.$$

Proof If $\mathsf{m} \in \lceil T\mathsf{s}_{\leq \mathsf{m}''} \rceil \wedge \mathsf{m}' \in \lceil T\mathsf{s}_{\leq \mathsf{m}''} \rceil$ then by lemma 184 we have $\mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{m}''} \rceil \wedge \mathsf{m}' \in \lceil F\mathsf{s}_{\leq \mathsf{m}''} \rceil$ and by lemma 181 we have $\mathsf{m} \in \lceil F\mathsf{s}_{\leq \mathsf{m}'} \rceil \vee \mathsf{m}' \in \lceil F\mathsf{s}_{\leq \mathsf{m}} \rceil$ and hence $\mathsf{m} \in \lceil T\mathsf{s}_{\leq \mathsf{m}'} \rceil \vee \mathsf{m}' \in \lceil T\mathsf{s}_{\leq \mathsf{m}} \rceil$ by lemma 184. ■

Lemma 188 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ and move $\mathsf{m} \in \mathsf{s}$ we have

$$\mathsf{m} \in \lceil T\mathsf{s} \rceil \Leftrightarrow \mathsf{m} \in \mathbf{thread}(\mathsf{s})$$

Proof First we assume that $\mathsf{m} \in \lceil T\mathsf{s} \rceil$ and show that $\mathsf{m} \in \mathbf{thread}(\mathsf{s})$. Let n be the initial opponent move in $\lceil \mathsf{s} \rceil$ and let n' be the initial opponent move in $\lceil \mathsf{s}_{\leq \mathsf{m}} \rceil$ and it suffices to show that $\mathsf{n} = \mathsf{n}'$. By lemma 178 we have $\mathsf{n} \in \lceil F\mathsf{s} \rceil$ and hence $\mathsf{n} \in \lceil T\mathsf{s} \rceil$ by lemma 184. Lemma 187 now yields $\mathsf{n} \in \lceil T\mathsf{s}_{\leq \mathsf{m}} \rceil$. By lemma 178 we have $\mathsf{n}' \in \lceil T\mathsf{s}_{\leq \mathsf{m}} \rceil$ and by lemma 186 we have $\mathsf{n} = \mathsf{n}'$.

Now we assume that $\mathsf{m} \in \mathbf{thread}(\mathsf{s})$ and show $\mathsf{m} \in \lceil T\mathsf{s} \rceil$. Let n be the initial opponent move in $\lceil \mathsf{s} \rceil$ and hence $\mathsf{n} \in \lceil \mathsf{s}_{\leq \mathsf{m}} \rceil$ by the definition of $\mathbf{thread}(-)$. By lemma 178 we have $\mathsf{n} \in \lceil F\mathsf{s} \rceil$ and $\mathsf{n} \in \lceil F\mathsf{s}_{\leq \mathsf{m}} \rceil$ and hence $\mathsf{n} \in \lceil T\mathsf{s} \rceil$ and $\mathsf{n} \in \lceil T\mathsf{s}_{\leq \mathsf{m}} \rceil$ by lemma 184. We can now apply lemma 187 to yield $\mathsf{m} \in \lceil T\mathsf{s} \rceil$.

■

Lemma 189 Given sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{V}_{\mathsf{A}\to\mathsf{B}}$ we have

$$\mathbf{thread}(\mathsf{s}) = \mathbf{thread}(\mathsf{s}') \Leftrightarrow \lceil T\mathsf{s} \rceil = \lceil T\mathsf{s}' \rceil.$$

Proof If $\mathbf{thread}(\mathsf{s}) = \mathbf{thread}(\mathsf{s}')$ then by lemma 188 we know that $\lceil T\mathsf{s} \rceil$ and $\lceil T\mathsf{s}' \rceil$ contain exactly the same moves. It is straightforward to show that $\lceil T\mathsf{s} \rceil = \lceil T\mathsf{s}' \rceil$ as we can simply add the appropriate extra justification pointers to $\mathbf{thread}(\mathsf{s})$ and $\mathbf{thread}(\mathsf{s}')$. Similarly $\mathbf{thread}(\mathsf{s})$ and $\mathbf{thread}(\mathsf{s}')$ can be recovered from $\lceil T\mathsf{s} \rceil$ and $\lceil T\mathsf{s}' \rceil$ by removing extra justification pointers. ■

Definition 190 Given a referee $\mathcal{R} \subseteq \mathcal{V}$ we define the umpire $\simeq_t : \mathcal{R}$ such that $\mathsf{s} \simeq_t \mathsf{s}'$ if and only if $\mathbf{thread}(\mathsf{s}) = \mathbf{thread}(\mathsf{s}')$. Clearly thread independence is equivalent to $\simeq_t$ observance.

Theorem 191 The umpire $\simeq_t : \mathcal{R}$ is compositional and copy sufficient.

Proof By lemma 183 we know that T is injective. For any arena A and sequences $\mathsf{s}, \mathsf{s}' \in \mathsf{A}$ we have the following by lemma 189:

$$(\mathsf{s} \simeq_t \mathsf{s}') \Leftrightarrow (\lceil T\mathsf{s} \rceil = \lceil T\mathsf{s}' \rceil)$$

and by lemma 185 we know that $T\mathsf{s}$ respects player visibility. Hence by lemma 160 it follows that $\simeq_t : \mathcal{R}$ is compositional and copy sufficient. ■

As a corollary we know by proposition 158 that the thread independent strategies form a lluf subcategory of $\mathbf{A}_{\mathcal{R}}$.

3.8 The Category of Legal Strategies

We are now in a position to demonstrate how we can use our reasoning referees and umpires to help us define the category of arenas that will be the subject of the next chapter.

Definition 192 We define the referee $\mathcal{L} = (\mathcal{V} \cap \mathcal{B})$. In other words for each arrow arena A we define the following set:

$$\mathcal{L}_{\mathsf{A}} = \{\mathsf{s} \in \mathcal{S}_{\mathsf{A}} \mid \mathsf{s} \in \mathcal{V}_{\mathsf{A}} \wedge \mathsf{s} \in \mathcal{B}_{\mathsf{A}}\}.$$

Lemma 193 The referee $\mathcal{L}$ is admissible by lemmas 134, 143 and 112 and we therefore have a category, $\mathbf{A}_{\mathcal{L}}$ of arenas where each morphism from arena A to B is a subset of $\mathcal{L}_{\mathsf{A}\to\mathsf{B}}$. For each arena A, the copycat strategy $\mathbf{copy}^{\mathcal{L}}_{\mathsf{A}}$ is the identity morphism.

Proof From lemmas 134 and 143 we know that the referees $\mathcal{V}$ and $\mathcal{B}$ are local, compositional and copy sufficient and hence by lemmas 109 and 112 we know that $\mathcal{L}$ is local, compositional and copy sufficient. Therefore, by theorem 106 we know that $\mathbf{A}_{\mathcal{L}}$ is a well-defined category. ■

Definition 194 Given an arrow arena A we say that a legal strategy is a prefix-closed, deterministic, thread independent subset of $\mathcal{L}_{\mathsf{A}}$.

Definition 195 We can now define the subcategory $\mathbf{A}^{T}_{\mathcal{L}}$ which is comprised of arenas and legal strategies. The lemmas 152, 153 and 191 assure us that the category is well-defined.

Chapter 4

Games Semantics for Idealized Algol

This chapter recalls the fully abstract games model for Idealized Algol with active expressions, or $\mathbf{IA}_a$ for short, due to Samson Abramsky and Guy McCusker [6] and as such some of the proofs are omitted. We make one or two cosmetic changes so as to make more obvious the connections between $\mathbf{IA}_a$, and its games model, and our own definitions of the interference controlled languages and the corresponding models which we present later. We will highlight any difference as it arises.

4.1 Idealized Algol

Idealized Algol is a programming language first proposed by John Reynolds [50]. The language consists of the simply-typed λ-calculus to which base types and imperative constants are added. The language can be considered as an imperative extension of the functional language PCF [44]. The resultant language is a combination of functional and imperative constructs. Owing to its elegance, Idealized Algol has been chosen again and again as the object language for fruitful research into the nature of imperative programming [42].

4.1.1 The $\mathbf{IA}_a$ Types

Reynolds originally made some sharp distinctions between different kinds of base type. Suppose we have a set of data values S. For example S could be $\mathbb{N}$, the set of natural numbers, or $\mathbb{B}$, the set of boolean values. Two distinct atomic types in Idealized Algol may be constructed from any given data set S.

1. τ_S: the type of program terms that yield a value from S. We call terms of this type *expressions*.
2. $\mathbf{var}_S$: the type of assignable terms that can store a value from S. We call terms of this type *variables*.

For our illustrative language we will consider only the data types $\mathbb{N}$, the set of natural numbers, and the singleton set $\mathbf{1} = \{\star\}$. We will not use the type $\mathbf{var_1}$ as it does not seem useful to store a value of unit type, however we write **com** for $\tau_\mathbf{1}$. We simply write **N** for $\tau_\mathbb{N}$ and **var** for $\mathbf{var}_\mathbb{N}$.

$\mathbf{IA}_a$

M	::=	x	identifiers
	|	n	numerals
	|	skip	null command
	|	MM	function application
	|	$\lambda x^{\theta}.M$	function abstraction
	|	$\langle M,M\rangle$	pairing
	|	$\pi_1 M$	left projection
	|	$\pi_2 M$	right projection
	|	$M;M$	sequential composition
	|	while $M=0$ do M	looping
	|	if $M=0$ then M else M	conditional
	|	new $M:=0$ in M	variable allocation
	|	do $M:=0$ then M	side effecting expressions
	|	$M:=M$	assignment
	|	$!M$	dereference
	|	mkvar M M	bad variable constructor
	|	succ M	successor
	|	pred M	predecessor
	|	Y M	fixed point combinator

Table 4.1: Term Grammar for $\mathbf{IA}_a$

Reynolds originally makes a distinction between terms of type **com**, which are executed primarily in order to produce side effects, and terms of any other type whose execution is always side effect free. The language we will consider here is different: we will allow terms of type **N** to produce side effects. This extended language is termed Idealized Algol with active expressions. A fully abstract games model for the language without active expressions was subsequently presented by Abramsky and McCusker in [10].

We have already described our atomic types:**N**, **com** and **var**. We use the metavariable τ to represent a type that is either **N** or **com**. The $\mathbf{IA}_a$ types, θ, are constructed as follows:

$$\theta ::= \tau \mid \mathbf{var} \mid \theta \times \theta \mid \theta \Rightarrow \theta.$$

Note that the presentation of $\mathbf{IA}_a$ in [6] does not include product types however it is well known that the games model presented therein is fully abstract for the language extended with product types.

4.1.2 $\mathbf{IA}_a$ Terms

Terms are constructed using the grammar defined in table 4.1. We include here some extra term

IA$_a$

$$\frac{}{\Gamma, x:\theta \vdash_A x:\theta}\ Axiom \qquad \frac{\Gamma \vdash_A P:\theta_0 \quad \Gamma \vdash_A Q:\theta_1}{\Gamma \vdash_A \langle P,Q\rangle : \theta_0 \times \theta_1} \times I$$

$$\frac{\Gamma \vdash_A \langle P,Q\rangle : \theta \times \theta'}{\Gamma \vdash_A \pi_1 P:\theta} \times E_1 \qquad \frac{\Gamma \vdash_A \langle P,Q\rangle : \theta \times \theta'}{\Gamma \vdash_A \pi_2 P:\theta'} \times E_2$$

$$\frac{\Gamma, x:\theta' \vdash_A P:\theta}{\Gamma \vdash_A \lambda x:\theta'.P:\theta' \Rightarrow \theta} \Rightarrow I \qquad \frac{\Gamma \vdash_A P:\theta' \Rightarrow \theta \quad \Gamma \vdash_A Q:\theta'}{\Gamma \vdash_A PQ:\theta} \Rightarrow E$$

Table 4.2: Typing Rules for λ-calculus fragment of **IA**$_a$.

constructors,

$$\mathsf{do}\ M := 0\ \mathsf{then}\ M$$

and

$$\mathsf{while}\ M = 0\ \mathsf{do}\ M$$

which are not present in [6], but will be present in the interference controlled languages that we present later. However, we need not be concerned as these new constructs can be regarded as syntactic sugar: $\mathsf{while}\ M = 0\ \mathsf{do}\ N$ is equivalent to

$$(\mathsf{Y}\ (\lambda f^{\mathbf{N}\Rightarrow\mathbf{com}\Rightarrow\mathbf{com}}.\lambda x^{\mathbf{N}}.\lambda y^{\mathbf{com}}.\mathsf{if}\ x = 0\ \mathsf{then}\ y;\ fxy\ \mathsf{else\ skip}))MN$$

and $\mathsf{do}\ x := 0\ \mathsf{then}\ M$ is equivalent to $\mathsf{new}\ x := 0\ \mathsf{in}\ M; x$.

4.1.3 Typing Rules for IA$_a$

The typing rules for the λ-calculus fragment of **IA**$_a$ are given in table 4.2. It is well known that the additive nature of the application rule implies that the following two rules are admissible:

$$\frac{\Gamma, x:\theta', y:\theta' \vdash_A P:\theta}{\Gamma, x:\theta' \vdash_A P[x/y]:\theta}\ \textbf{contraction}$$

$$\frac{\Gamma \vdash_A P:\theta}{\Gamma, \Delta \vdash_A P:\theta}\ \textbf{weakening}$$

Typing rules for the other **IA**$_a$ constructs are given in table 4.3.

4.2 Operational Semantics for IA$_a$

The operational semantics for **IA**$_a$ is call by name and is defined on terms with respect to stores. A **var** context $x_1, \ldots, x_n$ is one in which all identifiers are of type **var** and a store is a partial function that maps identifiers in such a context to natural numbers. Given store s, we define store update as follows.

$$\begin{aligned} (s|x \mapsto n)x &= n \\ (s|x \mapsto n)y &= sy \quad \text{if } y \neq x. \end{aligned}$$

IA$_a$

$$\frac{}{\vdash_A n : \mathbf{N}} \qquad \frac{}{\vdash_A \mathsf{skip} : \mathbf{com}}$$

$$\frac{\Gamma \vdash_A M : \mathbf{N}}{\Gamma \vdash_A \mathsf{succ}\, M : \mathbf{N}} \qquad \frac{\Gamma \vdash_A M : \mathbf{N}}{\Gamma \vdash_A \mathsf{pred}\, M : \mathbf{N}}$$

$$\frac{\Gamma \vdash_A M : \mathbf{com} \quad \Gamma \vdash_A N : \tau}{\Gamma \vdash_A M;N : \tau} \qquad \frac{\Gamma \vdash_A M : \theta \Rightarrow \theta}{\Gamma \vdash_A \mathsf{Y}\, M : \theta}$$

$$\frac{\Gamma \vdash_A L : \mathbf{N} \quad \Gamma \vdash_A M : \tau \quad \Gamma \vdash_A N : \tau}{\Gamma \vdash_A \mathsf{if}\ L = 0\ \mathsf{then}\ M\ \mathsf{else}\ N : \tau} \qquad \frac{\Gamma \vdash_A M : \mathbf{N} \Rightarrow \mathbf{com} \quad \Gamma \vdash_A N : \mathbf{N}}{\Gamma \vdash_A \mathsf{mkvar}\, M\, N : \mathbf{var}}$$

$$\frac{\Gamma \vdash_A M : \mathbf{var} \quad \Gamma \vdash_A N : \mathbf{N}}{\Gamma \vdash_A M := N : \mathbf{com}} \qquad \frac{\Gamma \vdash_A M : \mathbf{var}}{\Gamma \vdash_A !M : \mathbf{N}}$$

$$\frac{\Gamma, x : \mathbf{var} \vdash_A M : \mathbf{com}}{\Gamma \vdash_A \mathsf{new}\ x := 0\ \mathsf{in}\ M : \mathbf{com}} \qquad \frac{\Gamma, x : \mathbf{var} \vdash_A M : \mathbf{com}}{\Gamma \vdash_A \mathsf{do}\ x := 0\ \mathsf{then}\ M : \mathbf{N}}$$

Table 4.3: Typing Rules for the Language Constructs of **IA**$_a$

Canonical Forms

$$\frac{}{s, V \Downarrow_A s, V}$$

A subset of typeable terms at each type are described as being of canonical form, or values. These are $\mathsf{skip}, n, \langle M, N \rangle$ and $\lambda x.M$. The operational semantics is defined in the big step style using a relation $\Downarrow_A$ so that $s, M \Downarrow_A s', V$ means that evaluating term typeable term M in initial store s yields a canonical form V with final store s'.

Sequencing and Conditionals

$$\frac{s, M \Downarrow_A s', \mathsf{skip} \qquad s', N \Downarrow_A s'', V}{s, M;N \Downarrow_A s'', V}$$

$$\frac{s, L \Downarrow_A s', n+1 \qquad s', M \Downarrow_A s'', V}{s, \mathsf{if}\ L = 0\ \mathsf{then}\ M\ \mathsf{else}\ N \Downarrow_A s'', V}$$

$$\frac{s, L \Downarrow_A s', 0 \qquad s', N \Downarrow_A s'', V}{s, \mathsf{if}\ L = 0\ \mathsf{then}\ M\ \mathsf{else}\ N \Downarrow_A s'', V}$$

Arithmetic

$$\frac{s, M \Downarrow_A s', n}{s, \mathsf{succ}\ M \Downarrow_A s', n+1}$$

$$\frac{s, M \Downarrow_A s', n+1}{s, \mathsf{pred}\ M \Downarrow_A s', n}$$

Looping and Fixed Point

$$\frac{s, \mathsf{if}\ L = 0\ \mathsf{then}\ M; (\mathsf{while}\ L = 0\ \mathsf{do}\ M)\ \mathsf{else}\ \mathsf{skip} \Downarrow_A s', \mathsf{skip}}{s, \mathsf{while}\ L = 0\ \mathsf{do}\ M \Downarrow_A s', \mathsf{skip}}$$

$$\frac{s, M(\mathsf{Y}\ M) \Downarrow_A s', V}{s, \mathsf{Y}\ M \Downarrow_A s', V}$$

Projection

$$\frac{s, L \Downarrow_A s', \langle M, N \rangle \qquad s', M \Downarrow_A s'', V}{\pi_1 L, s \Downarrow_A s'', V}$$

$$\frac{s, L \Downarrow_A s', \langle M, N \rangle \qquad s', N \Downarrow_A s'', V}{s, \pi_2 L \Downarrow_A s'', V}$$

Application

$$\frac{s, L \Downarrow_A s', \lambda x.M \qquad s', M[N/x] \Downarrow_A s'', V}{s, L(N) \Downarrow_A s'', V}$$

Store

$$\frac{s, N \Downarrow_A s', n \qquad s', M \Downarrow_A s'', x}{s, M := N \Downarrow_A (s''|x \mapsto n), \mathsf{skip}}$$

$$\frac{s, M \Downarrow_A s', x}{s, !M \Downarrow_A s', n}\ s'(x) = n$$

$$\frac{s, N \Downarrow_A s', n \qquad s', M \Downarrow_A s'', \mathsf{mkvar}\ M_1\ M_2 \qquad s'', M_1 n \Downarrow_A s''', \mathsf{skip}}{s, M := N \Downarrow_A s''', \mathsf{skip}}$$

$$\frac{s, M \Downarrow_A s', \mathsf{mkvar}\ M_1\ M_2 \qquad s', M_2 \Downarrow_A s'', n}{s, !M \Downarrow_A s'', n}$$

$$\frac{(s|x \mapsto 0), M \Downarrow_A (s'|x \mapsto n), \mathsf{skip}}{s, \mathsf{new}\ x := 0\ \mathsf{in}\ M \Downarrow_A (s'|x \mapsto s(x)), \mathsf{skip}}$$

$$\frac{(s|x \mapsto 0), M \Downarrow_A (s'|x \mapsto n), \mathsf{skip}}{s, \mathsf{do}\ x := 0\ \mathsf{then}\ M \Downarrow_A s', n}$$

4.3 The Denotational Semantics of $\mathbf{IA}_a$

We are now going to show how to interpret $\mathbf{IA}_a$ in a category of arenas. The category that we will use is $\mathbf{A}^T_{\mathcal{L}}$ as defined at the end of chapter 3. As already stated, this interpretation follows very closely the games model of Abramsky and McCusker [6] and as such is not intended as original work. However some cosmetic differences do exist in our presentation — we do not suggest that these changes are an improvement but instead they are included to facilitate more direct reasoning about our model for SCI languages in chapter 6 by using results presented in this chapter.

4.3.1 Models of $\mathbf{IA}_a$

Abramsky and McCusker interpret $\mathbf{IA}_a$ in a category of games after first examining the categorical structure that any such a model should possess. As we have already seen $\mathbf{IA}_a$ is an applied simply typed λ-calculus therefore they ask for a model to be a cartesian-closed category [16]. To interpret recursion they ask for the category to be cpo-enriched [13], or at least rational [4].

4.3.2 The Cartesian Closed Structure of $\mathbf{A}^T_{\mathcal{L}}$

We omit proof of the categorical structure here. The sceptical reader can refer to proofs of similar claims in chapter 6.

Lemma 196 Given any sequence $\mathsf{s} \in \mathcal{L}_{\mathsf{A}}$ it is straightforward to show that $\mathbf{thread}(\mathsf{s}) \in \mathcal{L}_{\mathsf{A}}$.

Definition 197 Given a strategy σ we write $\sigma_{\mathsf{threads}}$ for the set of single threads in σ.

The idea is that any strategy σ in $\mathbf{A}^T_{\mathcal{L}}$ is completely determined by the set of the single threads it contains. We do this by showing that $-_{\mathsf{threads}}$ is injective by defining its inverse.

Definition 198 Given a set $S \subseteq \mathcal{L}_{\mathsf{A}}$ of single threaded sequences we define the set $\mathsf{weave}(S)$ as the least set such that

1. $\varepsilon \in \mathsf{weave}(S)$
2. Given $\mathsf{s} \in \mathsf{weave}(S)$ and $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in \mathcal{L}_{\mathsf{A}}$ such that $\mathbf{thread}(\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}') \in S$ then we have $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in \mathsf{weave}(S)$

It is straightforward to show that given a legal strategy $\sigma : \mathsf{A}$ it follows that:

$$\mathsf{weave}(\sigma_{\mathsf{threads}}) = \sigma.$$

We will find it convenient to define strategies in terms of the individual threads that they contain.

Definition 199 The **null arena** is defined as follows:

$$1 = \langle \emptyset, \emptyset, \emptyset \rangle.$$

For any object A in $\mathbf{A}^T_{\mathcal{L}}$ we have exactly one strategy for $\mathsf{A} \to 1$, the strategy containing only the empty sequence, hence 1 is terminal in $\mathbf{A}^T_{\mathcal{L}}$.

Definition 200 The product of arenas A and B has 1 as its unit and is defined as follows.

$$\begin{aligned} \mathsf{M}_{\mathsf{A}\times\mathsf{B}} &= \mathsf{M}_{\mathsf{A}} + \mathsf{M}_{\mathsf{B}} \\ \lambda_{\mathsf{A}\times\mathsf{B}} &= [\lambda_{\mathsf{A}}, \lambda_{\mathsf{B}}] \\ \vdash_{\mathsf{A}\times\mathsf{B}} &= \vdash_{\mathsf{A}} + \vdash_{\mathsf{B}} \end{aligned}$$

Given strategies $\sigma : \mathsf{A} \to \mathsf{C}$ and $\tau : \mathsf{B} \to \mathsf{D}$ we define:

$$\sigma \times \tau = \mathsf{weave}(\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}).$$

The projections $\pi_1 : \mathsf{A} \times \mathsf{B} \to \mathsf{A}$ and $\pi_2 : \mathsf{A} \times \mathsf{B} \to \mathsf{B}$ are defined by the obvious copycat strategies. Given strategies $\sigma : \mathsf{C} \to \mathsf{A}$ and $\tau : \mathsf{C} \to \mathsf{B}$ we form the pairing $\langle \sigma, \tau \rangle : \mathsf{C} \to \mathsf{A} \times \mathsf{B}$ as follows

$$\langle \sigma, \tau \rangle = \mathsf{weave}(\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}).$$

It is straightforward to show by means of surjective pairing that $\times$ is indeed the product.

Definition 201 Given negative arenas A and B we define the exponential formally as follows:

$$\begin{aligned} \mathsf{M}_{\mathsf{A}\Rightarrow\mathsf{B}} &= \mathsf{M}_{\mathsf{A}} + \mathsf{M}_{\mathsf{B}} \\ \lambda_{\mathsf{A}\Rightarrow\mathsf{B}} &= [\overline{\lambda_{\mathsf{A}}}, \lambda_{\mathsf{B}}] \\ \mathsf{m} \vdash_{\mathsf{A}\Rightarrow\mathsf{B}} \mathsf{m}' &\Leftrightarrow (\mathsf{m} \vdash_{\mathsf{A}} \mathsf{m}') \vee (\mathsf{m} \vdash_{\mathsf{B}} \mathsf{m}') \\ &\quad \vee (\star \vdash_{\mathsf{B}} \mathsf{m} \wedge \star \vdash_{\mathsf{A}} \mathsf{m}') \\ \star \vdash_{\mathsf{A}\Rightarrow\mathsf{B}} \mathsf{m}' &\Leftrightarrow \star \vdash_{\mathsf{B}} \mathsf{m}' \end{aligned}$$

It is simple to show that we have an isomorphism of hom-sets $\mathbf{A}^T_{\mathcal{L}}(\mathsf{A} \times \mathsf{B}, \mathsf{C}) \simeq \mathbf{A}^T_{\mathcal{L}}(\mathsf{A}, \mathsf{B} \Rightarrow \mathsf{C})$ and hence $\mathbf{A}^T_{\mathcal{L}}$ is cartesian closed.

Furthermore, it is straightforward to show that subset inclusion of the legal strategies for any arena form a directed-complete partial order and hence the category $\mathbf{A}^T_{\mathcal{L}}$ is CPO enriched.

4.3.3 The Interpretation of Types

Definition 202 (Flat QA Arenas) For any basic data type τ we define the *flat* QA arena: τ_f as follows:

$$\begin{aligned} \mathsf{M}_{\tau_f} &= \{\mathsf{q}\} \cup \tau \\ \lambda_{\tau_f}\mathsf{q} &= OQ \\ \lambda_{\tau_f} n &= PA \text{ for each } n \in \tau \\ \star &\vdash_{\tau_p} \mathsf{q} \\ \mathsf{q} &\vdash_{\tau_f} n \text{ for each } n \in \tau \end{aligned}$$

For the purpose of our illustrative language we will be particularly interested in the flat arenas where $\tau = \mathbb{N}$, the set of natural numbers. We write $\mathbf{N}$ for $\mathbb{N}_f$. We are also interested in the flat arena for the singleton data type for which we write **com** and with moves run and done for the unique initial move and corresponding unique answer. Note that the names of our arenas may coincide with the types in the languages; no confusion will arise in practice.

Definition 203 (Variable Arenas) For each basic data type τ we also define the variable QA arena $\mathbf{var}_\tau$ as follows:

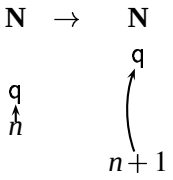
$$\begin{array}{rcl}
\mathsf{M}_{\mathbf{var}_\tau} & = & \{\mathsf{ok},\mathsf{q}\} \cup \tau \cup \{\mathsf{write}_n | \mathsf{n} \in \tau\} \\
\lambda_{\mathbf{var}_\tau}\mathsf{write}_n & = & OQ \\
\lambda_{\mathbf{var}_\tau}\mathsf{ok} & = & PA \\
\lambda_{\mathbf{var}_\tau}\mathsf{q} & = & OQ \\
\lambda_{\mathbf{var}_\tau} n & = & PA \\
\star & \vdash_{\mathbf{var}_\tau} & \mathsf{write}_n \\
\star & \vdash_{\mathbf{var}_\tau} & \mathsf{q} \\
\mathsf{write}_n & \vdash_{\mathbf{var}_\tau} & \mathsf{ok} \\
\mathsf{q} & \vdash_{\mathbf{var}_\tau} & \mathsf{n}
\end{array}$$

For the purposes of our illustrative language we will again be interested in variable arenas for the basic data type of natural numbers. We will simply write **var** for $\mathbf{var}_{\mathbb{N}}$.

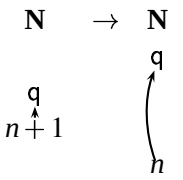
We will now define some useful strategies. Our definitions will only describe the non-empty single threaded plays that are *complete* in the sense that every question is answered but this is sufficient to reconstitute the strategy.

Definition 204

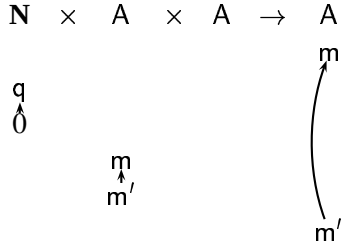
- For each natural number n we define the strategy $\mathbf{n} : \mathbf{N}$ to have single-threaded sequences of the form $\mathsf{q} \cdot n$.
- We define $\mathbf{s} : \mathbf{N} \to \mathbf{N}$ to have single-threaded complete plays of the form:

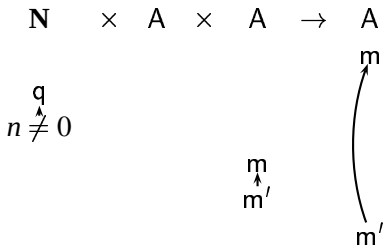


- We define $\mathbf{p} : \mathbf{N} \to \mathbf{N}$ to have single threaded complete plays of the form:

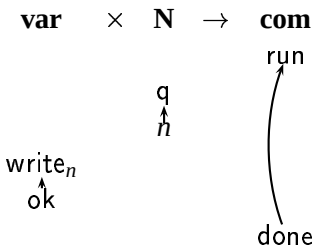


- For each flat arena A we define the strategy $\mathbf{ifz} : \mathbf{N} \times \mathsf{A} \times \mathsf{A} \to \mathsf{A}$ which has complete, single threaded plays of the following forms:

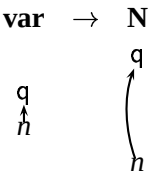




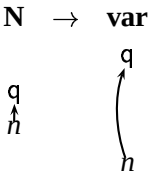
- We define the strategy **assign** : **var** × **N** → **com** to have single threaded complete plays of the form:

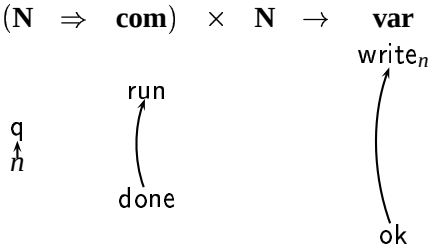


- The strategy **deref** : **var** → **N** has single threaded, complete plays of the following form:

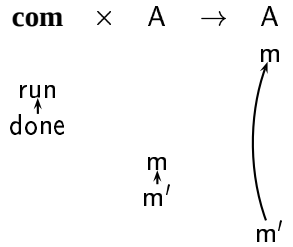


- The strategy **mkvar** : (**N** ⇒ **com**) × **N** → **var** has single threaded, complete plays of one of the following forms:

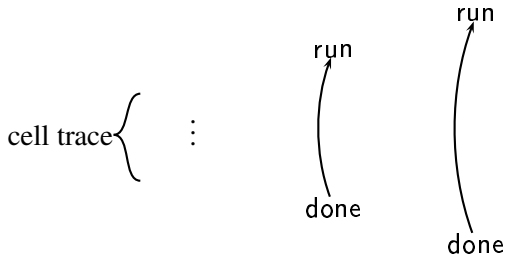




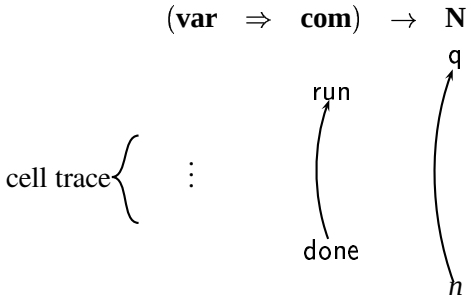
- Given a flat arena A we define the strategy $\mathbf{seq}_{\mathsf{A}}$ with single threaded complete plays of the following form.



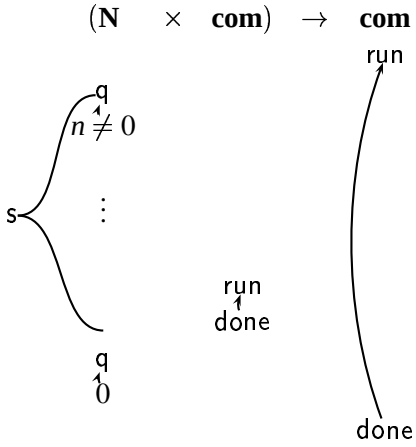
- The strategy **new** : $(\mathbf{var} \Rightarrow \mathbf{com}) \to \mathbf{com}$ has complete, single threaded sequences of the following form. Where a cell trace is of the form $(\mathsf{read} \cdot 0)^* \cdot ((\mathsf{write}_n \cdot \mathsf{ok} \cdot (\mathsf{read} \cdot n)^*)^*$.



- Similarly, the strategy **do** : $(\mathbf{var} \Rightarrow \mathbf{com}) \to \mathbf{N}$ has complete, single-threaded sequences of the following form with the cell trace defined as before and write_n is the last write move in the cell trace or $n = 0$ if no such move has been played.



- The strategy **while** : $\mathbf{N} \times \mathbf{com} \to \mathbf{com}$ has complete single-threaded plays of following form where the sequence s can be repeated zero or more times:



A type A will be modelled in $\mathbf{A}^T_{\mathcal{L}}$ by a QA arena $[\![A]\!]_A$. We start by defining the arenas corresponding to base types:

$$\begin{aligned} [\![\mathbf{N}]\!]_A &= \mathbf{N} \\ [\![\mathbf{var}]\!]_A &= \mathbf{var} \\ [\![\mathbf{com}]\!]_A &= \mathbf{com}. \end{aligned}$$

The interpretation of higher types is defined inductively:

$$[\![A \times B]\!]_A = [\![A]\!]_A \times [\![B]\!]_A \text{ and } [\![A \Rightarrow B]\!]_A = [\![A]\!]_A \Rightarrow [\![B]\!]_A.$$

4.3.4 The Interpretation of Typing Judgements

A typed term

$$x_1 : A_1 \ldots x_n : A_n \vdash_A M : A$$

will be modelled by a morphism from $[\![A_1]\!]_A \times \ldots \times [\![A_n]\!]_A$ to $[\![A]\!]_A$; in other words, a strategy for the arena

$$[\![A_1]\!]_A \times \ldots \times [\![A_n]\!]_A \to [\![A]\!]_A,$$

and specifically a closed term $\vdash_A M : A$ will be modelled by a strategy for the arena $1 \to [\![A]\!]_A$.

4.3.5 Interpreting the λ-calculus

The λ-calculus and pairing part of $\mathbf{IA}_a$ is interpreted using the Cartesian Closed structure. Identifiers are interpreted as projections:

$$[\![x_1 : A_1, \ldots, x_n : A_n \vdash_A x_i : A_i]\!]_A = \pi_i : [\![A_1]\!]_A \times \ldots \times [\![A_n]\!]_A \to [\![A_i]\!]_A.$$

Abstraction is modelled by currying:

$$[\![\Gamma \vdash_A M : A \Rightarrow B]\!]_A = \Lambda([\![\Gamma, x : A \vdash_A M : B]\!]_A) : [\![\Gamma]\!]_A \to [\![A \Rightarrow B]\!]_A.$$

Application is modelled by composition with the evaluation map:

$$[\![\Gamma \vdash_A MN : B]\!]_A = \langle [\![\Gamma \vdash_A M : A \Rightarrow B]\!]_A, [\![\Gamma \vdash_A N : A]\!]_A \rangle ; \mathsf{eval}_{[\![A \Rightarrow B]\!]_A}.$$

Pairing is modelled using the pairing map:

$$[\![\Gamma \vdash_A \langle M, N \rangle : A \times B]\!]_A = \langle [\![\Gamma \vdash_A M : A]\!]_A, [\![\Gamma \vdash_A N : B]\!]_A \rangle.$$

Projection is interpreted by composition with the projection map:

$$[\![\Gamma \vdash_A \pi_i N : A]\!]_A = [\![\Gamma \vdash_A N : A \times B]\!]_A ; \pi_i.$$

4.3.6 Interpreting $\mathbf{IA}_a$ Constants

The semantics of typed terms is defined inductively as follows:

$$\begin{aligned}
[\![\Gamma \vdash_A n]\!]_A &= \mathbf{n} \\
[\![\Gamma \vdash_A \mathsf{succ}\ n]\!]_A &= [\![\Gamma \vdash_A \mathsf{n}]\!]_A ; \mathbf{s} \\
[\![\Gamma \vdash_A \mathsf{pred}\ n]\!]_A &= [\![\Gamma \vdash_A \mathsf{n}]\!]_A ; \mathbf{p} \\
[\![\Gamma \vdash_A M := N]\!]_A &= \langle [\![\Gamma \vdash_A M]\!]_A, [\![\Gamma \vdash_A N]\!]_A \rangle ; \mathbf{assign} \\
[\![\Gamma \vdash_A !M]\!]_A &= [\![\Gamma \vdash_A M]\!]_A ; \mathbf{deref} \\
[\![\Gamma \vdash_A \mathsf{mkvar}\ M\ N]\!]_A &= \langle [\![\Gamma \vdash_A M]\!]_A, [\![\Gamma \vdash_A N]\!]_A \rangle ; \mathbf{mkvar} \\
[\![\Gamma \vdash_A M; N]\!]_A &= \langle [\![\Gamma \vdash_A M]\!]_A, [\![\Gamma \vdash_A N]\!]_A \rangle ; \mathbf{seq} \\
[\![\Gamma \vdash_A \mathsf{while}\ M = 0\ \mathsf{do}\ N]\!]_A &= \langle [\![\Gamma \vdash_A M]\!]_A, [\![\Gamma \vdash_A N]\!]_A \rangle ; \mathbf{while} \\
[\![\Gamma \vdash_A \mathsf{if}\ L = 0\ \mathsf{then}\ M\ \mathsf{else}\ N]\!]_A &= \langle [\![\Gamma \vdash_A M]\!]_A, [\![\Gamma \vdash_A M]\!]_A, [\![\Gamma \vdash_A N]\!]_A \rangle ; \mathbf{ifz} \\
[\![\Gamma \vdash_A \mathsf{new}\ x := 0\ \mathsf{in}\ M]\!]_A &= [\![\Gamma \vdash_A \lambda x.M]\!]_A ; \mathbf{new} \\
[\![\Gamma \vdash_A \mathsf{do}\ x := 0\ \mathsf{then}\ M]\!]_A &= [\![\Gamma \vdash_A \lambda x.M]\!]_A ; \mathbf{do}
\end{aligned}$$

To interpret terms of the form $\Gamma \vdash_A \mathsf{Y}\, M : \theta$ we appeal to the cpo-enriched nature of our category. Given a term $\Gamma \vdash_A M : A \Rightarrow A$ we can uncurry the interpretation to yield a strategy:

$$\sigma : [\![\Gamma]\!]_A \times [\![A]\!]_A \to M : [\![A]\!]_A.$$

We can now define the following chain of strategies:

$$\begin{aligned} \sigma_0 &= \bot \\ \sigma_{n+1} &= \langle \mathbf{id}_{[\![\Gamma]\!]_A}, \sigma_n \rangle ; \sigma \end{aligned}$$

and we interpret $\Gamma \vdash_A \mathsf{Y}\, M : \theta$ as the least upper bound of this chain. More concretely, we can inductively define, for each game A, a strategy $\mathbf{rec}_{\mathsf{A}} : (\mathsf{A} \Rightarrow \mathsf{A}) \to \mathsf{A}$. In order for us to distinguish the separate subgames in this definition we will explicitly tag the components and write $\mathbf{rec}_{\mathsf{A}} : \mathsf{A} \Rightarrow \mathsf{A}' \to \mathsf{A}''$. We will write m for a move from A, m' for the corresponding move in A' and, similarly, m'' for the corresponding move in A'' and we will write $\mathsf{j}, \mathsf{j}', \mathsf{j}''$ for the justifiers of $\mathsf{m}, \mathsf{m}', \mathsf{m}''$ respectively where appropriate. We write n^- for the move immediately preceding a given move n.

- $\varepsilon \in \mathbf{rec}_{\mathsf{A}}$.
- $\mathsf{s} \in \mathbf{rec}_{\mathsf{A}} \wedge \mathsf{s} \cdot \mathsf{m} \in \mathcal{L}_{\mathsf{A} \Rightarrow \mathsf{A}' \to \mathsf{A}''} \Rightarrow \mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in \mathbf{rec}_{\mathsf{A}}$ with m' initial if j is from A', otherwise $\mathsf{j}^- \curvearrowleft \mathsf{m}'$.
- $\mathsf{s} \in \mathbf{rec}_{\mathsf{A}} \wedge \mathsf{s} \cdot \mathsf{m}'' \in \mathcal{L}_{\mathsf{A} \Rightarrow \mathsf{A}' \to \mathsf{A}''} \Rightarrow \mathsf{s} \cdot \mathsf{m}'' \cdot \mathsf{m}' \in \mathbf{rec}_{\mathsf{A}}$ If m'' is initial then so is m', otherwise we have $\mathsf{j}''^- \curvearrowleft \mathsf{m}'$.
- Suppose $\mathsf{s} \in \mathbf{rec}_{\mathsf{A}}$ and $\mathsf{s} \cdot \mathsf{m}' \in \mathbf{rec}_{\mathsf{A}}$. If j'^- is from A then $\mathsf{s} \cdot \mathsf{m}' \cdot \mathsf{m} \in \mathbf{rec}_{\mathsf{A}}$ with $\mathsf{j}'^- \curvearrowleft \mathsf{m}$. If j'^- is from A'' then $\mathsf{s} \cdot \mathsf{m}' \cdot \mathsf{m}'' \in \mathbf{rec}_{\mathsf{A}}$ with $\mathsf{j}'^- \curvearrowleft \mathsf{m}''$.

Note that player is always copying the previous move. We can now see that the interpretion $\Gamma \vdash_A \mathsf{Y}\, M$ is equivalent to:

$$[\![\Gamma \vdash_A M]\!]_A ; \mathbf{rec}_{\mathsf{A}}.$$

For every $\mathbf{IA}_a$ type A we define the divergent term:

$$\Omega_A : A = \mathsf{Y}\, (\lambda x^A.x)$$

with semantics $\{\varepsilon\}$.

4.4 Inequational Soundness

The object of this section is to show that the following inequational soundness result holds. For all terms $\Gamma \vdash_A M : A$, $\Gamma \vdash_A N : A$ it follows that:

$$[\![M]\!]_A \subseteq [\![N]\!]_A \Rightarrow M \underset{\sim}{\sqsubseteq} N.$$

This of course implies that equality in the model implies observational equivalence and also that we can use subset inclusion of strategies to reason about the observational preorder.

The proof follows that given in [24] and starts with a substitution lemma.

Lemma 205 Given terms $\Gamma, x : A \vdash_A M : B$ and $\Gamma \vdash_A N : A$ it follows that $\Gamma \vdash_A M[N/x] : B$ is well typed and also that:

$$[\![M[N/x]]\!]_A = \langle \mathsf{id}_\Gamma, [\![N]\!]_A \rangle ; [\![M]\!]_A.$$

Proof The proof is by induction on the structure of M. ■

We need to make this notion more specific so we define a notion of state transition. Given a sequence $\mathsf{s} \in \mathcal{L}_{[\![\mathbf{var}]\!]}$, we define the transitions

$$n \xrightarrow{\mathsf{s}} n'$$

where n and n' are natural numbers, and $\mathsf{s} \in \mathcal{L}_{\mathbf{var}}$ as follows.

$$\frac{}{n \xrightarrow{\varepsilon} n} \qquad \frac{}{n \xrightarrow{\mathsf{read} \cdot n} n}$$

$$\frac{}{n \xrightarrow{\mathsf{write}(n') \cdot \mathsf{ok}} n'} \qquad \frac{n \xrightarrow{s} n' \quad n' \xrightarrow{s'} n''}{n \xrightarrow{s \cdot s'} n''}$$

We extend this to traces involving more than one **var** type as follows. Given a context $\Gamma = x_1 : \mathbf{var}, \ldots, x_n : \mathbf{var}$, a sequence $\mathsf{t} \in \mathcal{L}_{[\![\Gamma]\!]_A}$, and states s and s' in variables $x_1, \ldots, x_n$, we write

$$s \xrightarrow{\mathsf{t}} s'$$

if and only if

$$s(x_i) \xrightarrow{\mathsf{t} \upharpoonright [\![x_i]\!]_A} s'(x_i)$$

for each i.

Definition 206 We say that a sequence $\mathsf{s} \in \mathcal{L}_{\mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha}$ is a multi-cell trace if and only if for all $1 \leq i \leq n$ it follows that $\mathsf{s} \upharpoonright \mathbf{var}_i^\alpha$ is a cell trace.

4.4.1 Operational Soundness

The next step is to show that the model is sound for evaluation.

Lemma 207 Given any term $\Gamma \vdash_A M : A$, where A is not of type τ, it follows that:

$$M, s \Downarrow_A V, s' \Rightarrow [\![M]\!]_A = [\![V]\!]_A \text{ and } \mathsf{s} = \mathsf{s}'.$$

Proof This is proved by induction on the derivation of $M, s \Downarrow_A V, s'$ using the substitution lemma and the continuity of composition. ■

Lemma 208 Given any term $\Gamma \vdash_A M : \mathbf{com}$ such that $M, s \Downarrow_A \mathsf{skip}, s'$ it follows that there exists a sequence $\mathsf{t} \in [\![M]\!]_A$ such that

$$s \xrightarrow{\mathsf{t} \upharpoonright [\![\Gamma]\!]_A} s'$$

and $\mathsf{s} \upharpoonright [\![\mathbf{com}]\!]_A = \mathsf{run} \cdot \mathsf{done}$.

Similarly, given any term $\Gamma \vdash_A M : \mathbf{N}$, such that $M, s \Downarrow_A n, s'$ it follows that there exists a sequence $\mathsf{t} \in [\![M]\!]_A$ such that

$$s \xrightarrow{\mathsf{t} \upharpoonright [\![\Gamma]\!]_A} s'$$

and $\mathsf{s} \upharpoonright [\![\mathbf{N}]\!]_A = \mathsf{q} \cdot n$.

Proof This is proved by induction on the derivation of $M, s \Downarrow_A V, s'$ and of course yields the

following soundness result:

$$M, s \Downarrow_A V, s' \Rightarrow [\![M]\!]_A \neq \bot .$$

■

4.4.2 Computational Adequacy

Next comes a proof of computational adequacy which uses a computability predicate in the style of Tait-Girard-Plotkin [44]. The computability predicate is defined on *semi-closed* terms —terms whose only free identifiers are of type **var**.

Definition 209 (Computable Terms)

- A semi-closed term $\Gamma \vdash_A M : \mathbf{com}$ is computable if and only if whenever there exists a sequence $\mathsf{t} \in [\![M]\!]_A$ such that
$$s \xrightarrow{\mathsf{t} \restriction [\![\Gamma]\!]_A} s'$$
for some stores s and s', and $\mathsf{t} \restriction [\![A]\!]_A = \mathsf{run} \cdot \mathsf{done}$ then it follows that $M, s \Downarrow_A \mathsf{skip}, s'$.
- A semi-closed term $\Gamma \vdash_A M : \mathbf{N}$ is computable if and only if whenever there exists a sequence $\mathsf{t} \in [\![M]\!]_A$ such that
$$s \xrightarrow{\mathsf{t} \restriction [\![\Gamma]\!]_A} s'$$
for some stores s and s', and $\mathsf{t} \restriction [\![A]\!]_A = \mathsf{q} \cdot n$ then it follows that $M, s \Downarrow_A n, s'$.
- A semi-closed term $\Gamma \vdash_A M : \mathbf{var}$ is computable if and only if
$$\Gamma \vdash_A !M : \mathbf{N}$$
is computable, and for all numerals n
$$\Gamma \vdash_A M := n : \mathbf{com}$$
is computable.
- A semi-closed term $\Gamma \vdash_A M : A \Rightarrow B$ is computable if and only if $\vdash_A MN : B$ is computable for all semi-closed computable terms $\Gamma \vdash_A N : A$.
- An open term $x_1 : A_1, \ldots, x_n : A_n, \Gamma \vdash_A M : B$, where Γ contains only identifiers of type **var**, is computable if and only if for all computable semi-closed terms
$$\Gamma \vdash_A N : A_1, \ldots, \Gamma \vdash_A N : A_n$$
it follows that $\Gamma \vdash_A M[N_1/x_1, \ldots N_n/x_n]$ computable.

Computational adequacy is implied by the following lemma:

Lemma 210 All $\mathbf{IA}_a$ terms are computable.

Proof The first stage is to prove that terms built without subterms of the form $\mathsf{Y}\ N$, except for

the subterm Ω, are computable. This proof is by induction on the structure of the term. The final result is now proved by defining a term M_k, for every term M, where every subterm of the form $\Upsilon\, N$ is replaced with a term $\Upsilon^{\,k} N$ — an unwinding of the form of the form $N_1(\ldots N_k\Omega\ldots)$. It is now straightforward to show the following:

- If $[\![M]\!]_A \neq \perp$ then there exists some k such that $[\![M_k]\!]_A \neq \perp$.
- For all k it follows that $M_k \underset{\sim}{\sqsubseteq} M$.

We have already seen that every M_k is computable and so the above two fact imply computational adequacy:

$$\begin{aligned} [\![M]\!]_A \neq \perp &\Rightarrow [\![M_k]\!]_A \neq \perp \text{ for some } k \\ &\Rightarrow M_k, s\Downarrow_A \text{ for some } s \\ &\Rightarrow M, s\Downarrow_A \end{aligned}$$

■

Computational adequacy and the soundness of the model for evaluation together provide the required observational soundness result. Suppose $[\![M]\!]_A \subseteq [\![M']\!]_A$ and we are given any context $C[-]$ such that $C[M]\Downarrow_A$. By compositionality we must have $[\![C[M]]\!]_A \subseteq [\![C[M']]\!]_A$ and hence $[\![C[M']]\!]_A \neq [\![\Omega]\!]_A$ by soundness. By adequacy we have $C[M']\Downarrow_A$ and hence $M \underset{\sim}{\sqsubseteq} M'$.

4.4.3 Definability

Before we describe the definability result for $\mathbf{IA}_a$ from [6] we will look at the lluf subcategory of $\mathbf{A}^T_{\mathcal{L}}$ comprised of exactly the innocent strategies. It is straightforward to show that the semantics of any term that does not use the **new** or **do** construct is innocent. In other words the innocent strategies in $\mathbf{A}^T_{\mathcal{L}}$ correspond exactly to programs which make no use of store. The innocent strategies form a lluf subcategory of $\mathbf{A}^T_{\mathcal{L}}$ in which the functional language PCF can be soundly modelled. It was this innocent subcategory which yielded the first definability results for a model of PCF in [28, 39]. It is straightforward to show that the compact strategies, with respect to subset inclusion in the innocent subcategory of $\mathbf{A}^T_{\mathcal{L}}$, are exactly the strategies σ for which responses are made to a finite number of views; we will refer to such strategies as innocently compact. It is also straightforward to show that the compact strategies, with respect to subset inclusion in $\mathbf{A}^T_{\mathcal{L}}$, are exactly the strategies σ for which $\sigma_{\mathsf{threads}}$ is finite. It can now be shown that every compact strategy in our model is the denotation of some $\mathbf{IA}_a$ term. In common with many such results [25, 1, 30], this can be shown using a factorization argument; first proving definability for *innocent* strategies.

Lemma 211 (Innocent Definability) Given an innocent strategy, with finite view function,

$$\sigma : [\![A_1]\!]_A \times \ldots \times [\![A_{n-1}]\!]_A \to [\![A_n]\!]_A$$

such that each of the A_i are $\mathbf{IA}_a$ types, there exists a term

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash_A M : A_n$$

with semantics σ.

The reader can refer to [2, 4, 28, 36] for the proof.

It can now be shown that all strategies result from the composition of an innocent strategy and the stateful constant **new**.

Lemma 212 Any legal compact strategy $\sigma : 1 \to \mathsf{A}$ can be factorized into the strategy

$$\mathbf{new} : 1 \to ((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com}$$

and an innocently compact strategy

$$\sigma' : ((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com}) \to \mathsf{A}$$

such that $\mathsf{new}; \sigma' = \sigma$.

Lemmas 211 and 212 together yield the following result:

Proposition 213 Given a compact strategy,

$$\sigma : [\![A_1]\!]_A \times \ldots \times [\![A_{n-1}]\!]_A \to [\![A_n]\!]_A$$

such that each of the A_i are $\mathbf{IA}_a$ types, there exists a term

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash_A M : A_n$$

with semantics σ.

Proof First we curry the strategy to yield a strategy σ' for the arena:

$$1 \to [\![A_1]\!]_A \Rightarrow \ldots \Rightarrow [\![A_{n-1}]\!]_A \Rightarrow [\![A_n]\!]_A$$

We can then use the factorization result to yield an innocently compact strategy σ'' for the arena:

$$[\![((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com})]\!]_A \to [\![A_1]\!]_A \Rightarrow \ldots \Rightarrow [\![A_{n-1}]\!]_A \Rightarrow [\![A_n]\!]_A$$

We can then use the innocent definability lemma to define the following term with semantics σ'':

$$f : ((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com}) \vdash_A \lambda x_1^{A_1} \ldots . \lambda x_{n-1}^{A_{n-1}} . M : A_1 \Rightarrow \ldots \Rightarrow A_n.$$

We can now uncurry the original context and curry the function f to arrive at a term:

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash_A \lambda f^{((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com})} . M : ((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com}) \Rightarrow A$$

We can now apply this term to one which has the semantics of **new** to construct a term with semantics σ:

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash_A \lambda f^{((\mathbf{var} \Rightarrow \mathbf{com}) \Rightarrow \mathbf{com})} . M(\lambda g^{\mathbf{var} \Rightarrow \mathbf{com}} . \mathsf{new}\ y := 0\ \mathsf{in}\ gy) : A.$$

■

4.4.4 Full Abstraction

Full abstraction does not hold in $\mathbf{A}^T_{\mathcal{L}}$ but the definability result does facilitate the construction of a fully abstract model. We will first give some intuition as to why this is the case. It should be clear that given any open $\mathbf{IA}_a$ terms $x_1 : A_1, \ldots, x : A_{n-1} \vdash_A M : A_n$ and $y_1 : A_1, \ldots, y : A_{n-1} \vdash_A M' : A_n$ it follows that

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash_A M : A_n \sqsubseteq_{\sim} y_1 : A_1, \ldots, y : A_{n-1} \vdash_A M' : A_n$$

if and only if

$$\lambda x_1 \ldots \lambda x_{n-1}.M : A_n \sqsubseteq_{\sim} \lambda y_1 \ldots \lambda y_{n-1}.M' : A_n.$$

Furthermore, for any context $C[-] : \mathbf{com}$ we can create by abstraction an $\mathbf{IA}_a$ term $C' : A \Rightarrow \mathbf{com}$ such that for any closed $\mathbf{IA}_a$ term $M : A$ we have

$$C[M]\Downarrow_A \Leftrightarrow C'(M)\Downarrow_A.$$

There are only two legal strategies for the arena **com**: $[\![\mathsf{skip}]\!]_A$ and $[\![\Omega]\!]_A$. The point is that any strategy $\rho : \mathsf{A} \to \mathbf{com}$ can be thought of as a *test* for strategies for closed terms of type $[\![A]\!]_A$.

Definition 214 Given two strategies $\sigma : \mathsf{A}$ and $\tau : \mathsf{B}$ we say $\sigma \lesssim \tau$ if and only if for all $\rho : \mathsf{A} \to \mathbf{com}$ it follows that:

$$\sigma;\rho = \top \Rightarrow \tau;\rho = \top.$$

In other words $\sigma \lesssim \tau$ if and only if τ passes all the tests that σ does.

It is straightforward to show that if there exists some test that can distinguish a pair of strategies then there must exist a compact test that also distinguishes them. In other words, given two strategies $\sigma : \mathsf{A}$ and $\tau : \mathsf{B}$ then $\sigma \lesssim \tau$ if and only if for all *compact* $\rho : \mathsf{A} \to \mathbf{com}$ it follows that:

$$\sigma;\rho = \top \Rightarrow \tau;\rho = \top.$$

However we have already shown that all such tests, for the arenas in which we are interested, are the semantics of some $\mathbf{IA}_a$ term. Hence the following holds:

$$M \sqsubseteq_{\sim} N \Leftrightarrow [\![M]\!]_A \lesssim [\![N]\!]_A.$$

This is full abstraction — the definition exactly captures the observational preorder in denotational terms. It is usual to interpret the language $\mathbf{IA}_a$ in a quotient of $\mathbf{A}^T_{\mathcal{L}}$ written $\mathbf{A}^T_{\mathcal{L}}/\lesssim$, with the caveat that we have not shown that this collapsed category has the structure suggested of a model of $\mathbf{IA}_a$.

Chapter 5

Syntactic Control of Interference

5.1 Introduction

At the heart of imperative programming lies interference — the possibility that the evaluation of one program phrase may affect the later evaluation of another, by means of side-effects. Without this possibility, assigning values to variables can have no effect on the outcome of a program. It might seem to us that interference hampers our reasoning methods, both formal and informal, and is a source of programming error. However, whether we like it or not imperative programming has proved popular —perhaps interference is a good thing.

In *Syntactic Control of Interference* [47], Reynolds points out that problems really only arise when we encounter *covert interference* —where distinct identifiers are bound to phrases that access the same variable. This phenomena is known at ground type as *aliasing* and we encountered an example in chapter 1:

$$\mathsf{new}\ y := 0\ \mathsf{in}\ (\lambda x.x := 5; y := 2 \times !x)(y).$$

If we look inside the body of the procedure we see the subterm $x := 5; y := 2 \times !x$ and we might imagine that x will contain the value 5 and y the value 10. However, aliasing has resulted in both identifiers being bound to the same storage cell and hence they "both" contain the value 10. It is only when we have two language features in place, store and identifier binding (in the form of procedures) that we encounter this covert interference.

In chapter 1 we mentioned Reynolds' elegant solution to this problem [47]. He introduces the notion of a non-interference relation, $\sharp$, between program phrases with the intention that the relation holds between program fragments if it is *syntactically* obvious that they do not interfere.

- If $M \sharp N$ and M and N are not procedures then exercising M will have no effect on the exercise of N.
- If M is a procedure, with arguments $A_1, \ldots A_n$ such that $M \sharp N \wedge A_1 \sharp N \wedge \ldots \wedge A_n \sharp N$, then $M(A_1, \ldots, A_n) \sharp N$. That is, $M \sharp N$ implies that an application of M will not affect N if its arguments do not affect N.

As mentioned in chapter 1, the above does not form a definition of $\sharp$, but some minimum requirements. Given such a relation, Reynolds then proposes the following design principles for an SCI language:

SCI1 If $x\sharp y$ for all free identifiers x in M and y in N then $M\sharp N$. This simply states that all channels of interference are named by identifiers: there are no constants or programming constructs that interfere with each other.

SCI2 If x and y are distinct identifiers then $x\sharp y$.

SCI3 Terms of certain types are passive in that they do not assign to any global store. If M and N are both passive then $M\sharp N$.

Principle **SC1** is already in place in Algol like languages —there are no global channels by which one subphrase may affect another. The other criteria are maintained by restricting any application $M(N)$ to only those phrases such that $M\sharp N$. By ensuring that a function and its argument do not interfere we can ensure that N will not become bound to any identifier that can interfere with other identifiers in the body of M. Reynolds suggests an elegant but restrictive definition for the $\sharp$ relation as follows

$$M\sharp N \Leftrightarrow \mathrm{FV}(M)\cap\mathrm{FV}(N)=\emptyset.$$

That is, the relation holds between M and N if and only if they have no free identifiers in common. This is exactly the definition that Reynolds adopts in the first illustrative language he proposes in [47] for which the name Basic SCI (in this thesis $\mathbf{SCI}_b$ for short) was coined in [43].

In modern terms, Reynolds's restriction may be thought of as the imposition of a *multiplicative* application rule, i.e. a typing rule for procedure calls in which the function and argument have distinct contexts, and the elimination of the structural rule of contraction.

5.1.1 Passivity

Note that Reynolds' first definition of the $\sharp$ relation is particularly restrictive —it does not take into account the third of his design principles. As mentioned in chapter 4, Reynolds suggests, in [50], that a specified subset of types be passive —terms of passive type cannot change the store. The relaxation of the constraints for passive types in the third design principle amounts to the reintroduction of contraction at those types. Thus SCI has some of the flavour of linear logic [23] with passivity appearing somewhat analogous to the linear exponential, or !, logical connective.

Reynolds does propose a less restrictive definition of $\sharp$ in [47]. After identifying a set of passive types he defines for each term M the set $A(M)$ of free identifiers that occur at least once in M outside of any phrase of passive type. For example, suppose that command types are active and expression types are passive, then given the following judgement for a term M:

$$x:\mathbf{var}, y:\mathbf{var}\vdash x:=!y$$

we see that $A(M)=x$. Note that y is not included in $A(M)$ as its sole occurrence is in the passive subphrase $!y$. Reynolds then defines the $\sharp$ relation as follows:

$$M\sharp N \Leftrightarrow (A(M)\cap FV(N)=\emptyset)\wedge(A(N)\cap FV(M)=\emptyset).$$

This treatment, however, gives rise to problems —the subject reduction property is lost. Consider the following term:

$$x : \mathbf{var}, y : \mathbf{var} \vdash (\lambda w^{\mathbf{exp}}.\pi_1\langle 3,\ \ (\lambda z^{\mathbf{exp}}.y :=!y+1; x := z)\ x\rangle)\ !y.$$

This term is valid in Reynolds' system as the function and its argument are in the $\sharp$ relation because the use of y in both the function and its argument lie within passive expressions. However β-reduction yields the following:

$$x : \mathbf{var}, y : \mathbf{var} \vdash \pi_1\langle 3, (\lambda z^{\mathbf{exp}}.y :=!y+1; x := z)\,!y\rangle.$$

which is clearly not valid as the subterm $(\lambda z^{\mathbf{exp}}.y :=!y+1; x := z)\,!y$ has a function and argument that are clearly not in the $\sharp$ relation — the identifier y is used actively in the function. Note that this problem goes away when we reduce further —they program does after all obey Reynolds' SCI principles. This poses a problem for any type system that adheres to the design SCI principles, one of the following must be true of any such type system:

- The term $x : \mathbf{var}, y : \mathbf{var} \vdash \pi_1\langle 3, (\lambda z^{\mathbf{exp}}.y :=!y+1; x := z)\,!y\rangle$ should not be typeable. This seems unnecessarily restrictive as the program conforms to Reynolds' principles.
- The type system does not obey subject reduction.
- A term may be typeable despite having untypeable subterms. Clearly the term

  $$x : \mathbf{var}, y : \mathbf{var} \vdash (\lambda z^{\mathbf{exp}}.y :=!y+1; x := z)\,!y$$

  should never be typeable if the system is to conform to Reynolds' principles but perhaps the term

  $$x : \mathbf{var}, y : \mathbf{var} \vdash \pi_1\langle 3, (\lambda z^{\mathbf{exp}}.y :=!y+1; x := z)\,!y\rangle.$$

  should be typeable.

For the rest of this chapter we will recall different languages which adhere to the SCI design principles in different ways.

5.2 PCF

At one end of the spectrum of languages that adhere to the SCI design principle is **PCF**. As **PCF** is a purely functional language, it makes no use of state and thus all interference is banished. We might consider that it conforms to an extreme example of Reynolds' second definition of $\sharp$ in which all types are passive.

PCF has a long and distinguished history in theoretical computer science. It is considered the paradigm of sequential functional languages.

PCF

M	$::=$	n	numerals
	$\mid$	MM	function application
	$\mid$	$\lambda x^{\theta}.M$	function abstraction
	$\mid$	$\langle M, M \rangle$	pairing
	$\mid$	$\pi_1 M$	left projection
	$\mid$	$\pi_2 M$	right projection
	$\mid$	$\mathsf{if}\ M = 0\ \mathsf{then}\ M\ \mathsf{else}\ M$	conditional
	$\mid$	$\mathsf{succ}\ M$	successor
	$\mid$	$\mathsf{pred}\ M$	predecessor

Table 5.1: Term Grammar for PCF

PCF

$$\frac{}{\Gamma, x : \theta \vdash_P x : \theta}\ \mathit{Axiom} \qquad \frac{\Gamma \vdash_P P : \theta_0 \quad \Gamma \vdash_P Q : \theta_1}{\Gamma \vdash_P \langle P, Q \rangle : \theta_0 \times \theta_1} \times I$$

$$\frac{\Gamma \vdash_P \langle P, Q \rangle : \theta \times \theta'}{\Gamma \vdash_P \pi_1 P : \theta} \times E_1 \qquad \frac{\Gamma \vdash_P \langle P, Q \rangle : \theta \times \theta'}{\Gamma \vdash_P \pi_2 P : \theta'} \times E_2$$

$$\frac{\Gamma, x : \theta' \vdash_P P : \theta}{\Gamma \vdash_P \lambda x : \theta'.P : \theta' \multimap \theta} \multimap I \qquad \frac{\Gamma \vdash_P P : \theta' \multimap \theta \quad \Gamma \vdash_P Q : \theta'}{\Gamma \vdash_P P(Q) : \theta} \multimap E$$

Table 5.2: Typing Rules for the λ-calculus fragment of PCF.

5.2.1 The PCF Type System

The types, θ, are constructed from primitive types τ as follows.

$$\theta ::= \tau | \theta \times \theta | \theta \multimap \theta$$

For our illustrative language we will define τ as follows.

$$\tau ::= \mathbf{N}$$

Terms are constructed using the grammar defined in table 5.1.

5.2.2 Operational Semantics

The language $\mathbf{IA}_a$ is an *extension* of **PCF**. As such every term judgement of **PCF** is also a term judgement in $\mathbf{IA}_a$. In other words it is simple to show that:

$$\Gamma \vdash_P M : A \Rightarrow \Gamma \vdash_A M : A$$

PCF

$$\frac{}{\vdash_P n : \mathbf{N}} \qquad \frac{\Gamma \vdash_P M : \mathbf{N}}{\Gamma \vdash_P \mathsf{succ}\, M : \mathbf{N}}$$

$$\frac{\Gamma \vdash_P M : \mathbf{N}}{\Gamma \vdash_P \mathsf{pred}\, M : \mathbf{N}} \qquad \frac{\Gamma \vdash_P L : \mathbf{N} \quad \Gamma \vdash_P M : \tau \quad \Gamma \vdash_P N : \tau}{\Gamma \vdash_P \mathsf{if}\, L = 0\, \mathsf{then}\, M\, \mathsf{else}\, N : \tau}$$

Table 5.3: Typing Rules for PCF Language Constructs

The operational semantics of **PCF** is call by name and is typically defined, in the big step style, via a relation $\Downarrow_P$ from closed **PCF** terms to **PCF** values. Given a well-typed closed **PCF** term $\vdash_P M : A$ we choose the following definition, using the semantics of $\mathbf{IA}_a$:

$$(M : A \Downarrow_P V : A) \Leftrightarrow (s, M : A \Downarrow_A s', V : A).$$

It is simple to show that this definition is equivalent to the standard operational semantics of **PCF** and this will be useful when we prove soundness and adequacy for our games model of **PCF**. The definition does not depend on the stores s, s' so we may insist that they are the empty store.

5.2.3 Models of PCF

As with $\mathbf{IA}_a$ it is the case that **PCF** consists of the simply-typed λ-calculus as a basis to which ground types and language constants are added. Hence we should aim to interpret **PCF** in a cartesian-closed category.

The *standard model* of **PCF** [44], as it has come to be known, comprising of continuous functions on partial orders, fails to capture the sequential nature of **PCF** and hence there is no definability result for this model. However, Plotkin shows that if a parallel "or" operator is added to the language then the standard model is fully abstract [44]. It was not until the remarkable success of game semantics that models with appropriate definability results, yielded the first fully abstract models of **PCF** [28, 4, 39]. The innocent subcategory of $\mathbf{A}^T_{\mathcal{L}}$ that we defined in chapter 4 is simply a reworking of these models and resembles [28] most closely in style.

5.3 Basic SCI

Basic SCI, henceforth $\mathbf{SCI}_b$, is an example of a language that also follows the SCI design principles. It is at the other end of the spectrum to **PCF** in that no type is deemed passive and thus all aliasing is banned. This language was first presented in [47] and follows from Reynolds' initial definition of the $\sharp$ relation. Note that all types can have side effects —there is no notion of passivity. Note too that we do not have a general fixed point constructor as it is straightforward to show that this would lead to subject reduction problems,

$$\mathsf{Y}\, f \rightsquigarrow f(\mathsf{Y}\, f),$$

hence all recursion in the language is via the $(\mathsf{while} - = 0\, \mathsf{do} -)$ construct.

SCI$_b$

M	::=	n	numerals
	|	skip	null command
	|	MM	function application
	|	$\lambda x^{\theta}.M$	function abstraction
	|	$\langle M, M \rangle$	pairing
	|	$\pi_1 M$	left projection
	|	$\pi_2 M$	right projection
	|	$M;M$	sequential composition
	|	if $M = 0$ then M else M	conditional
	|	new $M := 0$ in M	variable allocation
	|	$M := M$	assignment
	|	$!M$	dereference
	|	mkvar M M	bad variable constructor
	|	succ M	successor
	|	pred M	predecessor
	|	while $M = 0$ do M	looping

Table 5.4: Term Grammar for **SCI$_b$**

5.3.1 The SCI$_b$ Type System

The types, θ, are constructed from primitive types τ as follows.

$$\theta ::= \tau | \mathbf{var} | \theta \times \theta | \theta \multimap \theta$$

For our illustrative language we will define τ as follows.

$$\tau ::= \mathbf{N} | \mathbf{com}$$

Terms are constructed using the grammar defined in table 5.4.

5.3.2 Operational Semantics

Our discussion of the operational semantics of **SCI$_b$** will be similar to that for **PCF**. The language **IA$_a$** is also an extension of **SCI$_b$** so every term judgement of **SCI$_b$** is also a term judgement in **IA$_a$**. In other words it is simple to show that:

$$\Gamma \vdash_B M : A \Rightarrow \Gamma \vdash_A M : A$$

The operational semantics of **SCI$_b$** is call by name and is typically defined, in the big step style, via a relation $\Downarrow_B$ from closed **SCI$_b$** terms to **SCI$_b$** values. Given **SCI$_b$** term $M : A$ well typed in a **var** store, we choose the following definition, using the semantics of **IA$_a$**:

$$(s, M : A \Downarrow_B s', V : A) \Leftrightarrow (s, M : A \Downarrow_A s', V : A).$$

It is simple to show that this definition is equivalent to the standard operational semantics of **SCI$_b$** and again this will prove useful in our soundness and adequacy proofs.

SCI$_b$

$$\frac{}{\Gamma, x:\theta \vdash_B x:\theta}\ \textit{Axiom} \qquad \frac{\Gamma \vdash_B P:\theta_0 \quad \Gamma \vdash_B Q:\theta_1}{\Gamma \vdash_B \langle P,Q\rangle : \theta_0 \times \theta_1}\ \times I$$

$$\frac{\Gamma \vdash_B \langle P,Q\rangle : \theta \times \theta'}{\Gamma \vdash_B \pi_1 P:\theta}\ \times E_1 \qquad \frac{\Gamma \vdash_B \langle P,Q\rangle : \theta \times \theta'}{\Gamma \vdash_B \pi_2 P:\theta'}\ \times E_2$$

$$\frac{\Gamma, x:\theta' \vdash_B P:\theta}{\Gamma \vdash_B \lambda x:\theta'.P:\theta' \multimap \theta}\ \multimap I \qquad \frac{\Gamma \vdash_B P:\theta' \multimap \theta \quad \Delta \vdash_B Q:\theta'}{\Gamma,\Delta \vdash_B P(Q):\theta}\ \multimap E$$

Table 5.5: Typing Rules for the Affine λ-calculus fragment of **SCI$_b$**.

SCI$_b$

$$\frac{}{\vdash_B n:\mathbf{N}} \qquad \frac{}{\vdash_B \mathsf{skip}:\mathbf{com}}$$

$$\frac{\Gamma \vdash_B M:\mathbf{N}}{\Gamma \vdash_B \mathsf{succ}\, M:\mathbf{N}} \qquad \frac{\Gamma \vdash_B M:\mathbf{N}}{\Gamma \vdash_B \mathsf{pred}\, M:\mathbf{N}}$$

$$\frac{\Gamma \vdash_B M:\mathbf{com} \quad \Gamma \vdash_B N:\tau}{\Gamma \vdash_B M;N:\tau} \qquad \frac{\Gamma \vdash_B M:\mathbf{N} \quad \Gamma \vdash_B N:\mathbf{com}}{\Gamma \vdash_B \mathsf{while}\, M=0\, \mathsf{do}\, N:\mathbf{com}}$$

$$\frac{\Gamma \vdash_B L:\mathbf{N} \quad \Gamma \vdash_B M:\tau \quad \Gamma \vdash_B N:\tau}{\Gamma \vdash_B \mathsf{if}\, L=0\, \mathsf{then}\, M\, \mathsf{else}\, N:\tau} \qquad \frac{\Gamma \vdash_B M:\mathbf{N} \quad \Gamma \vdash_B N:\mathbf{N} \Rightarrow \mathbf{com}}{\Gamma \vdash_B \mathsf{mkvar}\, M\, N:\mathbf{var}}$$

$$\frac{\Gamma \vdash_B M:\mathbf{var} \quad \Gamma \vdash_B N:\mathbf{N}}{\Gamma \vdash_B M:=N:\mathbf{com}} \qquad \frac{\Gamma \vdash_B M:\mathbf{var}}{\Gamma \vdash_B !M:\mathbf{N}}$$

$$\frac{\Gamma, x:\mathbf{var} \vdash_B M:\tau}{\Gamma \vdash_B \mathsf{new}\, x:=0\, \mathsf{in}\, M:\tau}$$

Table 5.6: Typing Rules for the Algol-Like Constructs of **SCI$_b$**

5.3.3 Models of $\mathbf{SCI}_b$

It is, of course, possible to model $\mathbf{SCI}_b$ in a cartesian-closed category. Indeed it is straightforward to show that we can give an operationally sound interpretation of $\mathbf{SCI}_b$ in the category $\mathbf{A}^T_{\mathcal{L}}$. However such an interpretation will necessitate an amount of "junk" in the model: morphisms for which there can be no denotation. For example there can never be a $\mathbf{SCI}_b$ term

$$x : \mathbf{com} \vdash_B M : \mathbf{com} \times \mathbf{com}$$

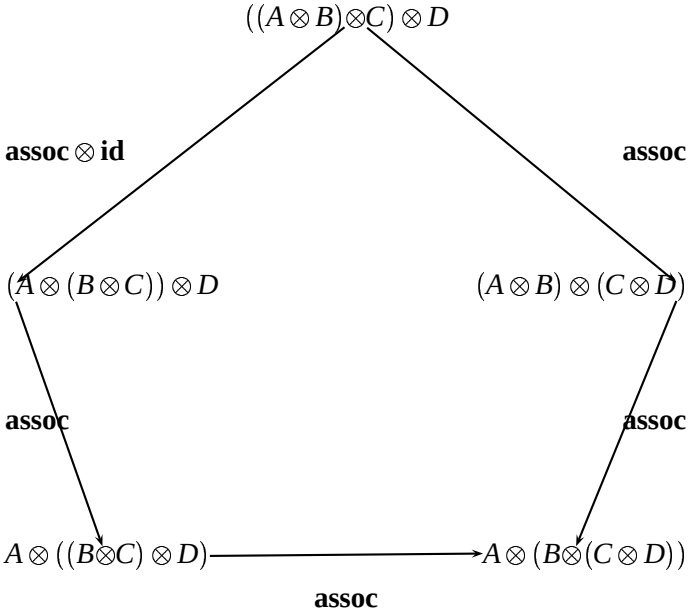
that has the contraction map as its denotation. Instead we allow a category with more general structure. We still insist that our model be equipped with products and we use these when we interpret products in our language and also when we interpret the language constants that are typed in the additive style. We will leave our treatment of these products until we give a concrete definition of a category with the relevant structure in chapter 6. For the rest of this section we will discuss the interpretation of the affine λ-calculus fragment of $\mathbf{SCI}_b$.

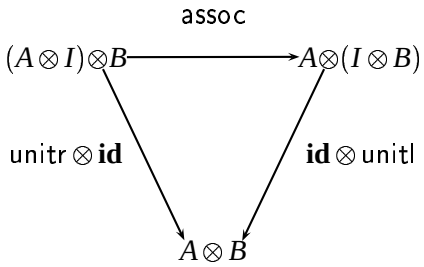
5.3.4 Monoidal Categories

In this section we recall the definition of a monoidal category from [34]. A monoidal category is specified by a tuple $\langle C, \otimes, I, \mathsf{assoc}, \mathsf{unitl}, \mathsf{unitr}\rangle$ consisting of a category C, a bifunctor $\otimes : C \times C \to C$, an object I in C and natural transformations

- $\mathsf{assoc}_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$
- $\mathsf{unitl}_A : I \otimes A \to A$
- $\mathsf{unitr}_A : A \otimes I \to A$

such that $\mathsf{unitl}_I = \mathsf{unitr}_I : I \otimes I \to I$ and the following diagrams commute for all objects A, B, C and D.

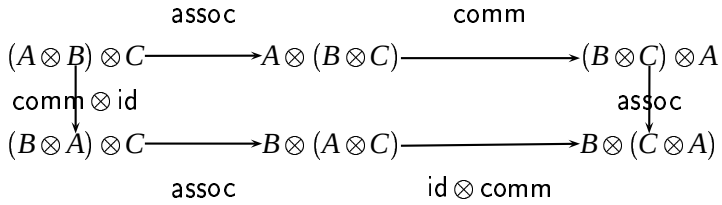




A symmetric monoidal category is a monoidal category with the further natural isomorphism

$$\mathsf{comm}_{A,B} : A \otimes B \to B \otimes A$$

such that $\mathsf{unitl} = \mathsf{comm}; \mathsf{unitr}$ and $\mathsf{comm} : \mathsf{comm} = \mathsf{id}$ and the following diagram commutes.



Henceforth the $\otimes$ symbol can be read as right-associative.

A symmetric monoidal closed category is a symmetric monoidal category in which every functor $- \otimes B$ has a specified right adjoint, the exponential, which is written $B \multimap -$. This induces a natural isomorphism, the currying operation, between the hom-sets

$$C(A \otimes B, C) \simeq C(A, B \multimap C).$$

A categorical model of $\mathbf{SCI}_b$ will ideally be a symmetric monoidal closed category (SMCC) with products. We further require that the unit of the monoid I, be the terminal object 1. However we should note that it is not always possible to have such structure.

5.3.5 The Interpretation of $\mathbf{SCI}_b$ in an SMCC

In this section we will explain how $\mathbf{SCI}_b$ can be modelled in an affine SMCC. Each $\mathbf{SCI}_b$ type A will be modelled in C by an object $[\![A]\!]$. We start by defining objects corresponding to each of the base types: $[\![\mathbf{N}]\!]$,$[\![\mathbf{com}]\!]$ and $[\![\mathbf{var}]\!]$. The interpretation of higher types is defined inductively:

$$[\![A \times B]\!] = [\![A]\!] \times [\![B]\!] \text{ and } [\![A \multimap B]\!] = [\![A]\!] \multimap [\![B]\!].$$

A typed term

$$x_1 : A_1, \ldots, x_n : A_n \vdash_B M : A$$

will be modelled by a morphism

$$[\![A_1]\!] \otimes \ldots \otimes [\![A_n]\!] \to [\![A]\!]$$

and a closed term $\vdash_B M : A$ will be modelled by a morphism $1 \to [\![A]\!]$. The λ-calculus fragment of $\mathbf{SCI}_b$ is interpreted using the affine symmetric monoidal closed structure. Identifiers are interpreted as projections:

$$[\![x_1 : A_1, \ldots, x_n : A_n \vdash_B x_i : A_i]\!] = \pi_i^{\otimes} : [\![A_1]\!] \otimes \ldots \otimes [\![A_n]\!] \to [\![A_i]\!].$$

Abstraction is modelled by currying:

$$[\![\Gamma \vdash_B M : A \multimap B]\!] = \Lambda([\![\Gamma, x : A \vdash_B M : B]\!]) : [\![\Gamma]\!] \to [\![A \multimap B]\!].$$

Application is modelled by composition with the evaluation map:

$$[\![\Gamma, \Delta \vdash_B MN : B]\!] = [\![\Gamma \vdash_B M : A \multimap B]\!] \otimes [\![\Delta \vdash_B N : A]\!]; \mathsf{eval}_{[\![A \multimap B]\!]}.$$

Pairing is modelled using the pairing map:

$$[\![\Gamma \vdash_B \langle M, N \rangle : A \times B]\!] = \langle [\![\Gamma \vdash_B M : A]\!], [\![\Gamma \vdash_B N : B]\!] \rangle.$$

Projection is interpreted by composition with the projection map:

$$[\![\Gamma \vdash_B \pi_i N : A]\!] = [\![\Gamma \vdash_B N : A \times B]\!]; \pi_i.$$

Lastly, in a categorical model of $\mathbf{SCI}_b$ we will require morphisms with which to model the Algol-like language constructs. We will only give the example of sequential composition of commands here, but see chapter 6 for a concrete example of how all the constructs may be interpreted. We must specify a map in our SMCC

$$\mathsf{seq} : [\![\mathbf{com}]\!] \times [\![\mathbf{com}]\!] \to [\![\mathbf{com}]\!]$$

and we can then define the semantics of sequential composition of commands as follows:

$$[\![\Gamma \vdash_B M; N]\!] = \langle [\![\Gamma \vdash_B M]\!], [\![\Gamma \vdash_B N]\!] \rangle; \mathbf{seq}_\alpha.$$

5.3.6 Concrete Models

A model of $\mathbf{SCI}_b$ that is of particular interest is Reddy's object spaces model [46] which was a precursor to the field of game semantics. In [37] McCusker gives a different presentation of Reddy's model and shows it to be fully abstract — thus demonstrating that Reddy's model was the first fully abstract model of a higher order imperative language.

We present a games model of $\mathbf{SCI}_b$ in chapter 6.

5.4 SCIR

In *Syntactic Control of Interference Revisited* [40], O'Hearn *et al.* proposed a type system, **SCIR**, that adheres to Reynolds' SCI principles. In this elegant system, contexts are split into two zones, active and passive, and contraction is permitted only in the passive zone. This idea is familiar from linear logic, in particular Barber and Plotkin's DILL [12] and has since been applied to other programming languages [11, 54]. The notion of passive type in **SCIR** is sophisticated. These zones show the distinction between passive types and *passive use*. Phrases of passive type are essentially side-effect free when viewed externally, they do not write to non-local store, but some phrases of passive type do affect the store locally. Free identifiers are used passively unless they have an occurrence outside of any subterm of passive type. This use of zones permits the construction of programs that contain untypeable subterms and thus permits the problematic intermediate terms

SCIR

M	::=	x	identifiers
	|	n	numerals
	|	skip	null command
	|	MM	function application
	|	$\lambda x^{\theta}.M$	function abstraction
	|	$\langle M, M \rangle$	pairing
	|	$\pi_1 M$	left projection
	|	$\pi_2 M$	right projection
	|	$M;M$	sequential composition
	|	if $M = 0$ then M else M	conditional
	|	new $M := 0$ in M	variable allocation
	|	do $M := 0$ then M	variable allocation
	|	$M := M$	assignment
	|	$!M$	dereference
	|	mkvar M M	bad variable constructor
	|	succ M	successor
	|	pred M	predecessor
	|	Y M	fixed point combinator
	|	promote M	promotion
	|	derelict M	dereliction

Table 5.7: Term Grammar for **SCIR**.

that lead to the failure of subject reduction in [47]. However, it is simple to show, by induction on typing derivation, that all subterms of an **SCIR** term that contain no free identifiers are themselves typeable in **SCIR**.

A type reconstruction algorithm for **SCIR** was given in [55].

5.4.1 The SCIR Type System

The **SCIR** types, θ, are constructed from primitive types τ as follows.

$$\theta ::= \tau | \theta \times \theta | \theta \multimap \theta | \mathsf{P}\theta$$

We should note here that we omit the non-interfering product types that occur in the original presentation of the **SCIR** language [40] as it is unclear how these would be interpreted in our model. A subset of the types are referred to as *passive*. Passive types ϕ are generated by the grammar

$$\phi ::= \phi \times \phi | \theta \multimap \phi | \mathsf{P}\theta.$$

Terms are constructed using the grammar defined in table 5.7. The terms of the system and their typing rules are shown in Figure 5.8.

SCIR

$$\frac{}{|x:\theta \vdash_S x:\theta}\ \text{Axiom} \qquad \frac{\Gamma|x:\theta,\Delta \vdash_S P:\phi}{\Gamma,x:\theta|\Delta \vdash_S P:\phi}\ \text{Passification}$$

$$\frac{\Gamma,x:\theta|\Delta \vdash_S P:\theta}{\Gamma|x:\theta,\Delta \vdash_S P:\theta}\ \text{Activation} \qquad \frac{\Gamma|\Delta \vdash_S P:\theta}{\Gamma,\Gamma'|\Delta,\Delta' \vdash_S P:\theta}\ \text{Weakening}$$

$$\frac{\Gamma,x:\theta,y:\theta|\Delta \vdash_S P:\theta'}{\Gamma,x:\theta|\Delta \vdash_S P[y/x]:\theta'}\ \text{Contraction} \qquad \frac{\Gamma|\Delta \vdash_S P:\theta_0 \quad \Gamma|\Delta \vdash_S Q:\theta_1}{\Gamma|\Delta \vdash_S \langle P,Q\rangle:\theta_0\times\theta_1}\ \times\mathbf{I}$$

$$\frac{\Gamma|\Delta \vdash_S P:\theta_0\times\theta_1}{\Gamma|\Delta \vdash_S \pi_i P:\theta_i}\ \times\mathbf{E}_i(i=0,1) \qquad \frac{\Gamma|\Delta,x:\theta' \vdash_S P:\theta}{\Gamma|\Delta \vdash_S \lambda x:\theta'.P:\theta'\multimap\theta}\ \multimap\mathbf{I}$$

$$\frac{\Gamma|\Delta \vdash_S P:\theta'\multimap\theta \quad \Gamma'|\Delta' \vdash_S Q:\theta'}{\Gamma,\Gamma'|\Delta,\Delta' \vdash_S P(Q):\theta}\ \multimap\mathbf{E} \qquad \frac{\Gamma| \vdash_S Q:\theta}{\Gamma| \vdash_S \mathbf{promote}(Q):\mathsf{P}\theta}\ P\mathbf{I}$$

$$\frac{\Gamma|\Delta \vdash_S Q:\mathsf{P}\theta}{\Gamma|\Delta \vdash_S \mathbf{derelict}(Q):\theta}\ P\mathbf{E}$$

Table 5.8: λ-calculus and Structural Rules for **SCIR**

SCIR

$$\frac{}{|\ \vdash_S n:\mathbf{N}} \qquad \frac{}{|\ \vdash_S \mathsf{skip}:\mathbf{com}}$$

$$\frac{\Gamma|\Delta \vdash_S M:\mathbf{N}}{\Gamma|\Delta \vdash_S \mathsf{succ}\,M:\mathbf{N}} \qquad \frac{\Gamma|\Delta \vdash_S M:\mathbf{N}}{\Gamma|\Delta \vdash_S \mathsf{pred}\,M:\mathbf{N}}$$

$$\frac{\Gamma|\Delta \vdash_S M:\mathbf{com} \quad \Gamma|\Delta \vdash_S N:\mathbf{com}}{\Gamma\mid\Delta \vdash_S M;N:\mathbf{com}} \qquad \frac{\Gamma\mid\Delta \vdash_S M:\mathsf{P}(\theta\Rightarrow\theta)}{\Gamma\mid\Delta \vdash_S \Upsilon\,M:\theta}$$

$$\frac{\Gamma|\Delta \vdash_S L:\mathbf{N} \quad \Gamma|\Delta \vdash_S M:\tau \quad \Gamma|\Delta \vdash_S N:\tau}{\Gamma|\Delta \vdash_S \mathsf{if}\ L=0\ \mathsf{then}\ M\ \mathsf{else}\ N:\tau} \qquad \frac{\Gamma|\Delta \vdash_S M:\mathbf{var} \quad \Gamma|\Delta \vdash_S N:\mathbf{N}}{\Gamma|\Delta \vdash_S M:=N:\mathbf{com}}$$

$$\frac{\Gamma|\Delta \vdash_S M:\mathbf{N} \quad \Gamma|\Delta \vdash_S N:\mathbf{N}\Rightarrow\mathbf{com}}{\Gamma|\Delta \vdash_S \mathsf{mkvar}\,M\,N:\mathbf{var}} \qquad \frac{\Gamma|\Delta \vdash_S M:\mathbf{var}}{\Gamma|\Delta \vdash_S !M:\mathbf{N}}$$

$$\frac{\Gamma|\Delta,x:\mathbf{var} \vdash_S M:\mathbf{com}}{\Gamma|\Delta \vdash_S \mathsf{new}\ x:=0\ \mathsf{in}\ M:\mathbf{com}} \qquad \frac{\Gamma|x:\mathbf{var} \vdash_S M:\mathbf{com}}{\Gamma|\ \vdash_S \mathsf{do}\ x:=0\ \mathsf{then}\ M:\mathbf{N}}$$

Table 5.9: Typing Rules for the Algol-Like Language Constructs of **SCIR**.

As in [40], we will consider an illustrative programming language based on this type system. For our base types, τ, we have the natural numbers **N**, commands **com** and variables **var**. We shall only use **N** passively so we will define the shorthand $\mathbf{exp} = \mathsf{PN}$. The language is equipped with a range of constants for imperative programming, including assignment and dereferencing of variables, sequential composition, conditionals, and two variable-allocation constructs:

- $\mathsf{new} - := 0 \,\mathsf{in}\, -$.
- $\mathsf{do} - := 0 \,\mathsf{then}\, -$.

The operational reading of these is as follows. The command $\mathsf{new}\, x := 0 \,\mathsf{in}\, M$ is executed by allocating a fresh storage variable, binding x to this variable, and then executing M. The expression $\mathsf{do}\, x := 0 \,\mathsf{then}\, M$ again binds x to a fresh storage variable and executes M, and then returns the final value stored in x. Note that the empty active zone in the typing rule for this construct ensures that M has no side-effects except for writes to x, so the overall expression is side-effect free.

The language also admits a recursion construct

$$Y_\theta : P(\theta \multimap \theta) \multimap \theta.$$

Note again the use of a passive type, this time to ensure that unfolding the recursion does not violate the constraints on application. We write Ω_θ for a divergent term at type θ.

5.4.2 Operational Semantics

SCIR has a call-by-name operational semantics defined in the standard fashion via a big step relation, as for $\mathbf{IA}_a$. Given terms M and V that are typeable in a **var** context we intend

$$M, s \Downarrow_S V, s'$$

to mean that term M with store s converges to value V with store s'. The semantics is given in terms of *stores* which are functions from locations to values of type **exp**. An evaluation relation of the form $s, M \Downarrow s', V$, where s and s' are stores and M and V are terms, is defined inductively; we omit the definition here. When M is a closed term we write $M \Downarrow V$ to indicate that $s, M \Downarrow s, V$ where s is the empty store. We write simply $M \Downarrow$ if there exists some value V such that $M \Downarrow V$. The term constructors **promote** and **derelict** have no operational effect.

Definition 215 We define the **observational preorder** $\underset{\sim}{\sqsubset}$ on typed terms. Given $\Gamma|\Delta \vdash_S M$ and $\Gamma|\Delta \vdash_S M'$, we write $\Gamma|\Delta \vdash_S M \underset{\sim}{\sqsubset} M'$, if and only if for any context $C[-]$ such that $\vdash_S C[M] : \mathbf{com}$ and $\vdash_S C[M'] : \mathbf{com}$, we have $C[M] \Downarrow$ implies $C[M'] \Downarrow$.

Our definition of the operational semantics for **SCIR** is slightly more difficult than that for **PCF** and $\mathbf{SCI}_b$. The language $\mathbf{IA}_a$ is not exactly an extension of **SCIR**. The language **SCIR** has more syntax in its types, its terms and its judgements. However we can define a very simple translation,

from **SCIR** types and contexts to $\mathbf{IA}_a$ types and contexts as follows:

$$\begin{aligned}
\tau^* &= \tau \\
(A \times B)^* &= A^* \times B^* \\
(A \multimap B)^* &= A^* \Rightarrow B^* \\
(\mathsf{P}A)^* &= A^* \\
(A_1, \ldots, A_{k-1} \mid A_k, \ldots, A_n)^* &= A_1^*, \ldots, A_{k-1}^*, A_k^*, \ldots, A_n^*
\end{aligned}$$

We overload the translation to map terms of **SCIR** to terms of $\mathbf{IA}_a$ by stripping away the derelict and promote term constructors. It is now simple to show that:

$$\Gamma \mid \Delta \vdash_S M : A \Rightarrow (\Gamma \mid \Delta)^* \vdash_A M^* : A^*$$

Lemma 216 Given a well-typed closed **SCIR** term $\vdash_S M : A$ it is straightforward to show the following:

$$(M : A \Downarrow_S V : A) \Rightarrow (M^* : A^* \Downarrow_A V^* : A^*).$$

and

$$(M^* : A^* \Downarrow_A V : A^*) \Rightarrow (\exists V' : A. V = V'^* \wedge M : A \Downarrow_S V' : A).$$

5.4.3 Models of SCIR

In this section we detail the categorical structure that we might expect from a model of **SCIR**.

A notion of categorical model of **SCIR**, bireflective categories, was proposed by O'Hearn *et al.* in [40], and a concrete functor-category model was defined. Bireflective categories are studied in [18] and have proved interesting in their own right. In [38] McCusker proposes a more general notion of a model of **SCIR** which we will now outline. McCusker's definition of a categorical model of **SCIR** is a category that possesses the following structure [38]:

- **C** has symmetric monoidal structure $\langle C, \otimes, I, \mathsf{assoc}, \mathsf{unitl}, \mathsf{unitr}, \mathsf{comm} \rangle$.
- **C** has finite products. Written $\times$ with terminal object 1.
- **C** has a full subcategory **P** of *passive objects* with inclusion $J : \mathbf{P} \hookrightarrow \mathbf{C}$. A passive type will therefore be interpreted as an object of the form JX.
- The inclusion functor J has both a left and right adjoint:

  $$S \dashv J \dashv P.$$

  We write η^S and ε^S for the unit and counit of $S \dashv J$ and we write η^P and ε^P for the unit and counit of $J \dashv P$.
- The functors S and J must be strong monoidal: we ask for natural isomorphisms

  $$s : SA \times SB \to S(A \otimes B) \qquad s' : 1 \to SI$$

  and

  $$j : JA \otimes JB \to J(A \times B) \qquad j' : I \to J1.$$

  To interpret contraction at passive types, it can be shown that that the monoidal structure of **C** restricts to a product structure on **P** and that I is terminal so weakening can be interpreted [38].

- In order to model application and abstraction we ask that **C** be an SMCC, or more minimally, that for any **C** object X and any **SCIR** type A we ask that the exponential $X \multimap [\![A]\!]$ exists.

To give a concrete model of **SCIR**, we must provide categories **C** and **P** which possess the aforementioned structure, together with a chosen object for each base type, which should be in the class of exponentiable objects, and a chosen morphism to interpret each of the constants. McCusker calls such a model a *categorical model of SCIR*.

We now show how such a model is used to interpret **SCIR**. Note that we simplify some of the definitions by sometimes identifying a pair of objects that we know are isomorphic.

As usual, an **SCIR** type A will be interpreted as an object $[\![A]\!]$ in **C**. We start by specifying the interpretation of the base types: $[\![\mathbf{N}]\!]$, $[\![\mathbf{var}]\!]_S$ and $[\![\mathbf{com}]\!]_S$. Interpretation of higher types is defined inductively:

$$\begin{aligned} [\![A \times B]\!] &= [\![A]\!] \times [\![B]\!] \\ [\![A \multimap B]\!] &= [\![A]\!] \multimap [\![B]\!] \\ [\![\mathsf{P}A]\!] &= \mathsf{P}[\![A]\!] \end{aligned}$$

A typed term

$$x_1 : A_1 \ldots x_{k-1} : A_{k-1} \mid x_k : A_k \ldots x_n : A_n \vdash_S M : A$$

will be modelled by a morphism

$$\mathsf{S}[\![A_1]\!] \otimes \ldots \otimes \mathsf{S}[\![A_{k-1}]\!] \otimes [\![A_k]\!] \ldots [\![A_n]\!] \to [\![A]\!]$$

and a closed term $\vdash_S M : A$ will be modelled by a morphism $1 \to [\![A]\!]$.

We should remind ourselves here that our semantics is constructed by induction on the typing derivation so we need a rule for constructing the semantics of a term corresponding to each typing rule.

The λ-calculus fragment of **SCIR** is interpreted using the symmetric monoidal closed structure. Identifiers are interpreted as identities:

$$[\![- \mid x : A \vdash_S x : A]\!] = \mathsf{id}_{[\![A]\!]}.$$

Abstraction is modelled by currying:

$$[\![\Gamma \mid \Delta \vdash_S M : A \multimap B]\!] = \Lambda([\![\Gamma \mid \Delta, x : A \vdash_S M : B]\!]) : [\![\Gamma \mid \Delta]\!] \to [\![A \multimap B]\!].$$

Application is modelled by composition with the evaluation map:

$$[\![\Gamma, \Gamma' \mid \Delta, \Delta' \vdash_S MN : B]\!] = [\![\Gamma \mid \Delta \vdash_S M : A \multimap B]\!] \otimes [\![\Gamma' \mid \Delta' \vdash_S N : A]\!]; \mathsf{eval}_{[\![A \multimap B]\!]}.$$

Pairing and projection are modelled using the categorical product:

$$[\![\Gamma \vdash_S \langle M, N \rangle : A \times B]\!] = \langle [\![\Gamma \vdash_S M : A]\!], [\![\Gamma \vdash_S N : B]\!] \rangle.$$

$$[\![\Gamma \vdash_S \pi_i N : A]\!] = [\![\Gamma \vdash_S N : A \times B]\!]; \pi_i.$$

We use the reflective and coreflective subcategorical structure of **C** to model the **SCIR** structural rules.

Passification and activation are modelled using the adjunction $S \dashv J$. For any map $f : A \to JB$ there exists a unique map $\mathsf{pass} f : SA \to JB$ such that $f = \eta^S_A ; \mathsf{pass} f$. Because the adjunction is monoidal it follows that given any map: $f : A \otimes B \to JC$ there exists a unique map $\mathsf{pass}_A(f) : SA \otimes B \to JC$ such that $\mathsf{id} \otimes \eta^S_A ; \mathsf{pass}_A(f) = f$. For proof see [38].

Activation is modelled by precomposition with the unit of the adjunction $\mathsf{S} \dashv J$:

$$[\![\Gamma \mid x : A, \Delta \vdash_S M : B]\!] = (\mathbf{id}_\Gamma \otimes \eta^S_A \otimes \mathbf{id}_\Delta) ; [\![\Gamma, x : A \mid \Delta \vdash_S M : B]\!].$$

Passification is modelled thus:

$$[\![\Gamma, x : A \mid \Delta \vdash_S M : B]\!] = \mathsf{pass}_A([\![\Gamma \mid x : A, \Delta \vdash_S M : B]\!])$$

Dereliction is modelled by composition with the counit of the adjunction $J \dashv \mathsf{P}$:

$$[\![\Gamma \mid \Delta \vdash_S \mathsf{derelict}\, M : A]\!] = [\![\Gamma \mid\vdash_S M : \mathsf{P}A]\!] ; \varepsilon^P_A.$$

Promotion is modelled by the adjunction $J \dashv \mathsf{P}$:

$$[\![\Gamma \mid\vdash_S \mathsf{promote}\, M : \mathsf{P}A]\!] = \mathsf{P}[\![\Gamma \mid\vdash_S M : A]\!].$$

Weakening is interpreted using the fact that the terminal object is the unit of the tensor.

$$[\![\Gamma, \Gamma' \mid \Delta, \Delta' \vdash_S M : A]\!] = \mathsf{id}_{\mathsf{S}[\![\Gamma]\!]} \otimes ! \otimes \mathsf{id}_{[\![\Delta]\!]} \otimes ! ; [\![\Gamma \mid \Delta \vdash_S M : A]\!].$$

Finally contraction is modelled by precomposition with the contraction map $\Delta : \mathsf{SA} \to \mathsf{SA} \otimes \mathsf{SA}$:

$$[\![\Gamma, z : A \mid \Delta \vdash_S M : B]\!] = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]} \otimes \Delta \otimes \mathbf{id}_{[\![\Delta]\!]}) ; [\![\Gamma, x : A, y : A \mid \Delta \vdash_S M : B]\!].$$

We know that the contraction map exists because $\mathsf{SA} \otimes \mathsf{SA} \simeq \mathsf{SA} \times \mathsf{SA}$.

Lastly, in a categorical model of **SCIR** we will require morphisms with which to model the Algol-like language constructs. We will only give the example of sequential composition of commands here, but see chapter 6 for a concrete example of how all the constructs may be interpreted. We must specify a map in **C**

$$\mathbf{seq}_\alpha : [\![\mathbf{com}]\!] \times [\![\mathbf{com}]\!] \to [\![\mathbf{com}]\!]$$

and we can then define the semantics of sequential composition of commands as follows:

$$[\![\Gamma \mid \Delta \vdash_B M ; N]\!] = \langle [\![\Gamma \mid \Delta \vdash_B M]\!], [\![\Gamma \mid \Delta \vdash_B N]\!] \rangle ; \mathbf{seq}_\alpha.$$

The interpretation we have given assigns a semantics inductively on the derivation. However, a judgement may in general have several possible derivations. It is therefore desirable to show that every derivation of a term yields the same semantics. This property is known as *coherence*. Given a categorical model of **SCIR** we are not guaranteed coherence. To this end McCusker [38] and O'Hearn *et al.* [40] propose extra structure to be required of a model. In [40] the notion of bireflectivity is proposed, which we will encounter shortly. McCusker's solution is more general than bireflectivity, and is termed a *retractive model of **SCIR***.

A categorical model of **SCIR** is retractive if and only if for every object A, there is a map $\alpha_A : JSA \to A$ such that $\alpha_A; \eta^S_A = \mathsf{id}_A$. McCusker shows that any retractive categorical model of **SCIR** is coherent [38]. It is interesting to note that the retractive property implies that passification can be defined using the retraction map.

$$\begin{aligned} f &= \eta^S_A; \mathsf{pass}\, f \\ \alpha_A; f &= \alpha_A; \eta^S_A; \mathsf{pass}\, f \\ \alpha_A; f &= \mathsf{pass}\, f \end{aligned}$$

Passification is therefore modelled as follows:

$$[\![\Gamma, x : A \mid \Delta \vdash_S M : B]\!] = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]} \otimes \alpha_{\mathsf{A}} \otimes \mathbf{id}_{[\![\Delta]\!]}); [\![\Gamma \mid x : A, \Delta \vdash_S M : B]\!].$$

O'Hearn *et al.* also propose adding structure to a categorical model of **SCIR** with their notion of bireflectivity [40]. A bireflective model is a categorical model of **SCIR** in which the left and right adjoints are equal. Coherence also holds for bireflective models of **SCIR**. A proof of coherence is sketched in [40] and it is this proof that motivates the extra structure of bireflectivity. However, it is trivial to show that a bireflective model is necessarily retractive and hence McCusker's proof suffices. It is interesting to note that in a bireflective model it is the case that the promotion rule can be equivalently modelled by post-composition with η^P. We only know of three models of **SCIR**. In [40] O'Hearn *et al.* give a concrete example of a bireflective, and hence retractive, semantics using functor categories. The system has also been studied by Reddy, who provided a model for it based on coherence spaces [46]. Reddy's model turns out to be retractive but not bireflective. The third model is the games model that we will present in chapter 6 which is also retractive but not bireflective.

5.5 Other Approaches to SCI

5.5.1 SCI 2

In [49] Reynolds proposes a language that conforms to all three of the SCI language principles. This language uses subtyping to counteract the subject reduction problems of Reynold's original proposal of an SCI language that incorporates passive types [47].

5.5.2 Bunched Typing — SCI+

In [43] O'Hearn proposes elegant typing systems based on the logic of bunched implications [41]. In bunched implication logic there are two implication connectives: a linear implication and an intuitionistic implication. In the type systems that O'Hearn proposes there are two kinds of function types; one where a a function and argument are permitted to interfere, and one where they are not. O'Hearn proposes a type system, **SCI+**, which subsumes both $\mathbf{IA}_a$ and $\mathbf{SCI}_b$. Furthermore O'Hearn goes on to show how bunched typing can be extended to accommodate passivity in the style of **SCIR**.

Chapter 6

A Games Model for SCI

In this chapter we will define the novel category of arenas, **C**, which we will later use to interpret languages that adhere to the SCI design principles: **PCF**, $\mathbf{SCI}_b$ and **SCIR**. We will then demonstrate that the category possesses the required structure described in chapter 5

- All finite products.
- A symmetric monoid with enough exponentials to model $\mathbf{SCI}_b$.
- A cartesian closed subcategory to model **PCF**.
- The necessary structure for a retractive model of **SCIR**.

Finally we will construct sound and adequate models of each of the languages in **C**.

6.1 SCI Arenas

The objects in our category will possess more structure than the QA arenas described in chapter 3

Definition 217 An SCI arena (in this chapter simply an arena) A is a tuple $\langle \mathsf{M_A}, \vdash_{\mathsf{A}}, \lambda_{\mathsf{A}}, \sim_{\mathsf{A}} \rangle$ where:

- $\mathsf{M_A}$ is a set of moves.
- $\lambda_{\mathsf{A}} : \mathsf{M_A} \to \{O,P\} \times \{Q,A\} \times \{\alpha,\pi\}$ is a labelling function that tells us whether a move is an O-move or a P-move, whether it is a question (Q) or an answer (A) and whether it is active (α) or passive (π). We sometimes write λ^{QA} if we are only concerned whether a given move is a question or an answer. We define λ^{OP}, $\lambda^{\alpha\pi}$, λ^{OPQA} etc. similarly.
- $\vdash_{\mathsf{A}} \subseteq (\mathsf{M_A} + \{\star\}) \times \mathsf{M_A}$, where $\star$ is just some dummy symbol, is known as the enabling relation and it must satisfy the following:
  1. $\star \vdash_{\mathsf{A}} \mathsf{m} \Rightarrow \lambda^{QA}_{\mathsf{A}}\mathsf{m} = Q \wedge (\mathsf{m}' \vdash \mathsf{m} \Leftrightarrow \mathsf{m}' = \star)$. We call such a move m an *initial move*.

2. If $m \vdash_A m'$ and $m \neq \star$ then
$$\lambda_A^{OP} m \neq \lambda_A^{OP} m'$$
and
$$\lambda_A^{QA} m = Q.$$

- $\sim_A \subseteq \{q \in M_A | \lambda^{QA} q = Q\} \times \{q \in M_A | \lambda^{QA} q = Q\}$ satisfying symmetry, reflexivity and
$$q \sim q' \Rightarrow \lambda^{OP} q = \lambda^{OP} q'.$$

We now draw the attention of the reader to several features of SCI arenas:

- Once again we are going to allow initial player moves.
- The distinction made by the labelling function between active and passive moves is not a novel feature. It has already been included by Abramsky and McCusker [10] in a fully abstract games model of Idealized Algol with passive expressions.
- We do not allow answers to justify questions. This may be considered a shortcoming of our model; we therefore cannot model the lifted sum types that are present in the language modelled by McCusker in [36]. It is not obvious to the author how best to extend the category to include lifted sums.
- The $\sim$ relation is also familiar and can be found in Laird's model of a language with linearly-used continuations [29] which was developed concurrently with the model presented in this chapter. The idea here is that a pair of moves in the $\sim$ relation may possibly interfere with one another. For this reason we have insisted that $\sim$ must be reflexive and, as we shall see later, when we form the product SCI arena $A + B$ we will insist that initial moves from A are in the $\sim$ relation with those from B.
- For any SCI arena
$$\langle M_A, \vdash_A, \lambda_A, \sim_A \rangle,$$
there is obviously an underlying QA arena
$$\langle M_A, \vdash_A, \lambda_A^{OPQA} \rangle.$$

Much of the terminology that we define for SCI arenas is inherited from that defined for QA arenas in chapter 3.

Definition 218 (Negative Arenas) We say that a negative SCI arena is one for which all initial moves are opponent moves.

Definition 219 Given a labelling function λ_A we define $\overline{\lambda_A}$ as the labelling function such that for all $m \in M_A$ we have

- $\overline{\lambda_A}^{QA\pi\alpha} m = \lambda_A^{QA\pi\alpha} m$.
- $\overline{\lambda_A}^{OP} m \neq \lambda_A^{OP} m$.

Definition 220 (Justified Sequences) A justified sequence for an SCI arena A is precisely a justified sequence for the underlying QA arena.

Definition 221 (Arrow Arena) Given negative SCI arenas A and B we define the SCI arrow arena $\mathsf{A} \to \mathsf{B} = \langle \mathsf{M}_{\mathsf{A}\to\mathsf{B}}, \vdash_{\mathsf{A}\to\mathsf{B}}, \lambda_{\mathsf{A}\to\mathsf{B}}, \sim_{\mathsf{A}\to\mathsf{B}} \rangle$ as follows:

$$\begin{aligned} \mathsf{M}_{\mathsf{A}\to\mathsf{B}} &= \mathsf{M}_{\mathsf{A}} + \mathsf{M}_{\mathsf{B}} \\ \vdash_{\mathsf{A}\to\mathsf{B}} &= \vdash_{\mathsf{A}} + \vdash_{\mathsf{B}} \\ \lambda_{\mathsf{A}\to\mathsf{B}} &= [\overline{\lambda_{\mathsf{A}}}, \lambda_{\mathsf{B}}] \\ \sim_{\mathsf{A}\to\mathsf{B}} &= \sim_{\mathsf{A}} + \sim_{\mathsf{B}} \end{aligned}$$

Once again, given a sequence $\mathsf{s} \in \mathcal{J}_{\mathsf{A}\to\mathsf{B}}$ we write $\mathsf{s} \upharpoonright \mathsf{A}$ for that subsequence of moves in s comprising exactly those moves from M_{A}; we define $\mathsf{s} \upharpoonright \mathsf{B}$ similarly.

Definition 222 We define the empty SCI arena:

$$\mathbf{null} = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

Lemma 223 Given a negative SCI arena A it follows that $\mathsf{A} = \mathbf{null} \to \mathsf{A}$. This follows directly from the definitions; the tagging of moves to form disjoint unions is not important here.

We define bracketing, visibility and switching conditions for sequences of moves for SCI arenas exactly as we do for the underlying arenas. Hence, given an arrow SCI arena A we define the sets $\mathcal{S}_{\mathsf{A}}$, $\mathcal{V}_{\mathsf{A}}$, $\mathcal{B}_{\mathsf{A}}$ as in chapter 3. Our $\sim$ relation alone is too general as an indicator of pairs of move occurrences that may interfere with each other, so we provide the following definition.

Definition 224 Given a sequence $\mathsf{s} \in \mathcal{J}_{\mathsf{A}}$ and moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that $\mathsf{m} \sim \mathsf{m}'$ then we say that $\mathsf{m} \leftrightsquigarrow \mathsf{m}'$ if either both moves are initial or they share a justifier.

6.2 New Constraints on Sequences

Strategies for SCI arenas are going to be strategies for the underlying QA arenas. In order to get definability we will, of course, have to place further constraints on the strategies. As we define the constraints we will attempt to outline some of the intuition behind each of them.

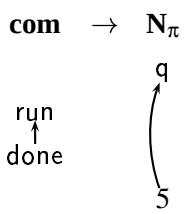
6.2.1 The Activity Condition

When we model **SCIR** in **C** we will need to be aware about the connection between passive moves and passive types.

Definition 225 We say that a sequence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ respects the activity condition if and only if for any initial active player question $\mathsf{q}_P \in \mathsf{s}$ it follows that $\mathbf{thread}(\mathsf{s}_{\leq \mathsf{q}_P})$ commences with an active move. We write $\mathcal{A}_{\mathsf{A}}$ for the set of sequences in $\mathcal{S}_{\mathsf{A}}$ which respect the activity condition.

A stronger restriction is present in Abramsky and McCusker's games model of Idealized Algol with passive expressions [10].

The activity condition, therefore, bans active initial player moves from occurring in passive threads. The following sequence which breaks this condition and is included in the semantics of the $\mathbf{IA}_a$ side-affecting expressions $x;5 : \mathbf{N}$ which is not typeable in **SCIR** or **PCF**.



6.2.2 The Nesting Condition

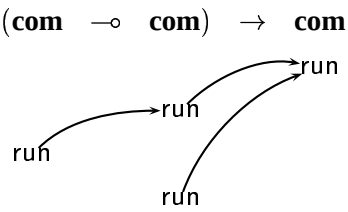
Definition 226 We say that player (respectively opponent) obeys the nesting condition in a sequence $\mathsf{s} \in \mathcal{S}_\mathsf{A}$ if and only if for any pair of player (respectively opponent) moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that $\mathsf{m} < \mathsf{m}' \wedge \mathsf{m} \leftrightsquigarrow \mathsf{m}'$ and $\lambda^{\pi\alpha}\mathsf{m} = \alpha$ or $\lambda^{\pi\alpha}\mathsf{m}' = \alpha$ it follows that

$$\exists \mathsf{a} \in \mathsf{s}_{<\mathsf{m}'}.\mathsf{m} \curvearrowleft \mathsf{a} \wedge \lambda^{QA}\mathsf{a} = A.$$

In other words m must be answered before m' is played. Note that a pair of passive move occurrences in the $\leftrightsquigarrow$ relation may nest without violating the nesting condition. We write $\mathcal{N}_\mathsf{A}$ for the subset of $\mathcal{S}_\mathsf{A}$ that obeys the nesting condition. We say that a sequence $\mathsf{s} \in \mathcal{L}_\mathsf{A}$ is honest if opponent obeys the nesting condition. We write $\mathcal{H}_\mathsf{A}$ for the subset of honest sequences in $\mathcal{S}_\mathsf{A}$.

This control of nesting has been of interest in the field of algorithmic game semantics and similar ideas are present in a model for a finitary language with parallel composition and binary semaphores in [22] and in Abramsky's Serially Reentrant Idealized Algol [3].

A typical sequence which violates the nesting condition is



In a term of $\mathbf{SCI}_b$ functions and their arguments must contain disjoint free identifiers and sequence like the above arises in the semantics of an $\mathbf{IA}_a$ term of the form $\lambda f.f(\ldots f\ldots)$. Such a term can only be built using *additive* application of an active function (or alternatively explicit contraction) and hence is not typeable in $\mathbf{SCI}_b$. Such a term, however, will be typeable in **PCF**, and possibly also in **SCIR** if the identifier f is used only passively.

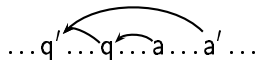
6.2.3 The SCI Condition

The nesting condition alone does not suffice to eliminate all sharing of resources between function and argument. To this end we introduced the key technical novelty: a partial order on moves which captures the independence of certain subsequences. The partial order is constructed for any sequence via a directed, acyclic graph inductively using the $\leftrightsquigarrow$ relation and the justification and labelling information.

Definition 227 Given a sequence $s \in \mathcal{S}_A$ with moves $a, a' \in s$ we say $a \smile a'$ if and only if

$$\lambda^{QA} a, a' = A \wedge \exists q, q' \in s.q' \curvearrowleft q \wedge q' \curvearrowleft a' \wedge q \curvearrowleft a$$

This relation can be described diagrammatically where $a \smile a'$ if and only if our sequence has the following form:

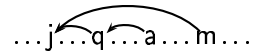


Note that well-bracketing ensures that we have $(a \smile a') \Rightarrow (a < a')$.

Definition 228 Given a sequence $s \in \mathcal{H}_A$ we define a relation between move occurrences such that $m \lhd m'$ if and only if we have one of the following.

1. $m \curvearrowleft m'$.
2. $(\exists q \in s_{\leq m'}.q \curvearrowleft m \wedge q \leftrightsquigarrow m') \wedge (\lambda^{\pi\alpha QA} m = \alpha A)$.
3. $(m \smile m') \wedge (\lambda^{\pi\alpha} m = \alpha)$.

The first item of the above definition is straightforward. The other two items are related and can be best seen diagrammatically. Consider the following sequence:



We intend $a \lhd m$ if and only if a is an active answer and either m answers j or $m \sim q$.

Definition 229 Given a sequence $s \in \mathcal{H}_A$ we define $\prec_s$ to be the least relation such that

- $(\lambda^{OP} m' = O \wedge m \lhd m') \Rightarrow (m \prec_s m')$
- $(\lambda^{OP} m = P) \Rightarrow (m^- \prec_s m)$

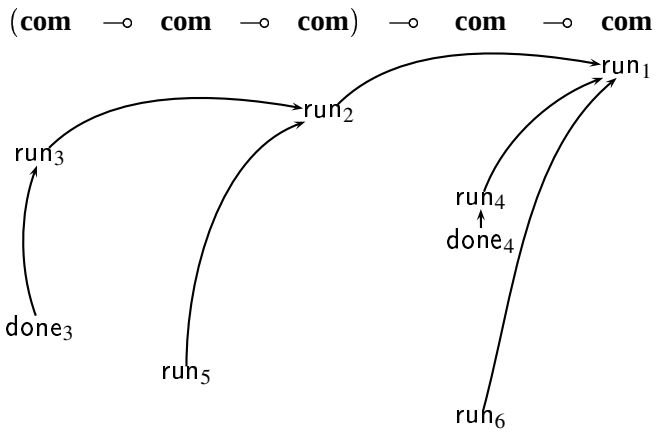
We use $\overset{*}{\prec}_s$ to represent the transitive, reflexive closure of $\prec_s$ and $\overset{*}{\prec\succ}_s$ to represent the symmetric closure of $\overset{*}{\prec}$. We omit subscripts if it makes things clearer.

Definition 230 We say that a justified sequence, $s \in \mathcal{H}_A$, satisfies the **SCI condition** if and only if for all moves $m, m' \in s$ such that $m \lhd m'$ then it is also the case that $m \overset{*}{\prec} m'$.

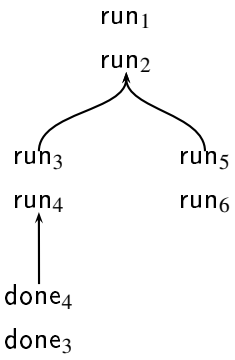
Definition 231 We refer to a sequence $s \in \mathcal{S}_A$ as an **SCI sequence** if it satisfies well-bracketing, total visibility, the nesting condition, the activity condition and the SCI condition. We write I_A for the subset of legal SCI sequences in $\mathcal{S}_A$.

To understand the intuition behind the SCI condition, consider the following sequence s, which is again permissible in the games model of $\mathbf{IA}_a$ and would appear in the semantics of a term that

uses an additive function application such as $\lambda fx.f(\ldots x\ldots)(\ldots x\ldots)$.



From this sequence one can build the partial order $\overset{*}{\prec}_{\mathsf{s}}$ that is represented by the following Hasse diagram (with least element at the top!). In this case the SCI condition is violated as $\mathsf{run}_4 \leftrightsquigarrow \mathsf{run}_6$ and both moves are active hence $\mathsf{done}_4 \lhd \mathsf{run}_6$, but in this graph the SCI condition is violated because we do not have $\mathsf{done}_4 \overset{*}{\prec} \mathsf{run}_6$. Intuitively, the two "arms" of the graph descending from the node run_2 represent non-interfering subcomputations, and as such they must not reference the same resource represented by the "interfering" moves run_4 and run_6.



6.2.4 SCI Strategies

A strategy for an SCI arrow arena A will be a set of even length sequences from I_{A} but we place further restrictions upon which sequences can and must coexist in a given strategy.

Definition 232 As with strategies for QA arenas, a strategy σ for area A is prefix-closed if and only if it is not empty and for any sequence $\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{m}' \in \sigma$ we have $\mathsf{s} \in \sigma$. All prefix-closed strategies contain the empty sequence ε.

Furthermore, a prefix-closed strategy σ is deterministic if and only if for all sequences $\mathsf{s}\cdot\mathsf{m}, \mathsf{s}'\cdot\mathsf{m}' \in \sigma$ we have $\mathsf{s} = \mathsf{s}' \Rightarrow \mathsf{s}\cdot\mathsf{m} = \mathsf{s}'\cdot\mathsf{m}'$.

Similarly we inherit the notion of copycat strategies from QA arenas. Given an SCI arena A we define the following strategy:

$$\mathbf{copy}^I_{\mathsf{A}} = \{\mathsf{s} \in I_{\mathsf{A}'\to\mathsf{A}''} \mid \forall \mathsf{s}' \sqsubseteq_{\mathsf{even}} \mathsf{s}.\mathsf{s}' \upharpoonright \mathsf{A}' = \mathsf{s}' \upharpoonright \mathsf{A}''\}$$

where the arenas A' and A'' are simply used to distinguish the different copies of A. These copycat strategies are going to be our identities so we will sometimes simply refer to $\mathbf{copy}^I_{\mathsf{A}}$ as $\mathbf{id}_{\mathsf{A}}$.

The definition of thread-independence is also inherited from QA arenas. We say that a deterministic strategy $\sigma \subseteq I_{\mathsf{A}}$ is thread-independent if and only if for all sequences $\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{n},\mathsf{s}' \in \sigma$ such that there exists a sequence $\mathsf{s}'\cdot\mathsf{m}' \in I_{\mathsf{A}}$ for which $\mathbf{thread}(\mathsf{s}\cdot\mathsf{m}) = \mathbf{thread}(\mathsf{s}'\cdot\mathsf{m}')$ then there must exist some extension $\mathsf{s}'\cdot\mathsf{m}'\cdot\mathsf{n}' \in \sigma$ such that $\mathbf{thread}(\mathsf{s}'\cdot\mathsf{m}'\cdot\mathsf{n}') = \mathbf{thread}(\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{n})$.

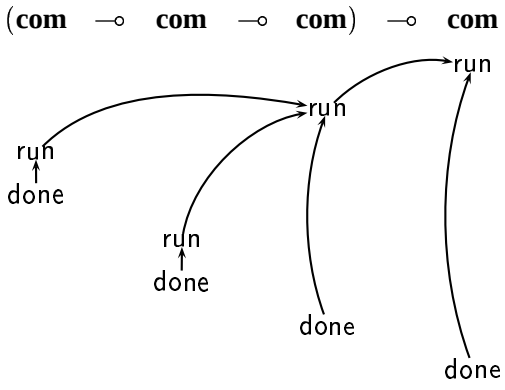
Definition 233 (SCI-view) The SCI-view of a sequence $\mathsf{s}\cdot\mathsf{m}$, notated $\lceil \mathsf{s}\cdot\mathsf{m}\rceil_{\prec}$, is the partial order $\langle M, \overset{*}{\prec}_{\mathsf{s}}\rangle$ where M is the set $\{\mathsf{m}' \in \mathsf{s}\cdot\mathsf{m} | \mathsf{m}' \overset{*}{\prec}_{\mathsf{s}} \mathsf{m}\}$. For completeness we say $\lceil\varepsilon\rceil_{\prec} = \langle\emptyset,\emptyset\rangle$. Given sequences s and s' we equate their SCI-views $\langle M, \overset{*}{\prec}_{\mathsf{s}}\rangle$ and $\langle M', \overset{*}{\prec}_{\mathsf{s}'}\rangle$ if and only if there exists a bijection $f : M \to M'$ between occurrences of the same move such that for any move occurrences $\mathsf{m},\mathsf{n} \in M$ and $\mathsf{m}',\mathsf{n}' \in M'$ where $f(\mathsf{m}) = \mathsf{m}'$ and $f(\mathsf{n}) = \mathsf{n}'$ we have:

- The bijection respects justification: $(\mathsf{m} \curvearrowleft \mathsf{n}) \Leftrightarrow (\mathsf{m}' \curvearrowleft \mathsf{n}')$.
- The bijection respects the $\prec$ relation: $(\mathsf{m} \prec_{\mathsf{s}} \mathsf{n}) \Leftrightarrow (\mathsf{m}' \prec_{\mathsf{s}'} \mathsf{n}')$.

Definition 234 (SCI-Innocence) We say that a deterministic strategy $\sigma \subset I_{\mathsf{A}}$ is SCI-innocent if and only if for all sequences $\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{n},\mathsf{s}' \in \sigma$ such that there exists a sequence $\mathsf{s}'\cdot\mathsf{m}' \in I_{\mathsf{A}}$ for which $\lceil \mathsf{s}\cdot\mathsf{m}\rceil_{\prec} = \lceil \mathsf{s}'\cdot\mathsf{m}'\rceil_{\prec}$ then there must exist some extension $\mathsf{s}'\cdot\mathsf{m}'\cdot\mathsf{n}' \in \sigma$ such that $\lceil \mathsf{s}'\cdot\mathsf{m}'\cdot\mathsf{n}'\rceil_{\prec} = \lceil \mathsf{s}\cdot\mathsf{m}\cdot\mathsf{n}\rceil_{\prec}$.

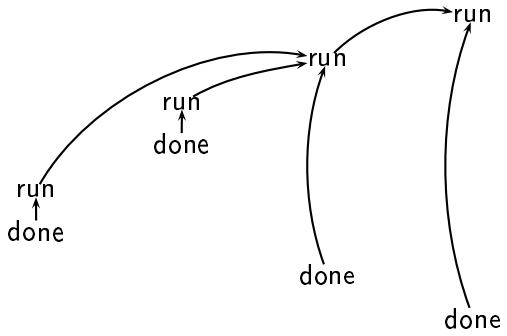
Definition 235 We say that a strategy $\sigma \subseteq I_{\mathsf{A}}$ is an SCI strategy if and only if it is both thread-independent and SCI-innocent.

To understand the intuition behind the definition of SCI-innocence we consider the terms $\lambda x^{\mathbf{com}} y^{\mathbf{com}}.x;y$ and $\lambda x^{\mathbf{com}} y^{\mathbf{com}}.y;x$. It is evident that these two terms are contextually equivalent in $\mathbf{SCI}_b$ so if we are to get a definability result for our semantics then any strategy containing the sequence

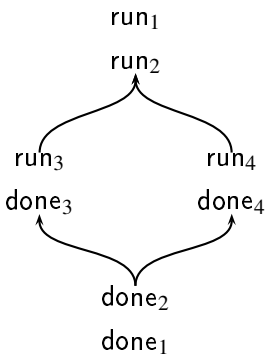


must also contain the following sequence.

$$(\mathbf{com} \multimap \mathbf{com} \multimap \mathbf{com}) \multimap \mathbf{com}$$

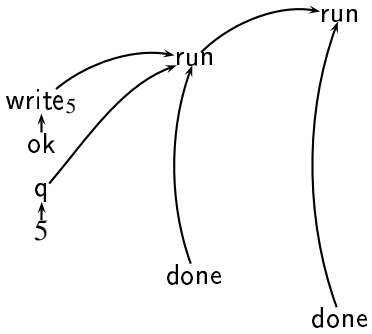


This is assured by SCI-innocence as the sequences form the same $\overset{*}{\prec}$ graph shown below. The idea is that for any odd length sequence, $s \cdot m$, player must play as a function of $\lceil s \cdot m \rceil_{\prec}$.



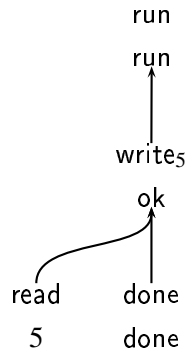
We also give an intuition for the SCI-view of the following sequence, corresponding to an **SCIR** type where the natural number type is passive.

$$(\mathbf{var}^{\pi} \multimap \mathbf{com}) \multimap \mathbf{com}$$



The idea here is that we have written a function that takes an argument $f : \mathbf{var} \multimap \mathbf{com}$. We can only observe partial behaviour about f: our function might be able to ascertain the sequence of assignments that f makes to its argument but it can never tell us whether or not f dereferences its argument. This intuition is captured when we examine the SCI-view of the above sequence in

which player has no record of a read move in the SCI-view of the complete sequence.



6.3 The Category C

Definition 236 We are now in a position to define the category **C**. Objects in **C** are negative SCI arenas, a morphism from A to B is an SCI strategy for the SCI arena $\mathsf{A} \to \mathsf{B}$, the identity for A is the strategy $\mathbf{id}_{\mathsf{A}}$ and composition is defined as composition of strategies for the underlying QA arenas.

We now must show that **C** is a well defined category. This is where the power of the methods we used in chapter 3 becomes apparent. We have already shown that $\mathcal{L}$ is an admissible referee in lemma 193. It is simple to show that $\mathcal{N}$ is separable and hence by lemma 114 we know that $\mathcal{N}$ is admissible. It is also simple to check that $\mathcal{A}$ is admissible. By lemma 112 we therefore know that the referee $(\mathcal{N} \cap \mathcal{L} \cap \mathcal{A})$ is admissible. We have to do a little more work to prove that I is admissible.

Definition 237 We define a sequence transformer $K : (\mathcal{N} \cap \mathcal{L} \cap \mathcal{A})$ as follows. Given an arrow arena A and a sequence $\mathsf{s} = \langle \mathsf{M}_\mathsf{s}, \prec_\mathsf{s}, \curvearrowleft_\mathsf{s} \rangle$ in $(\mathcal{N} \cap \mathcal{L} \cap \mathcal{A})_\mathsf{A}$ we define:

$$K_\mathsf{A}\mathsf{s} = \langle \mathsf{M}_\mathsf{s}, \prec_\mathsf{s}, \lhd_\mathsf{s} \rangle.$$

In other words K adds extra justification pointers between moves m and m' if $\mathsf{m} \lhd_\mathsf{s} \mathsf{m}'$.

Lemma 238 The category **C** is well defined.

Proof First we show that the referee I is admissible. It is simple to check that $\mathsf{s} \in (\mathcal{N} \cap \mathcal{L})_\mathsf{A}$ it follows that:

$$\mathsf{s} \in I_\mathsf{A} \Leftrightarrow K\mathsf{s} \text{ is player visible.}$$

We can now apply lemma 125 to show that I is an admissible referee.

We now know from theorem 106 that we can form a category $\mathbf{A}_I$ where the objects are negative SCI arenas and a strategy from A to B is a subset of $\mathsf{A} \to \mathsf{B}$. Furthermore, we know that the identity strategy for an SCI arena A is $\mathbf{copy}^I_\mathsf{A}$. By lemmas 152 and 153 we know that we can create a subcategory of $\mathbf{A}_I$ comprised of only the prefix-closed deterministic strategies. By lemma 191 we know that we can take a further subcategory containing only the thread-independent strategies. Lastly we must check that equality of SCI views is a compositional and copy complete umpire. We know that for any sequences $\mathsf{s}, \mathsf{s}' \in I_\mathsf{A}$ it follows that $K\mathsf{s}$ is player visible and it is straightforward to show the following:

- $\lceil s \rceil_{\prec} = \lceil s' \rceil_{\prec}$ if and only if $\lceil Ks \rceil = \lceil Ks' \rceil$.
- K is an injective sequence transformer.

We can now apply lemma 160 to show that equality of SCI views is a compositional and copy complete umpire and hence by theorem 158 it follows that **C** is a well defined category. ■

6.4 Weaving Threads

In this section we will show that an SCI strategy σ is determined exactly by the single threaded plays that it contains. This will later allow us to simplify some definitions and proofs about the categorical structure of **C**. We write $\sigma_{\mathsf{threads}}$ for the set of single threaded plays in σ. First we need to study some properties of sequences in SCI strategies.

6.4.1 Sequences in SCI Strategies

Lemma 239 Given an arrow arena A and a sequence $s \in \mathcal{L}_A$ with $q, a, j \in s$ where a answers q and $j \curvearrowleft m$ and $q \leq j \leq a$ then it follows that $m \leq a$.

Proof We construct a proof by induction on the length of the subsequence of s that lies strictly between j and m.

Base Case If j and m are adjacent we know that a cannot justify any move so $j < a$ and hence $m \leq a$.

Inductive Step We only consider the case when m is a player move, as the other case is similar. The visibility condition ensures us that $j \in \lceil s_{\leq m} \rceil$ and therefore this view will take the following form.

$$\cdots j \cdot p_1 \overset{\frown}{\cdot} o_1 \cdots p_n \overset{\frown}{\cdot} o_n \cdot m$$

We apply the inductive hypothesis to show that a cannot lie strictly between a pair of moves p_i and o_i, such that $p_i \curvearrowleft o_i$. We also see that a cannot be in this view unless $a = m$ as each of the p_i is a justifier and by the definition of $\vdash$ we know that an answer cannot do this, and each of the o_i is justified by a move that lies strictly after j. Therefore we conclude that $m \leq a$. ■

Lemma 240 Given an arrow arena A and a sequence $s \in \mathcal{L}_A$ with moves $q, a \in s$ such that a answers q we have $q \in \lceil s \rceil \Rightarrow a \in \lceil s \rceil$ and $q \in \lfloor s \rfloor \Rightarrow a \in \lfloor s \rfloor$.

Proof We prove only the first statement in the case when q is a player move. The other proofs are similar.

We know that q is answered in s so it is not the final move in the sequence s and therefore not the final move in $\lceil s \rceil$. Note that $q \in \lceil s \rceil$ implies that the move immediately succeeding q in the view is justified by q. Let us call this move n. We can see that $\lceil s \rceil$ takes the following form

$$\cdots q \overset{\frown}{\cdot} n \cdot p_1 \overset{\frown}{\cdot} o_1 \cdots p_n \overset{\frown}{\cdot} o_n \cdots$$

If a is not in this view then it must be that there exists some pair of moves in the view, $p_i \curvearrowleft o_i$ and $p_i < a < o_i$ but this conflicts lemma 239. ■

Lemma 241 Given an arrow arena A and a sequence $\mathsf{s} \cdot \mathsf{m} \in \mathcal{L}_{\mathsf{A}}$ with move $\mathsf{m}' \in \mathsf{s} \cdot \mathsf{m}$ then

$$\lambda \mathsf{m}' = PA \wedge \mathsf{m}' \in \lceil \mathsf{s} \cdot \mathsf{m} \rceil \Rightarrow \mathsf{m} = \mathsf{m}'$$

and similarly

$$\lambda \mathsf{m}' = OA \wedge \mathsf{m}' \in \lfloor \mathsf{s} \cdot \mathsf{m} \rfloor \Rightarrow \mathsf{m} = \mathsf{m}'.$$

Proof We only prove the first statement as proof of the second is similar. We carry out a proof by induction on the length of $\mathsf{s} \cdot \mathsf{m}$.

Base Case If $\mathsf{s} = \varepsilon$ the proof is trivial.

Inductive Step

case: If m is a player move then we apply our inductive hypothesis to show that there are no player answers in s.

case: If m is an initial opponent move then the proof is trivial.

case: If m is an opponent move justified by j then by definition of player view we have $\lceil \mathsf{s} \cdot \mathsf{m} \rceil = \lceil \mathsf{s}_{\leq \mathsf{j}} \rceil \cdot \mathsf{m}$. We apply our inductive hypothesis to $\mathsf{s}_{\leq \mathsf{j}}$ and note that j cannot be an answer as it justifies m. ■

Lemma 242 Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ with adjacent moves $\mathsf{q}, \mathsf{a} \in \mathsf{s}$ such that a answers q we have the following

- $\forall \mathsf{m} \in \mathsf{s}.\mathsf{q} \prec \mathsf{m} \Leftrightarrow \mathsf{a} = \mathsf{m}$.
- $\forall \mathsf{m} \in \mathsf{s}.\mathsf{m} \prec \mathsf{a} \Leftrightarrow \mathsf{m} = \mathsf{q}$.

Proof If a is a player move then a simple inspection of the definition of $\prec$ suffices. Otherwise, if a is an opponent we simply apply lemma 239 to show that q justifies no other moves and then inspect the definition of $\prec$. ■

Lemma 243 Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ with adjacent moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ with m' a player move we have

$$\forall \mathsf{m}'' \in \mathsf{s}.\mathsf{m}'' \overset{*}{\prec\succ} \mathsf{m} \Leftrightarrow \mathsf{m}'' \overset{*}{\prec\succ} \mathsf{m}'.$$

Proof First we assume $\mathsf{m}'' \overset{*}{\prec\succ} \mathsf{m}$ to prove $\mathsf{m}'' \overset{*}{\prec\succ} \mathsf{m}'$.

If $\mathsf{m}'' \overset{*}{\prec} \mathsf{m}$ then we simply appeal to the transitivity of $\overset{*}{\prec}$. Otherwise we note that $\mathsf{m} \overset{*}{\prec} \mathsf{m}''$ implies that either we have $\mathsf{m} = \mathsf{m}''$ in which case $\mathsf{m}'' \overset{*}{\prec} \mathsf{m}'$, or we have

$$\exists \mathsf{n} \in \mathsf{s}.\mathsf{m} \prec \mathsf{n} \overset{*}{\prec} \mathsf{m}''$$

and inspection of the definition of $\prec$ reveals that $\mathsf{n} = \mathsf{m}'$ and hence $\mathsf{m}' \overset{*}{\prec} \mathsf{m}''$.

The proof that $\mathsf{m}'' \overset{*}{\prec\succ} \mathsf{m}' \Rightarrow \mathsf{m}'' \overset{*}{\prec\succ} \mathsf{m}$ is similar. ■

Lemma 244 Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ we have

$$\forall \mathsf{m}, \mathsf{m}' \in \mathsf{s}.\mathsf{m} \curvearrowleft \mathsf{m}' \Rightarrow \mathsf{m} \overset{*}{\prec} \mathsf{m}'.$$

Proof If m' is an opponent move then we have $\mathsf{m} \lhd \mathsf{m}'$ by definition hence $\mathsf{m} \prec \mathsf{m}'$ and $\mathsf{m} \overset{*}{\prec} \mathsf{m}'$.

If m' is a player move then we have a little more work to do. Note that we must have $m \in \lceil s_{\leq m'} \rceil$ by virtue of the visibility condition. We know that $\lceil s_{\leq m'} \rceil$ has the form

$$\cdots m \cdot p_1 \overset{\frown}{\cdot} o_1 \cdots p_n \overset{\frown}{\cdot} o_n \cdot m'.$$

Note that in this sequence we have each move related to the next by $\prec$ and hence, by transitivity, we have $m \overset{*}{\prec} m'$. ■

Lemma 245 Given a sequence $s \in \mathcal{H}_A$ and moves $j, m, m' \in s$,

$$j \curvearrowleft m \wedge m \overset{*}{\prec\!\succ} m' \Rightarrow j \overset{*}{\prec\!\succ} m'.$$

Proof We know that $j \curvearrowleft m \Rightarrow j \overset{*}{\prec} m$ by lemma 244. Therefore, if $m \overset{*}{\prec} m'$ then $j \overset{*}{\prec} m'$ by transitivity.

Otherwise we have $m' \overset{*}{\prec} m$ and we carry out a proof by induction on the length of that segment s lying strictly between j and m.

Base Case If s is of the form $s' \cdot j \cdot m \cdot s''$ and $m' \overset{*}{\prec} m$ then either $m' = m$ and $j \overset{*}{\prec} m'$ by lemma 244 or $\exists n \in s. m' \overset{*}{\prec} n \prec m$. So we examine the definition of $\prec$ to find a possible n. In this case we must have $n = j$ by inspection of the definition of $\prec$. Hence $m' \overset{*}{\prec} j$.

Inductive Step

case: Suppose m is a player move. If we examine the sequence $\lceil s_{\leq m} \rceil$ then we know that j is in this view by virtue of the visibility condition. So we know that this view is of the following form

$$s' \cdot j \cdot p_1 \overset{\frown}{\cdot} o_1 \cdots p_n \overset{\frown}{\cdot} o_n \cdot m$$

Each consecutive pair of moves, $n\ n^+$, in that part of this view that lies inclusively between j and m has the following property

$$m' \overset{*}{\prec\!\succ} n^+ \Rightarrow m' \overset{*}{\prec\!\succ} n$$

This is evident by applying lemma 243 if n^+ is a player move, or by applying the inductive hypothesis if it is an opponent move. We now simply observe that the transitivity of this implication permits this property to be considered consecutively, starting at m and working backwards until we have $m' \overset{*}{\prec\!\succ} m \Rightarrow m' \overset{*}{\prec\!\succ} j$.

case: If m is an opponent move then we again look at the possible cases where $m' \overset{*}{\prec} m$. As before, either $m' = m$ and $j \overset{*}{\prec} m'$ by lemma 244 or else there exists some move $n \in s$ such that $m' \overset{*}{\prec} n \prec m$. Now, examining the definition of $\prec$, either n justifies m in which case $n = j$ and $m' \overset{*}{\prec} j$ or else we have some move q such that $j \curvearrowleft q \curvearrowleft n$ with n answering q. Now we apply the inductive hypothesis twice to yield

$$m' \overset{*}{\prec} n \Rightarrow m' \overset{*}{\prec\!\succ} q \Rightarrow m' \overset{*}{\prec\!\succ} j.$$

■

Lemma 246 Given a sequence $s \in \mathcal{H}_A$,

$$\forall m, m', m'' \in s.m \overset{*}{\prec\!\succ} m' \wedge m'' \in \lceil s_{\leq m} \rceil \Rightarrow m' \overset{*}{\prec\!\succ} m''.$$

Proof Note that each pair of consecutive moves $n, n^+ \in \lceil s_{\leq m} \rceil$ it follows that

$$m' \overset{*}{\prec\succ} n^+ \Rightarrow m' \overset{*}{\prec\succ} n.$$

We see this by applying lemma 245 if n^+ is an opponent move and by applying lemma 243 if it is a player move. We may, therefore, perform an induction on the length of this view and simply appeal to the transitivity of this implication to prove the lemma. ■

Lemma 247 Given a sequence $s \in \mathcal{H}_A$ with $q, m \in s$ where q is a question, unanswered in $s_{\leq m}$ we have

$$q \overset{*}{\prec} m \Rightarrow q \in \lceil s_{\leq m} \rceil.$$

Proof We carry out a proof by induction on the length of $s_{<m}$.

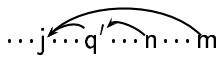
Base Case If $s_{<m} = \varepsilon$ then $q = m$ and the lemma is trivial.

Inductive Step First, suppose m is a player move. We note that if $q \overset{*}{\prec} m$ it must be that that either $q = m$, in which case the lemma is trivial, or else there exists some move $n \in s$ such that $q \overset{*}{\prec} n \prec m$. From the definition of $\prec$ we see that n is the immediate predecessor of m. We have $q \in \lceil s_{\leq n} \rceil$ by the induction hypothesis and from the definition of player view we see that $q \in \lceil s_{\leq m} \rceil$.

Now suppose m is an opponent move. As $q \overset{*}{\prec} m$ then either $q = m$ and the lemma is trivial or there must be some move $n \in s$ such that $q \overset{*}{\prec} n \prec m$. Now we inspect the definition of $\prec$ to find out what n could be. Either $n \curvearrowleft m$, in which case $q \in \lceil s_{\leq n} \rceil$ by the inductive hypothesis and therefore $q \in \lceil s_{\leq m} \rceil$ by inspection of the definition of player view. Otherwise n is the answer to some question q' such that $m \leftrightsquigarrow q'$, hence we have one of the following two cases.

case: m and q' are both initial questions. Because q is unanswered in $s_{\leq m}$, it must be true that $q < q'$ by the bracketing condition. We know by lemma 245 that $q \overset{*}{\prec} q'$ so by inductive hypothesis we have $q \in \lceil s_{\leq q'} \rceil$ and as q' is an initial opponent move it must be that $q = q'$ which is incompatible with $q < q'$.

case: m and q' share a justifier j



Once again the bracketing condition assures us that $q < q'$ and again we know by lemma 245 that $q \overset{*}{\prec} q'$ and therefore by inductive hypothesis we have $q \in \lceil s_{\leq q'} \rceil$. By inspection of the definition of player view we must have $q \leq j$. Lemma 245 now gives us $q \overset{*}{\prec} j$ and by inductive hypothesis $q \in s_{\leq j}$ hence, from the definition of player view, $q \in \lceil s_{\leq m} \rceil$.

■

Lemma 248 Suppose we are given a sequence $s \in \mathcal{H}_A$ and moves $q, a, m \in s$ such that:

- $q \leq m \leq a$.
- Player move a answers q.
- $q \in \lceil s_{\leq m} \rceil$.
- The move m is an active player question then m is either initial or has a justifier that lies in $s_{\leq q}$.
- All opponent questions strictly between q and m in $\lceil s_{\leq m} \rceil$ are active.

Then it follows that $\mathsf{m} \overset{*}{\prec} \mathsf{a}$.

Proof We construct a proof by induction on the length of s.

Base Case If $\mathsf{s} = \varepsilon$ then the lemma is trivial.

Inductive Step

case: Suppose q is an opponent move. If $\mathsf{m} \in \lceil \mathsf{s}_{\leq \mathsf{a}} \rceil$ then we have $\mathsf{m} \overset{*}{\prec} \mathsf{a}$ by lemma 246. Otherwise we apply lemma 139 to show that $\lceil \mathsf{s}_{\leq \mathsf{a}} \rceil$ is of the form

$$\cdots \mathsf{q} \cdot \mathsf{q}_P^1 \cdot \mathsf{a}_O^1 \ldots \mathsf{q}_P^n \cdot \mathsf{a}_O^n \cdot \mathsf{a}$$

and let m be occluded from this view by some pair of moves $\mathsf{q}_P^i \cdot \mathsf{a}_O^i$. Note that every consecutive pair of moves, n and n^+ in $\lceil \mathsf{s}_{\leq \mathsf{a}} \rceil$ that lie inclusively between q and a have the property that $\mathsf{n} \in \lceil \mathsf{s}_{\leq \mathsf{m}} \rceil \Rightarrow (\mathsf{m} < \mathsf{n}^+ \vee \mathsf{n} \in \lceil \mathsf{s}_{\leq \mathsf{m}} \rceil)$ so a simple induction on the value of i yields $\mathsf{q}_P^i \in \lceil \mathsf{s}_{\leq \mathsf{m}} \rceil$ and we can apply the inductive hypothesis to $\mathsf{s}_{\leq \mathsf{a}_O^i}$ to yield $\mathsf{m} \overset{*}{\prec} \mathsf{a}_O^i$ and hence $\mathsf{m} \overset{*}{\prec} \mathsf{a}$ by lemma 246 and transitivity.

case: Now suppose that q is a player move. If $\mathsf{m} = \mathsf{q}$ then we simply apply lemma 244 to yield $\mathsf{m} \overset{*}{\prec} \mathsf{a}$ and if $\mathsf{m} = \mathsf{a}$ we simply apply reflexivity. Otherwise, let q_O be the move immediately following q in $\lceil \mathsf{s}_{\leq \mathsf{m}} \rceil$. Well bracketing assure us that q_O is answered by some move a_P before a is played and lemmas 241 and 240 assure us that $\mathsf{m} \leq \mathsf{a}_P$.

$$\cdots \mathsf{q} \ldots \mathsf{q}_O \ldots \mathsf{m} \ldots \mathsf{a}_P \ldots \mathsf{a}$$

We can now apply the inductive hypothesis to $\mathsf{s}_{\leq \mathsf{a}_P}$ to yield $\mathsf{m} \overset{*}{\prec} \mathsf{a}_P$. We have a_P active by initial assumption, hence $\mathsf{a}_P \prec \mathsf{a}$ by definition and $\mathsf{m} \overset{*}{\prec} \mathsf{a}$ by transitivity. ■

Lemma 249 Given a sequence $\mathsf{s} \in \mathcal{H}_\mathsf{A}$ with moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that m is a passive player answer then $\mathsf{m} \overset{*}{\prec} \mathsf{m}'$ implies $\mathsf{m} = \mathsf{m}'$.

Proof If $\mathsf{m} \neq \mathsf{m}'$ there must exist some move $\mathsf{k} \in \mathsf{s}$ such that $\mathsf{m} \prec \mathsf{k} \overset{*}{\prec} \mathsf{m}'$ but inspection of the definition of $\prec$ reveals that there can be no such k. ■

Lemma 250 Given a sequence $\mathsf{s} \in \mathcal{L}_\mathsf{A}$ then for all $\mathsf{q}, \mathsf{a}, \mathsf{m} \in \mathsf{s}$ such that a answers q and $\mathsf{q} < \mathsf{a} \leq \mathsf{m}$

$$\mathsf{q} \overset{*}{\prec} \mathsf{m} \Rightarrow \mathsf{a} \overset{*}{\prec} \mathsf{m}.$$

Proof We carry out a proof by induction on the number of moves strictly between q and a.

Base Case As a base case we have s of the form $\mathsf{s}' \cdot \mathsf{q} \cdot \mathsf{a} \cdot \mathsf{s}''$. We know that $\mathsf{q} \overset{*}{\prec} \mathsf{m}$, and we know that $\mathsf{q} \neq \mathsf{m}$ as $\mathsf{q} < \mathsf{a} \leq \mathsf{m}$, so by the definition of $\overset{*}{\prec}$ we know that there exists some move $\mathsf{k} \in \mathsf{s}$ such that $\mathsf{q} \prec \mathsf{k} \overset{*}{\prec} \mathsf{m}$. However, we know from lemma 242 that in this case k must be a hence $\mathsf{a} \overset{*}{\prec} \mathsf{m}$.

Inductive Step

case: Consider the case when q is an opponent move. By visibility and well bracketing we know that $\lceil \mathsf{s}_{\leq \mathsf{a}} \rceil$ is of the form

$$\cdots \mathsf{q} \cdot \mathsf{q}_1 \cdot \mathsf{a}_1 \cdots \mathsf{q}_n \cdot \mathsf{a}_n \cdot \mathsf{a}.$$

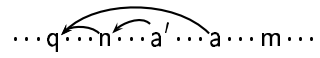
Note that for any pair of adjacent moves, n and n' in this sequence that lie inclusively between q and a have the following property

$$n \overset{*}{\prec} m \Rightarrow n' \overset{*}{\prec} m'.$$

This results from an application of the inductive hypothesis if n is a player move. Otherwise we apply lemma 243. It is plain to see that repeated application of the above result commencing at q yields

$$q \overset{*}{\prec} m \Rightarrow a \overset{*}{\prec} m.$$

case: We now consider the case when q is a player move. If $q \overset{*}{\prec} m$ then, as $q \neq m$, there exists some move $n \in s$ such that $q \prec n \overset{*}{\prec} m$. By inspection of the definition of $\prec$ we observe that n must be justified by q as q is a player question. If n is an answer then it must be a by the bracketing condition and we have $a \overset{*}{\prec} m$. Otherwise n must be a question. Lemma 240 ensures that $n < a$ and the bracketing condition ensures that it is answered, by a move a', before a is played. We have the following situation.



As $n \overset{*}{\prec} m$ we can apply the inductive hypothesis to yield $a' \overset{*}{\prec} m$ and lemma 249 tells us that a' must therefore be active, hence $a' \prec a$ by definition.

Now let us define the following set of move occurrences.

$$A = \{l \in s | l \prec a \wedge l \overset{*}{\prec} m\}$$

We have shown that a' is in this set. We now take the move in this set which is greatest with respect to the $\leq$ relation and let us call this move a_{max}. As $a_{max} \overset{*}{\prec} m$ there must be some move $k' \in s$ such that $a'' \prec k' \overset{*}{\prec} m$. We now examine this move.

Suppose that k' is a question. By inspection of the definition of $\overset{*}{\prec}$ we see that $q \curvearrowleft k'$. Of course, lemma 240 ensures that $k' < a$ and the bracketing condition ensures that such a question would be answered before a. Let us call this answer l. We then apply the induction hypothesis to get $l \overset{*}{\prec} m$ and the following situation.

By lemma 249 l is active hence $l \prec a$ by definition and $l \in A$. As $a_{max} < l$ this conflicts with our definition of a_{max}.

Therefore k' must be an answer and inspection of the definition of $\prec$ yields $k' = a$ and hence $a \overset{*}{\prec} m$. ■

Lemma 251 Given sequence $s \in \mathcal{L}_A$ we have $s \cdot m \in \mathcal{L}_A$ if and only if $\lceil s \cdot m \rceil \in \mathcal{L}_A$ when m is a player move and $\lfloor s \cdot m \rfloor \in \mathcal{L}_A$ when m is an opponent move.

Lemma 252 Given a sequence $s \in \mathcal{J}_A$ we have

$$\begin{aligned} s \in \mathcal{L}_A &\Rightarrow \lceil s \rceil \in \mathcal{L}_A \\ s \in \mathcal{H}_A &\Rightarrow \lceil s \rceil \in \mathcal{H}_A \\ s \in I_A &\Rightarrow \lceil s \rceil \in I_A \end{aligned}$$

Proof Suppose $s \in \mathcal{L}_A$. Inspection of the definition of player view ensures that $\lceil s \rceil$ alternates and

commences with an opponent move. Every non-initial move $m \in \lceil s \rceil$ has a justifier, by construction if m is an opponent move and by visibility if it is a player move. s is well bracketed by lemma 138 hence $\lceil s \rceil \in \mathcal{L}_A$.

Now suppose $s \in \mathcal{H}_A$. We know that $\lceil s \rceil \in \mathcal{L}_A$ so we only need to check that opponent respects the nesting condition in $\lceil s \rceil$. Given opponent questions $m, m' \in \lceil s \rceil$ such that $m \leftrightsquigarrow m'$ with $m < m$, and either m or m' active, as $s \in \mathcal{H}_A$ we know that m must be answered, by some move $a \in s$ before m' is played. Lemma 240 assures us that $a \in \lceil s \rceil$ and hence $\lceil s \rceil \in \mathcal{H}_A$.

Finally suppose $s \in I_A$. As $\lceil s \rceil \in \mathcal{H}_A$ we need to check that player respects the nesting condition and the SCI-condition in order to show that $\lceil s \rceil \in I_A$. Given player moves $q, q' \in \lceil s \rceil$ such that $m \leftrightsquigarrow m'$ with $m < m$, and either m or m' active, as $s \in I_A$ we know that m must be answered, by some move $a \in s$ before m' is played and again lemma 240 assures us that this move will also be in $\lceil s \rceil$ hence the player respects the nesting condition. Lemma 246 ensures that $m \overset{*}{\prec} m'$ and hence the SCI-condition is also respected. ■

6.4.2 Threads in SCI Strategies

In this section we will show that a strategy is uniquely determined by the set of single threaded sequences that it contains. We find single threaded sequences easier to reason with and we will later define connectives in the category via the sets of single threaded sequences they contain. We shall now define the *thread rules* — a set of constraints that we place on a set of single threaded sequences that exactly characterize the subset of threads that may exist in an SCI strategy.

Definition 253 (Thread Rules) Given a set of even length single threaded sequences $T \subseteq \mathcal{S}_A$ we term the following conditions the thread rules:

T1 $T \subseteq I_A$.

T2 $s \cdot m \cdot m' \in T \Rightarrow s \in T$.

T3 For all sequences $s \cdot m \cdot n, s' \cdot m' \cdot n' \in T$ such that $s \cdot m = s' \cdot m'$ we have $s \cdot m \cdot n = s' \cdot m' \cdot n'$.

T4 For all sequences $s \cdot m \cdot n, s' \in T$ where there exists an extension $s' \cdot m' \in I_A$ such that $\lceil s \cdot m \rceil_{\prec} = \lceil s' \cdot m' \rceil_{\prec}$ it follows that there is some extension $s' \cdot m' \cdot n' \in T$ such that $\lceil s \cdot m \cdot n \rceil_{\prec} = \lceil s' \cdot m' \cdot n' \rceil_{\prec}$.

T5 For all sequences $q_O \cdot s \cdot q_P \cdot t \cdot m, q'_O \cdot s' \cdot q'_P \in T$ such that the following hold:

- $q_O \overset{*}{\curvearrowleft} m$.
- q_P and q'_P are initial player questions.
- $q_P \sim q'_P$.
- q_P is active.

then it follows that $q_O \sim q'_O$.

T6 For all sequences $q_O \cdot s \cdot q_P \cdot t \cdot m, q'_O \cdot s' \cdot q'_P \in T$ such that the following hold:

- $q_O \overset{*}{\curvearrowleft} m$.

- q_P and q'_P are initial player questions.
- $q_P \sim q'_P$.
- q'_P is active.
- q_P is unanswered in $q_O \cdot s \cdot q_P \cdot t \cdot m$.

then it follows that $q_O \sim q'_O$.

Rules **T5-6** may seem a little cryptic. Intuitively the idea is that if two threads both use interfering resources then the threads themselves interfere. However, the thread rules are a little weaker and a lot more complicated than we might expect. There is no rule in the construction of the category **C** that banishes the following pair of sequences from coexisting in the same strategy:

com	$\to$	**com**	$\otimes$	**com**
				run
run				

com	$\to$	**com**	$\otimes$	**com**
		run		
run				

In fact a valid SCI strategy can contain exactly this pair of sequences and the empty sequence. However, if we complete one of the sequences as follows

com	$\to$	**com**	$\otimes$	**com**
				run
run				
done				
				done

it becomes clear that the extended sequence cannot coexist with the others in a valid SCI strategy as thread independence would require the inclusion of the following play which breaks the SCI condition.

com	$\to$	**com**	$\otimes$	**com**
				run
run				
done				
				done
		run		
run				

The following pair of lemmas show more concretely why rules **T5-6** are important.

Lemma 254 Given an SCI strategy $\sigma : A$ with single threaded sequences $q_O \cdot s \cdot q_P \cdot t \cdot m, q'_O \cdot s' \cdot q'_P \in \sigma$ such that the following hold:

- $q_O \overset{*}{\curvearrowleft} m$.
- q_P and q'_P are initial player questions.

- $\mathsf{q}_P \sim \mathsf{q}'_P$.
- q_P is active.

then it follows that $\mathsf{q}_O \sim \mathsf{q}'_O$.

Proof It is straightforward to show that if $\mathsf{q}_O \not\sim \mathsf{q}'_O$ then it follows that we can concatenate the sequences to yield

$$\mathsf{q}_O \cdot \mathsf{s} \cdot \mathsf{q}_P \mathsf{t} \cdot \mathsf{m} \cdot \mathsf{q}'_O \cdot \mathsf{s}' \cdot \mathsf{q}'_P \in \mathcal{H}_{\mathsf{A}}.$$

By thread independence of σ it is now simple to show that

$$\mathsf{q}_O \cdot \mathsf{s} \cdot \mathsf{q}_P \mathsf{t} \cdot \mathsf{m} \cdot \mathsf{q}'_O \cdot \mathsf{s}' \cdot \mathsf{q}'_P \in \sigma.$$

If q_P is not answered in t then this sequence breaks the nesting condition. Otherwise, if q_P is answered by some move a_O, it is simple to check that in the above sequence $\mathsf{a}_O \overset{*}{\not\curvearrowleft} \mathsf{q}'_P$ and hence the SCI-condition is broken. ■

Lemma 255 Given an SCI strategy $\sigma : \mathsf{A}$ with single threaded sequences $\mathsf{q}_O \cdot \mathsf{s} \cdot \mathsf{q}_P \cdot \mathsf{t} \cdot \mathsf{m}, \mathsf{q}'_O \cdot \mathsf{s}' \cdot \mathsf{q}'_P \in T$ such that the following hold:

- $\mathsf{q}_O \overset{*}{\curvearrowleft} \mathsf{m}$.
- q_P and q'_P are initial player questions.
- $\mathsf{q}_P \sim \mathsf{q}'_P$.
- q'_P is active.
- q_P is unanswered in $\mathsf{q}_O \cdot \mathsf{s} \cdot \mathsf{q}_P \cdot \mathsf{t} \cdot \mathsf{m}$.

then it follows that $\mathsf{q}_O \sim \mathsf{q}'_O$.

Proof It is straightforward to show that if $\mathsf{q}_O \not\sim \mathsf{q}'_O$ then it follows that we can concatenate the sequences to yield

$$\mathsf{q}_O \cdot \mathsf{s} \cdot \mathsf{q}_P \mathsf{t} \cdot \mathsf{m} \cdot \mathsf{q}'_O \cdot \mathsf{s}' \cdot \mathsf{q}'_P \in \mathcal{H}_{\mathsf{A}}.$$

and as q_P is not answered it follows that this sequence breaks the nesting condition. ■

Lemma 256 Given an SCI-strategy $\sigma : \mathsf{A}$ it follows that $\sigma_{\mathsf{threads}}$ obeys the thread rules **T1-6**.

Proof

T1 follows from the definition of an SCI-strategy.

T2 also follows from the definition of an SCI-strategy.

T3 follows from the determinism of σ.

T4 follows from the SCI-innocence of σ.

T5 follows from lemma 254.

T6 follows from lemma 255.

■

Lemma 257 Given sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{V}_{\mathsf{A}}$ with $\mathbf{thread}(\mathsf{s}) = \mathbf{thread}(\mathsf{s}')$ we have $\lceil \mathsf{s} \rceil = \lceil \mathsf{s}' \rceil$.

Proof The proof follows directly from the definitions of $\mathbf{thread}(-)$ and $\lceil - \rceil$. ■

Lemma 258 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ with moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that $\mathsf{m} \curvearrowleft \mathsf{m}'$ we have $\mathsf{m} \in \mathbf{thread}(\mathsf{s}) \Rightarrow \mathsf{m}' \in \mathbf{thread}(\mathsf{s})$.

Proof Note that if m is a player move then $\mathsf{m} \in \lceil \mathsf{s}_{\leq \mathsf{m}'} \rceil$ by construction and if m is an opponent move then $\mathsf{m} \in \lceil \mathsf{s}_{\leq \mathsf{m}'} \rceil$ by visibility. It follows that $\lceil \mathsf{s}_{\leq \mathsf{m}'} \rceil$ and $\lceil \mathsf{s}_{\leq \mathsf{m}'} \rceil$ commence with the same opponent move hence $\mathsf{m} \in \mathbf{thread}(\mathsf{s}) \Rightarrow \mathsf{m}' \in \mathbf{thread}(\mathsf{s})$. ■

Lemma 259 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ we have $\lceil \mathsf{s} \rceil = \lceil \mathbf{thread}(\mathsf{s}) \rceil$.

Proof This follows immediately from the fact that every move in $\lceil \mathsf{s} \rceil$ is also in $\mathbf{thread}(\mathsf{s})$ by inspection of the definition of $\mathbf{thread}(-)$. ■

Lemma 260 Given a sequence $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ and moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ such that $\mathsf{m} \curvearrowleft \mathsf{m}'$ then it follows that:

- If $\mathsf{m}, \mathsf{m}' \in \mathbf{thread}(\mathsf{s})$ then $\mathsf{m} \in \lfloor \mathbf{thread}(\mathsf{s})_{\leq \mathsf{m}'} \rfloor$.
- If $\mathsf{m}, \mathsf{m}' \notin \mathbf{thread}(\mathsf{s})$ then the subsequence of moves strictly between m and m' that is comprised of only those moves from $\mathbf{thread}(\mathsf{s})$ is either empty or of the form $\mathsf{n} \cdot \mathsf{t} \cdot \mathsf{n}'$ where n' is a player move, n is an opponent move and $\mathsf{n} \in \lfloor \mathbf{thread}(\mathsf{s})_{\leq \mathsf{n}'} \rfloor$.

Proof We carry out a proof by induction on the length of the subsequence of s consisting of the moves lying strictly between m and m'.

Base Case If the subsequence is empty then the lemma is simple.

Inductive Step Let us first consider the case when m' is an opponent move. As $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ we know that $\lfloor \mathsf{s}_{\leq \mathsf{m}'} \rfloor$ will be of the form

$$\cdots \mathsf{m} \cdot \mathsf{o}_1 \overset{\curvearrowleft}{\cdot} \mathsf{p}_1 \cdots \mathsf{o}_n \overset{\curvearrowleft}{\cdot} \mathsf{p}_n \cdot \mathsf{m}'.$$

We apply the inductive hypothesis to each pair of moves $\mathsf{o}_i \cdot \mathsf{p}_i$ so that the subsequence lying inclusively between each pair that is comprised of moves from $\mathbf{thread}(\mathsf{s})$ is either empty or of the form $\mathsf{n} \cdot \mathsf{t} \cdot \mathsf{n}'$ where n' is a player move, n is an opponent move and $\mathsf{n} \in \lfloor \mathbf{thread}(\mathbf{s})_{\leq \mathsf{n}'} \rfloor$. We then examine the definition of opponent view to prove the lemma.

Now we consider the case when m' is a player move. If $\mathsf{m}, \mathsf{m}' \in \mathbf{thread}(\mathsf{s})$ then $\mathsf{m} \in \lfloor \mathbf{thread}(\mathbf{s})_{\leq \mathsf{m}'} \rfloor$ follows directly from the definition of opponent view. Otherwise, if $\mathsf{m}, \mathsf{m}' \notin \mathbf{thread}(\mathsf{s})$ then as $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ we know that $\lceil \mathsf{s}_{\leq \mathsf{m}'} \rceil$ will be of the following form.

$$\cdots \mathsf{m} \cdot \mathsf{p}_1 \overset{\curvearrowleft}{\cdot} \mathsf{o}_1 \cdots \mathsf{p}_n \overset{\curvearrowleft}{\cdot} \mathsf{o}_n \cdot \mathsf{m}'$$

The definition of $\mathbf{thread}(-)$ ensures that none of the pairs of moves $\mathsf{p}_i \overset{\curvearrowleft}{\cdot} \mathsf{o}_i$ is in $\mathbf{thread}(\mathsf{s})$. We apply the inductive hypothesis to each such pair of moves so that the subsequence lying strictly between each pair that comprises of moves from $\mathbf{thread}(\mathsf{s})$ is either empty or of the form $\mathsf{n} \cdot \mathsf{t} \cdot \mathsf{n}'$ where n' is a player move, n is an opponent move and $\mathsf{n} \in \lfloor \mathbf{thread}(\mathbf{s})_{\leq \mathsf{n}'} \rfloor$. Again we examine the definition of opponent view to prove the lemma. ■

Lemma 261 Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ with moves $\mathsf{m}, \mathsf{m}' \in \mathbf{thread}(\mathsf{s})$ it follows that:

$$\mathsf{m} \overset{*}{\prec}_{\mathsf{s}} \mathsf{m}' \Leftrightarrow \mathsf{m} \overset{*}{\prec}_{\mathbf{thread}(\mathsf{s})} \mathsf{m}'$$

Proof We only prove the implication in the left to right direction as the other direction is similar.

Our proof is by induction on the length of s.

Base Case $s = \varepsilon$ and the lemma is trivially satisfied.

Inductive Step We know that $m \overset{*}{\prec_s} m'$ so we either have $m = m'$ and we also have $m \overset{*}{\prec_{\mathbf{thread}(s)}} m'$, or we have some move $k \in s$ such that

$$m \overset{*}{\prec_s} k \prec_s m'.$$

Note that m' cannot be an initial opponent move as there would be no move preceding it in $\mathbf{thread}(s)$ it is therefore straightforward to check that k is also in $\mathbf{thread}(s)$ and hence $k \prec_{\mathbf{thread}(s)} m'$. We can now apply our inductive hypothesis to $s_{\leq k}$ to yield $m \overset{*}{\prec_{\mathbf{thread}(s)}} k$ and hence $m \overset{*}{\prec_{\mathbf{thread}(s)}} m'$ by transitivity. ■

Lemma 262 Given sequences $s, s' \in \mathcal{H}_A$ we have

$$(\lceil s \rceil_\prec = \lceil s' \rceil_\prec) \Rightarrow (\lceil \mathbf{thread}(s) \rceil_\prec = \lceil \mathbf{thread}(s') \rceil_\prec).$$

Proof We construct a proof by induction on the length of s. To ease the proof we strengthen the inductive hypothesis to state that the bijection that renders $\lceil \mathbf{thread}(s) \rceil_\prec = \lceil \mathbf{thread}(s') \rceil_\prec$ is a restriction of that which renders $\lceil s \rceil_\prec = \lceil s' \rceil_\prec$.

Base Case If $s = \varepsilon$ then the lemma is trivial.

Inductive Step Let $s = t \cdot n$. If $\lceil s \rceil_\prec = \lceil s' \rceil_\prec$ then s' must be of the form $t' \cdot n$.

case: If n is a player move we have:

$$\begin{array}{lll} & \lceil t \cdot n \rceil_\prec = \lceil t' \cdot n \rceil_\prec & \\ \Rightarrow & \lceil t \rceil_\prec = \lceil t' \rceil_\prec & \text{by the definition of } \lceil - \rceil_\prec \\ \Rightarrow & \lceil \mathbf{thread}(t) \rceil_\prec = \lceil \mathbf{thread}(t') \rceil_\prec & \text{by inductive hypothesis} \\ \Rightarrow & \lceil \mathbf{thread}(t) \rceil_\prec = \lceil \mathbf{thread}(t') \rceil_\prec & \text{by the definition of } \mathbf{thread}(-). \end{array}$$

case: If n is an initial opponent move then the lemma is straightforward.

case: Suppose n is a non-initial opponent move. From the definition of equality of SCI-views we see that for any move occurrence $n' \in s$ such that $n' \prec_s n$ we have a corresponding occurrence $n' \in s'$ such that $\lceil s_{n'} \rceil_\prec = \lceil s_n \rceil_\prec$. Inspection of the definition of $\prec$ informs us that if n is not initial then $n' \in \mathbf{thread}(s)$ and $n' \in \mathbf{thread}(s')$. By inductive hypothesis we have $\lceil \mathbf{thread}(s_{n'}) \rceil_\prec = \lceil \mathbf{thread}(s'_{n'}) \rceil_\prec$ so it is simple to extend the bijection. ■

Lemma 263 Given a sequence $s \in \mathcal{S}_A$ we have:

$$\begin{array}{rcl} s \in \mathcal{V}_A & \Rightarrow & \mathbf{thread}(s) \in \mathcal{V}_A. \\ s \in \mathcal{L}_A & \Rightarrow & \mathbf{thread}(s) \in \mathcal{L}_A. \\ s \in \mathcal{H}_A & \Rightarrow & \mathbf{thread}(s) \in \mathcal{H}_A. \\ s \in I_A & \Rightarrow & \mathbf{thread}(s) \in I_A. \end{array}$$

Proof Suppose $s \in \mathcal{V}_A$. Note that by definition $\mathbf{thread}(s)$ commences with an opponent move.

Every non-initial opponent move $m \in \mathbf{thread}(s)$ has a justifier by construction and every non-initial player move $m \in \mathbf{thread}(s)$ has a justifier in $\lceil s_m \rceil$ by the visibility of s and hence in $\mathbf{thread}(s)$ by lemma 259. Lemma 259 also ensures that player respects the visibility condition in $\mathbf{thread}(s)$ and lemma 260 ensures that opponent respects visibility hence $\mathbf{thread}(s) \in \mathcal{V}_A$.

If $s \in \mathcal{L}_A$ then we have already shown that $\mathbf{thread}(s) \in \mathcal{V}_A$. It is clear that $\mathbf{thread}(s)$ is also well bracketed as we see that any pair of moves $q, a \in s$ such that a answers q are either both included in $\mathbf{thread}(s)$ or both absent.

Now suppose $s \in \mathcal{H}_A$. We know that $\mathbf{thread}(s) \in \mathcal{L}_A$ so we only need to check that opponent respects the nesting condition in $\mathbf{thread}(s)$. Suppose we have two opponent moves $m, m' \in \mathbf{thread}(s)$ such that $m \leftrightsquigarrow m'$ with $m < m$, and either m or m' active. Then as $s \in \mathcal{H}_A$ we know that m must be answered, by some move $a \in s$ before m' is played. Lemma 258 assures us that $a \in \mathbf{thread}(s)$ and hence $\mathbf{thread}(s) \in \mathcal{H}_A$.

Finally suppose $s \in I_A$. We know that $\mathbf{thread}(s) \in \mathcal{H}_A$ so we only need to check that the SCI-condition is not violated and that player respects the nesting condition in $\mathbf{thread}(s)$. Suppose we have two player moves $m, m' \in \mathbf{thread}(s)$ such that $m \leftrightsquigarrow m'$ with $m < m$, and either m or m' active. Then as $s \in I_A$ we know that m must be answered, by some move $a \in s$ before m' is played. Again by lemma 258 we have $a \in \lceil s \rceil$ and hence player respects the nesting condition. Lemma 261 ensures that the SCI-condition is respected in $\mathbf{thread}(s)$ hence $\mathbf{thread}(s) \in I_A$. ■

As with the category $\mathbf{A}^T_{\mathcal{L}}$, thread independent strategies are completely determined by the single threaded sequences they contain. We show that the function $-_{\mathsf{threads}}$ is injective by defining its inverse.

Definition 264 Given a non-empty set $S \subseteq \mathcal{H}_A$ of even-length single threaded sequences S such that $s \cdot m \cdot m' \in S \Rightarrow s \in S$ we define the set $\mathsf{weave}(S)$ as the least set such that:

1. $\varepsilon \in \mathsf{weave}(S)$.
2. Given $s \in \mathsf{weave}(S)$ and an extension $s \cdot m \in \mathcal{H}_A$ such that there exists a sequence $\exists \mathbf{thread}(s \cdot m \cdot m') \in S$ then we have $s \cdot m \cdot m' \in \mathsf{weave}(S)$.

Lemma 265 Given an SCI-strategy $\sigma : A$ it follows that

$$\mathsf{weave}(\sigma_{\mathsf{threads}}) = \sigma.$$

Proof First we show that for all $s \in \mathsf{weave}(\sigma_{\mathsf{threads}})$ we have $s \in \sigma$ by induction on the length of s.

Base Case $\varepsilon \in \mathsf{weave}(\sigma_{\mathsf{threads}})$ and $\varepsilon \in \sigma$.

Inductive Step Let $s = s' \cdot m \cdot n$. From the definition of $\mathsf{weave}(-)$ we must have $s' \in \mathsf{weave}(\sigma_{\mathsf{threads}})$ and a thread $t \cdot m' \cdot n' \in \sigma_{\mathsf{threads}}$ such that $s' \cdot m$ is in I_A and $t \cdot m = \mathbf{thread}(s' \cdot m)$. By inductive hypothesis we have $s' \in \sigma$ and by the corollary of lemma 263 we must also have $t \cdot m \cdot n \in \sigma$ hence by the thread-independence of σ it follows that $s' \cdot m \cdot n \in \sigma$.

We now show that for all $s \in \sigma$ we have $s \in \mathsf{weave}(\sigma_{\mathsf{threads}})$ by induction on the length of s.

Base Case Once again $\varepsilon \in \mathsf{weave}(\sigma_{\mathsf{threads}})$ and $\varepsilon \in \sigma$.

Inductive Step Let $s = s' \cdot m \cdot n$. By induction hypothesis we have $s' \in \mathsf{weave}(\sigma_{\mathsf{threads}})$ and clearly we have $s' \cdot m \in I_A$ and $\mathbf{thread}(s' \cdot m \cdot n) \in \sigma_{\mathsf{threads}}$ hence by the definition of $\mathsf{weave}(-)$ we have $s' \cdot m \cdot n \in \mathsf{weave}(\sigma_{\mathsf{threads}})$. ■

We now examine the conditions that we must place on a set of single threaded sequences T so that $\mathsf{weave}(T)$ is an SCI strategy.

Lemma 266 Given a set T of even length single threaded sequences $T \subseteq \mathcal{S}_A$ such that the thread rules **1-6** hold it follows that $\mathsf{weave}(T)$ is an SCI-strategy for A.

Proof The definition of $\mathsf{weave}(-)$ ensures that $\mathsf{weave}(T)$ is a prefixed closed set of sequences of even length.

We now prove that for any sequence $s \in \mathsf{weave}(T)$ we have $s \in I_A$ by induction on the length of s.

Base Case If $s = \varepsilon$ then $s \in I_A$.

Inductive Step Let $s = s' \cdot m \cdot m'$. Inspection of the definition of $\mathsf{weave}(-)$ implies that $s' \cdot m \in I_A$. Furthermore, we have the following:

$\mathbf{thread}(s' \cdot m \cdot m') \in T$			by the definition of $\mathsf{weave}(-)$
	$\Rightarrow$	$\mathbf{thread}(s' \cdot m \cdot m') \in I_A s'$	by **T1**
	$\Rightarrow$	$\lceil \mathbf{thread}(s' \cdot m \cdot m') \rceil \in I_A$	by lemma 252
	$\Rightarrow$	$\lceil s' \cdot m \cdot m' \rceil \in I_A$	by lemma 257
	$\Rightarrow$	$s' \cdot m \cdot m' \in \mathcal{L}_A$	by lemma 251.

We now must check that m' respects the nesting condition. Suppose we have any move $n \in s'$ such that $n \leftrightsquigarrow m'$. We have two cases to consider:

case: Suppose $n \in \mathbf{thread}(s' \cdot m \cdot m')$. We have already shown that $\mathbf{thread}(s' \cdot m \cdot m') \in I_A$ so if either m' or n are active it must be that n is answered before n is played in $\mathbf{thread}(s' \cdot m \cdot m')$ and hence also in $s' \cdot m \cdot m'$.

case: Now suppose $n \notin \mathbf{thread}(s' \cdot m \cdot m')$. As $n \leftrightsquigarrow m'$ it follows that both moves are initial. Let $\mathbf{thread}(s' \cdot m \cdot m')$ commence with the move q'_o and $\mathbf{thread}(s'_{\leq n})$ commence with the move q_o. As n and m' are in different threads we know that there must be some consecutive pair of moves $k, k^+ \in s'$ such that:

- $n \leq k$.
- $n \in \mathbf{thread}(s'_{\leq k})$.
- $k^+ \notin \mathbf{thread}(s' \cdot m \cdot m')$.

By lemma 168 we know that k must be in the same strand as the initial move in q_o. If n is active then by thread rule **T5** we know that $q_o \sim q'_o$. By the activity rule we know that q_o is active. We apply the inductive hypothesis to $s_{\leq q'_o}$ to see that by the nesting condition it must be that q_o is answered before q'_o is played and hence by well bracketing it follows that n is also answered and s respects the nesting condition. If m' is active then it follows from thread rule **T6** that either $q_o \sim q'_o$

and our reasoning is similar to the above or else n is answered in $\mathsf{s}_{\leq \mathsf{k}}$ and the nesting condition is obeyed or else.

Lastly we must check that $\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}'$ respects the SCI-condition. Suppose we have some move $\mathsf{n} \in \mathsf{s}'$ such that $\mathsf{n} \lhd \mathsf{m}'$. Again we have two cases to consider:

case: Suppose $\mathsf{n} \in \mathbf{thread}(\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}')$. We know that $\mathsf{n} \overset{*}{\prec}_{\mathbf{thread}(\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}')} \mathsf{m}'$ by the SCI-condition and thus $\mathsf{n} \overset{*}{\prec}_{\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}'} \mathsf{m}'$ by lemma 261.

case: Alternatively, suppose $\mathsf{n} \notin \mathbf{thread}(\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}')$ then it must be the case that n is the answer to some active initial player question q_P such that $\mathsf{q}_P \sim \mathsf{m}'$. Let q_O be the first move in $\mathbf{thread}(\mathsf{s}'_{\leq \mathsf{n}})$ and let q'_O be the first move in $\mathbf{thread}(\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}')$. Once again, as n and m' are in different threads we know that there must be some consecutive pair of moves $\mathsf{k}, \mathsf{k}^+ \in \mathsf{s}'$ such that:

- $\mathsf{n} \leq \mathsf{k}$.
- $\mathsf{n} \in \mathbf{thread}(\mathsf{s}'_{\leq \mathsf{k}})$.
- $\mathsf{k}^+ \notin \mathbf{thread}(\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}')$.

By lemma 168 we know that k must be in the same strand as the initial move in q_O. Hence by thread rule **T5** it must be that $\mathsf{q}_O \sim \mathsf{q}'_O$. We can apply the inductive hypothesis and from the activity condition that q_O is active so by the nesting condition it must be that q_O is answered by some move a_P before q'_O is played. By definition we have $\mathsf{n} \lhd \mathsf{a}_P$ and hence $\mathsf{n} \overset{*}{\prec} \mathsf{a}_P$ by the inductive hypothesis and the SCI-condition. We have $\mathsf{a}_P \prec \mathsf{q}'_O$ by definition and $\mathsf{q}'_O \overset{*}{\prec} \mathsf{m}'$ by lemma 246 hence $\mathsf{n} \overset{*}{\prec} \mathsf{m}'$ by transitivity.

We now show that $\mathsf{weave}(T)$ is deterministic. Suppose we have sequences $\mathsf{s} \cdot \mathsf{m}, \mathsf{s}' \cdot \mathsf{m}' \in \mathsf{weave}(T)$ such that $\mathsf{s} = \mathsf{s}'$ then inspection of the definition of $\mathsf{weave}(-)$ reveals that there must be sequences $\mathbf{thread}(\mathsf{s} \cdot \mathsf{m}), \mathbf{thread}(\mathsf{s}' \cdot \mathsf{m}') \in T$ and hence $\mathbf{thread}(\mathsf{s}) = \mathbf{thread}(\mathsf{s}')$ and by condition **T3** we have $\mathbf{thread}(\mathsf{s} \cdot \mathsf{m}) = \mathbf{thread}(\mathsf{s}' \cdot \mathsf{m}')$ and hence $\mathsf{s} \cdot \mathsf{m} = \mathsf{s}' \cdot \mathsf{m}'$.

We finally must show that $\mathsf{weave}(T)$ is SCIR coherent. Suppose we have sequences $\mathsf{t}, \mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in \mathsf{weave}(T)$ such that $\mathsf{t} \cdot \mathsf{m} \in \mathcal{H}_{\mathsf{A}}$ and $\lceil \mathsf{t} \cdot \mathsf{m} \rceil_{\prec} = \lceil \mathsf{s} \cdot \mathsf{m} \rceil_{\prec}$. From the definition of $\mathsf{weave}(-)$ we have $\mathbf{thread}(\mathsf{t}), \mathbf{thread}(\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}') \in T$. By lemma 263 we have $\mathbf{thread}(\mathsf{t} \cdot \mathsf{m}) \in \mathcal{H}_{\mathsf{A}}$ and by lemma 262 we have $\lceil \mathbf{thread}(\mathsf{t} \cdot \mathsf{m}) \rceil_{\prec} = \lceil \mathbf{thread}(\mathsf{s} \cdot \mathsf{m}) \rceil_{\prec}$. By condition **T4** we therefore have $\lceil \mathbf{thread}(\mathsf{t} \cdot \mathsf{m}) \rceil_{\prec} \cdot \mathsf{m}' \in T$ and hence $\mathsf{t} \cdot \mathsf{m} \cdot \mathsf{m}' \in \mathsf{weave}(T)$ by the definition of $\mathsf{weave}(-)$.

■

6.5 The Categorical Structure of C

6.5.1 The Terminal Object

We now examine the categorical structure of **C**.

Definition 267 The **null game** is defined as follows

$$\mathbf{null} = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle.$$

For any object A in **C** we have exactly one strategy for $\mathsf{A} \to \mathbf{null}$, the strategy containing only the empty sequence, hence **null** is terminal in **C**.

6.5.2 The Symmetric Monoid

Definition 268 The category $\mathbf{C}$ is symmetric monoidal. We define the tensor bifunctor $\otimes$ with **null** as its unit. The object $A \otimes B$ is defined as follows:

$$\begin{array}{rcl} M_{A\otimes B} & = & M_A + M_B \\ \lambda_{A\otimes B} & = & [\lambda_A, \lambda_B] \\ m \vdash_{A\otimes B} m' & \Leftrightarrow & m \vdash_A m' \vee m \vdash_B m' \\ \star \vdash_{A\otimes B} m' & \Leftrightarrow & \star \vdash_A m' \vee \star \vdash_B m' \\ m \sim_{A\otimes B} m' & \Leftrightarrow & m \sim_A m' \vee m \sim_B m' \end{array}$$

Given strategies $\sigma : A \to B$ and $\tau : C \to D$ we define the set $\sigma \otimes \tau$ as follows

$$\sigma \otimes \tau = \mathsf{weave}(\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}).$$

Given SCI strategies $\sigma : A \to B$ and $\tau : C \to D$ the set $\sigma \otimes \tau$ is an SCI-strategy.

Proof First we note, by lemma 256 that $\sigma_{\mathsf{threads}}$ and τ_{threads} obey thread rules **T1-6**. It is straightforward to now check that the set $\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}$ also obeys thread rules **T1-6**. Lemma 266 now assures us that the construction is an SCI strategy $\sigma \otimes \tau : (A \otimes B) \to (C \otimes D)$. ■

It is simple to check that $\otimes$ is indeed bifunctorial; $\mathbf{copy}^I_A \otimes \mathbf{copy}^I_B = \mathbf{copy}^I_{A\otimes B}$ and $(\sigma \otimes \sigma'); (\tau \otimes \tau') = (\sigma;\tau) \otimes (\sigma';\tau')$.

6.5.3 The Exponentials

Unfortunately, we do not have symmetric monoidal closure in $\mathbf{C}$ but we do have sufficient to model the SCI languages that we are interested in.

Definition 269 Given SCI arenas A, B and we define the arena $A \multimap B$ as follows:

$$\begin{array}{rcl} M_{A\multimap B} & = & M_A + M_B \\ \lambda_{A\multimap B} & = & [\overline{\lambda_A}, \lambda_B] \\ m \vdash_{A\multimap B} m' & \Leftrightarrow & (m \vdash_A m') \vee (m \vdash_B m') \\ & & \text{when } m \neq \star \wedge \star \not\vdash_A m' \\ m \vdash_{A\multimap B} m' & \Leftrightarrow & (\lambda^{\pi\alpha} m' = \alpha \Rightarrow \lambda^{\pi\alpha} m = \alpha) \\ & & \text{when } \star \vdash_B m \wedge \star \vdash_A m' \\ \star \vdash_{A\multimap B} m' & \Leftrightarrow & \star \vdash_B m' \\ \sim_{A\multimap B} & = & \sim_A + \sim_B \end{array}$$

As we have already indicated, we do not have every exponential, but we can internalize the arrows into two interesting kinds of objects.

Lemma 270 Given SCI arenas A, B and C such that for any initial moves in $m, m' \in M_C$ we have $m \sim m'$ it follows that the SCI strategies for the arena $(A \otimes B) \to C$ are in one to one correspondence with the strategies for the arena $A \to (B \multimap C)$, and hence that $B \multimap C$ is the exponential of A by B.

Proof Suppose we are given an SCI strategy $\sigma : (A \otimes B) \to C$. It follows from lemma 256

that $\sigma_{\mathsf{threads}}$ obeys thread rules **T1-6**. We can transform each of these threads to form the set $\sigma' \subseteq I_{A\to(B\multimap C)}$ by a simple adjustment of the tagging that was introduced when forming the disjoint unions and by attaching a justification pointer from all initial moves from M_B to the initial opponent move in the thread. It is then simple to check that the resultant σ' obeys thread rules **T1-6** and hence we know that $\mathsf{weave}(\sigma')$ is an SCI strategy by lemma 266. Similarly it is simple to define the inverse of the operation taking σ to $\mathsf{weave}(\sigma')$. Given an SCI strategy $\tau : A \to (B \multimap C)$ we can transform each sequence in τ_{threads} to form the set $\tau^* \subseteq I_{(A\otimes B)\to C}$ by removing the justifiers from initial moves from M_A. We then check that τ^* respects the thread rules **T1-6** and apply lemma 266. Note here that our insistence on initial moves from C being in the $\sim$ relationship with each other is needed in order to ensure that τ^* respects **T5** and **T6**. ■

Lemma 271 Given SCI arenas A, B and C such that all initial moves from M_C are passive it follows that the SCI strategies for the arena $(A \otimes B) \to C$ are in one to one correspondence with the strategies for the arena $A \to (B \multimap C)$, and hence that $B \multimap C$ is the exponential of A by B.

Proof The proof is similar to that for lemma 270. ■

6.5.4 The Products

Definition 272 Given arenas A and B we define the arena $A \times B$ as follows:

$$\begin{array}{rcl} M_{A\times B} & = & M_A + M_B \\ \lambda_{A\times B} & = & [\lambda_A, \lambda_B] \\ m \vdash_{A\times B} m' & \Leftrightarrow & m \vdash_A m' \vee m \vdash_B m' \\ \star \vdash_{A\times B} m' & \Leftrightarrow & \star \vdash_A m' \vee \star \vdash_B m' \\ m \sim_{A\times B} m' & \Leftrightarrow & m \sim_A m' \vee m \sim_B m' \\ & & \vee(\star \vdash_A m \wedge \star \vdash_B m') \end{array}$$

Note that the product is similar to the tensor except in the definition of $\sim_{A\times B}$. Intuitively, the two components of a product may contain interfering terms, but the two components of a tensor may not.

Given SCI strategies $\sigma : A \to C$ and $\tau : B \to D$ we define the set

$$\sigma \times \tau = \mathsf{weave}(\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}).$$

This set $\sigma \times \tau$ is an SCI strategy for the arena $(A \times B) \to (C \times D)$.

Proof First we note, by lemma 256 that $\sigma_{\mathsf{threads}}$ and τ_{threads} obey thread rules **T1-6**. It is simple to check that the set $\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}$ also obeys thread rules **T1-6** when given the labeling from arena $(A \times B) \to (C \times D)$. Lemma 266 now assures us that the construction is an SCI strategy $\sigma \times \tau : (A \times B) \to (C \times D)$. ■

It is simple to check that $\times$ is indeed bifunctorial; $\mathbf{copy}^I_A \times \mathbf{copy}^I_B = \mathbf{copy}^I_{A\times B}$ and $(\sigma \times \sigma'); (\tau \times \tau') = (\sigma;\tau) \times (\sigma';\tau')$.

Lemma 273 The category $\mathbf{C}$ has finite products.

Proof The projections $\pi_1 : A \times B \to A$ and $\pi_2 : A \times B \to B$ are defined by the obvious copycat

strategies. Note that these strategies are comprised of exactly the same sequences as the identities $\mathbf{copy}^I_{\mathsf{A}}$ and $\mathbf{copy}^I_{\mathsf{B}}$. It is simple to use lemmas 256 and 266 to demonstrate that these strategies are well defined. Given strategies $\sigma : \mathsf{C} \to \mathsf{A}$ and $\tau : \mathsf{C} \to \mathsf{B}$ we form the pairing $\langle \sigma, \tau \rangle : \mathsf{C} \to \mathsf{A} \times \mathsf{B}$ as follows

$$\langle \sigma, \tau \rangle = \mathsf{weave}(\sigma_{\mathsf{threads}} + \tau_{\mathsf{threads}}).$$

Again we can use lemmas 256 and 266 to demonstrate that $\langle \sigma, \tau \rangle : \mathsf{C} \to \mathsf{A} \times \mathsf{B}$ is a well defined SCI strategy. It is simple to show that $\langle \sigma, \tau \rangle ; \pi_1 = \sigma$ and $\langle \sigma, \tau \rangle ; \pi_2 = \tau$.

To prove that the objects we have defined are indeed products we show *surjective pairing*: given a strategy $\sigma : \mathsf{C} \to \mathsf{A} \times \mathsf{B}$ there exist strategies $\sigma_A : \mathsf{C} \to \mathsf{A}$ and $\sigma_B : \mathsf{C} \to \mathsf{B}$ such that $\sigma = \langle \sigma_A, \sigma_B \rangle$.

From lemma 256 we know that $\sigma_{\mathsf{threads}}$ obeys the thread rules **1-6**. It is possible to separate the nonempty sequences in $\sigma_{\mathsf{threads}}$ into two disjoint sets $\sigma^A_{\mathsf{threads}}$ and $\sigma^B_{\mathsf{threads}}$ depending on whether the unique initial opponent move is from arena A or from arena B. It is straightforward to check that both $\sigma^A_{\mathsf{threads}}$ and $\sigma^B_{\mathsf{threads}}$ obey thread rules **T1-6** and hence by lemma 266 we have SCI strategies $\mathsf{weave}(\sigma^A_{\mathsf{threads}}) : \mathsf{C} \to \mathsf{A}$ and $\mathsf{weave}(\sigma^B_{\mathsf{threads}}) : \mathsf{C} \to \mathsf{B}$ and hence

$$\sigma = \langle \mathsf{weave}(\sigma^A_{\mathsf{threads}}), \mathsf{weave}(\sigma^B_{\mathsf{threads}}) \rangle.$$

■

Note that given arenas A, B and C and SCI strategies $\sigma : \mathsf{A} \to \mathsf{B}$ and $\tau\mathsf{A} \to \mathsf{C}$ we should not, in general, have a pairing map for the arena $\mathsf{A} \to (\mathsf{B} \otimes \mathsf{C})$ — it is a completeness property that the product and the monoid in our category be distinguished. Our proof of the existence of a pairing map for products fails in the case of the tensor because the union of the set of single threaded plays from σ and τ does not in general respect thread rules **T5-6**.

6.5.5 The Cartesian Closed Passive Subcategory

Definition 274 We define **P** to be the full subcategory of **C** containing only those arenas which have no active initial opponent moves. We write the inclusion functor as $J : \mathbf{P} \hookrightarrow \mathbf{C}$.

Lemma 275 The monoidal structure in **C** restricts to a product structure on **P**. Given arenas $J\mathsf{A}$ and $J\mathsf{B}$ we have

$$J\mathsf{A} \otimes J\mathsf{B} \simeq J\mathsf{A} \times J\mathsf{B}.$$

Proof The isomorphism and its inverse are constructed from the obvious copycat strategies. We know that $\mathbf{copy}^I_{\mathsf{JA}_{\mathsf{threads}}}$ and $\mathbf{copy}^I_{\mathsf{JB}_{\mathsf{threads}}}$ respect thread rules **T1-6** and it is then trivial to show that $\mathbf{copy}^I_{\mathsf{JA}_{\mathsf{threads}}} + \mathbf{copy}^I_{\mathsf{JB}_{\mathsf{threads}}}$ also respect the thread rules when given labelling from either arena $J\mathsf{A} \otimes J\mathsf{B}$ or arena $J\mathsf{A} \times J\mathsf{B}$ and hence by lemma 266 we know that our proposed copycat isomorphism and its inverse are SCI strategies. We should note that it is the passivity of the initial player moves that ensures that thread rules **T5-6** are obeyed when we take the union of the identity strategies. ■

Lemma 276 The category **P** is cartesian closed.

Proof It is straightforward to show that the initial object **null** is in **P** and we have already shown

that **P** has products. For any arenas JA and JB we see that $A \times B$ is in **P**. Lemma 271 assures us that for all arenas JA and JB we have the well defined exponential $JA \to JB$ and by inspection this object is in **P**. We know from lemma 275 that monoids in **C** restrict to products in **P** hence **P** is cartesian closed. ■

6.5.6 A Retractive Model of SCIR

We now show that **C** has the categorical structure described in section 5.4.3.

Definition 277 Given a game A we relabel all initial moves as passive to make PA. We define the game PA formally as

$$\begin{aligned}
M_{PA} &= M_A \\
\lambda_{PA}^{OPQA} &= \lambda_A^{OPQA} \\
\lambda_{PA}^{\alpha\pi} m &= \pi \text{ if } \star \vdash_A m \\
\lambda_{PA}^{\alpha\pi} m &= \lambda_A^{\alpha\pi} m \text{ if } \star \nvdash_A m \\
\vdash_{PA} &= \vdash_A \\
\sim_{PA} &= \sim_A
\end{aligned}$$

Given a strategy $\sigma : A \to B$ we define $P(\sigma) : PA \to PB$ as

$$P\sigma = \mathsf{weave}(\sigma_{\mathsf{threads}}).$$

The intention here is of course that the weaving is done with respect to the labelling from the game $PA \to PB$. We know from lemma 256 that $\sigma_{\mathsf{threads}}$ respects thread rules **T1-6** with respect to the labelling from the arena $A \to B$ and it is simple to show that it follows that $\sigma_{\mathsf{threads}}$ also respects thread rules **T1-6** with respect to the labelling from the arena $PA \to PB$. Hence by lemma 266 we know that $P\sigma$ is a well defined SCI strategy.

Furthermore, it is straightforward to show that P is functorial; $P\mathbf{copy}_A^I = \mathbf{copy}_{PA}^I$ and $P\sigma; P\tau = P(\sigma;\tau)$. Note that for any arenas A and B it follows that

$$P(A \otimes B) = PA \otimes PB,$$

in other words P is a monoidal functor.

Lemma 278 The functor P is the right adjoint to the inclusion functor J.

Proof Given any SCI strategy $\sigma : JA \to B$ we know from lemma 256 that $\sigma_{\mathsf{threads}}$ respects thread rules **T1-6**. It is straightforward to show that $\sigma_{\mathsf{threads}}$ respects the thread rules when given labelling from arena $A \to PB$ and hence $\mathsf{weave}(\sigma_{\mathsf{threads}}) : A \to PB$ is an SCI strategy for this arena by lemma 266.

Similarly given any SCI strategy $\tau : A \to PB$ we know from lemma 256 that τ_{threads} respects thread rules **T1-6** and we can check that τ_{threads} respects the thread rules when given labelling from arena $JA \to B$ and hence $\mathsf{weave}(\tau_{\mathsf{threads}}) : JA \to B$ is an SCI strategy for this arena by lemma 266.

We therefore know that there is a bijection between strategies for $J\mathsf{A} \to \mathsf{B}$ and those for $\mathsf{A} \to \mathsf{PB}$ and therefore $J \dashv \mathsf{P}$. ■

For any adjoint pair of functors $J \dashv \mathsf{P}$ where J is full and faithful we know for any object $J\mathsf{A}$ we have $\mathsf{A} \simeq \mathsf{P}J\mathsf{A}$ [35], and that the unit of the adjunction $\eta^P_{\mathsf{A}} : \mathsf{A} \to \mathsf{P}J\mathsf{A}$ is the isomorphism. In $\mathbf{C}$ the arenas are equal and the unit is the identity strategy.

It is straightforward to show that the counit of the adjunction $\varepsilon^P_{\mathsf{A}} : J\mathsf{PA} \to \mathsf{A}$ can be defined as follows

$$\varepsilon^P_{\mathsf{A}} = \mathsf{weave}(\mathbf{copy}^I_{\mathsf{A}_{\mathsf{threads}}}).$$

where it is intended that the weaving is performed with respect to the labelling from arena $J\mathsf{PA} \to \mathsf{A}$. As usual we can use lemmas 256 and 266 to show that our definition is indeed an SCI strategy and it is not hard to show that this strategy contains exactly the same plays as the identity strategy $\mathbf{copy}^I_{\mathsf{A}}$.

Definition 279 Given a game A we discard active initial moves to make SA. We define the game SA formally as follows

$$\mathsf{M}_{\mathsf{SA}} = \{\mathsf{m} \in \mathsf{M}_{\mathsf{A}} | \star \vdash_{\mathsf{A}} \mathsf{m} \Rightarrow \lambda^{\pi\alpha}_{\mathsf{A}}\mathsf{m} = \pi\}$$

and for all $\mathsf{m}, \mathsf{m}' \in \mathsf{M}_{\mathsf{SA}}$ we have the following

$$\begin{aligned} \lambda_{\mathsf{SA}}\mathsf{m} &\Leftrightarrow \lambda_{\mathsf{A}}\mathsf{m} \\ \mathsf{m} \vdash_{\mathsf{SA}} \mathsf{m}' &\Leftrightarrow \mathsf{m} \vdash_{\mathsf{A}} \mathsf{m}' \\ \mathsf{m} \sim_{\mathsf{SA}} \mathsf{m}' &\Leftrightarrow \mathsf{m} \sim_{\mathsf{A}} \mathsf{m}' \end{aligned}$$

Given a strategy $\sigma : \mathsf{A} \to \mathsf{B}$ we define

$$\mathsf{S}\sigma = \mathsf{weave}(I_{\mathsf{SA} \to \mathsf{SB}} \cap \sigma_{\mathsf{threads}}).$$

We know from lemma 256 that $\sigma_{\mathsf{threads}}$ obeys the thread rules **T1-6**. It is straightforward to show that if we discard the threads in $\sigma_{\mathsf{threads}}$ that commence with an active move that the resultant set also obeys the thread rules and hence by lemma 266 we know that $\mathsf{S}\sigma$ is a well-defined SCI strategy.

It is straightforward to show that S is a functor; $\mathsf{S}\mathbf{copy}^I_{\mathsf{A}} = \mathbf{copy}^I_{\mathsf{SA}}$ and $\mathsf{S}\sigma; \mathsf{S}\tau = \mathsf{S}(\sigma; \tau)$. Note too that for any arenas A and B it follows that

$$\mathsf{S}(\mathsf{A} \otimes \mathsf{B}) = \mathsf{SA} \otimes \mathsf{SB},$$

in other words S is a monoidal functor.

Lemma 280 The functor S is left adjoint to the inclusion functor J.

Proof Given any SCI strategy $\sigma : \mathsf{A} \to J\mathsf{B}$ we know from lemma 256 that $\sigma_{\mathsf{threads}}$ respects thread rules **T1-6**. By the activity condition we know that no active initial move from A may be played hence $\sigma_{\mathsf{threads}}$ respects the thread rules when given labelling from arena $\mathsf{SA} \to \mathsf{B}$ and hence $\mathsf{weave}(\sigma_{\mathsf{threads}}) : \mathsf{A} \to \mathsf{PB}$ is an SCI strategy for this arena by lemma 266.

Similarly given any SCI strategy $\tau : \mathsf{SA} \to \mathsf{B}$ we know from lemma 256 that τ_{threads} respects thread rules **T1-6** and we can check hat τ_{threads} respects the thread rules when given labelling from arena $\mathsf{A} \to J\mathsf{B}$ and hence $\mathsf{weave}(\tau_{\mathsf{threads}}) : \mathsf{A} \to J\mathsf{B}$ is an SCI strategy for this arena by lemma 266.

We therefore know that there is a bijection between strategies for $\mathsf{A} \to J\mathsf{B}$ and those for $\mathsf{SA} \to \mathsf{PB}$ and therefore $\mathsf{S} \dashv J$. ■

For any adjoint pair of functors $\mathsf{S} \dashv J$ where J is full and faithful we know for any arena $J\mathsf{A}$ we have $\mathsf{A} \simeq \mathsf{S}J\mathsf{A}$ [35], and that the counit of the adjunction $\varepsilon^S_{\mathsf{A}} : \mathsf{S}J\mathsf{A} \to \mathsf{A}$ is the isomorphism. In **C** the arenas are equal and the counit is the identity strategy.

It is straightforward to show that the unit of the adjunction $\eta^S_{\mathsf{A}} : \mathsf{A} \to J\mathsf{SA}$ can be defined as following copycat strategy:

$$\eta^S_{\mathsf{A}} = \mathsf{weave}(\mathbf{copy}^I_{\mathsf{A}\,\mathsf{threads}}).$$

where it is intended that the weaving is performed with respect to the labelling from arena $\mathsf{A} \to J\mathsf{SA}$. As usual we can use lemmas 256 and 266 to show that our definition is indeed an SCI strategy and it is not hard to show that this strategy contains exactly the same plays as the identity strategy $\mathbf{copy}^I_{\mathsf{SA}}$.

Definition 281 We define a map $\alpha_{\mathsf{A}} : J\mathsf{SA} \to \mathsf{A}$ as follows:

$$\alpha_{\mathsf{A}} = \mathsf{weave}(\mathbf{copy}^I_{\mathsf{SA}\,\mathsf{threads}})$$

where it is intended that the weaving is performed with respect to the labelling from arena $J\mathsf{SA} \to \mathsf{A}$. We use lemmas 256 and 266 to show that our definition is indeed an SCI strategy and it is not hard to show that this strategy also contains exactly the same plays as the identity strategy $\mathbf{copy}^I_{\mathsf{SA}}$.

Lemma 282 The category **C** and subcategory **P** form a retractive categorical model of SCIR as described in section 5.4.3. For every arena A we have the following:

$$\alpha_{\mathsf{A}}; \eta^S_{\mathsf{A}} = \mathbf{copy}^I_{\mathsf{SA}}.$$

Proof As already stated both strategies $\alpha_{\mathsf{A}} : J\mathsf{SA} \to \mathsf{A}$ and η^S_{A} contain exactly the same plays as the identity strategy $\mathbf{copy}^I_{\mathsf{SA}}$ and so the proof is trivial. ■

6.6 Denotational Semantics in C

We are going to construct sound models of **PCF**, $\mathbf{SCI}_b$ and **SCIR** in **C**. There is a certain amount of overlap when we define the three models so before we construct the models we will define some SCI arenas and strategies that we are going to use to model the types and constructs in our languages.

Definition 283 (Flat SCI Arenas) For any basic data type τ, with corresponding set of values S_τ, we define two *flat* arenas:

- The passive flat arena τ_π

$$\begin{array}{rcl} M_{\tau_\pi} & = & \{q\} \cup V_\tau \\ \lambda_{\tau_\pi} q & = & OQ\pi \\ \lambda_{\tau_\pi} n & = & PA\pi \\ \star & \vdash_{\tau_\pi} & q \\ q & \vdash_{\tau_\pi} & n \\ \sim_{\tau_\pi} & = & \{(q,q)\} \end{array}$$

- And the active flat arena τ_α

$$\begin{array}{rcl} M_{\tau_\alpha} & = & \{q\} \cup V_\tau \\ \lambda_{\tau_\alpha} q & = & OQ\alpha \\ \lambda_{\tau_\alpha} n & = & PA\alpha \\ \star & \vdash_{\tau_\alpha} & q \\ q & \vdash_{\tau_\alpha} & n \\ \sim_{\tau_\alpha} & = & \{(q,q)\} \end{array}$$

For the purpose of our illustrative languages we will be particularly interested in the flat arenas where $\tau = \mathbb{N}$, the set of natural numbers. We write $\mathbf{N}_\pi$ for $\mathbb{N}_\pi$ and $\mathbf{N}_\alpha$ for $\mathbb{N}_\alpha$. We are also interested in the active flat arena for the singleton data type for which we write **com** and also run and done for the unique initial move and corresponding unique answer. Note that the names of our arenas may coincide with the types in the languages; no confusion will arise in practice.

Definition 284 (Variable Arenas) Similarly, for each basic data type τ we define two variable arenas.

- The passive variable arena $\mathbf{var}^\pi_\tau$

$$\begin{array}{rcl} M_{\mathbf{var}^\pi_\tau} & = & \{ok, q\} \cup \tau \cup \{write_n | n \in \tau\} \\ \lambda_{\mathbf{var}^\pi_\tau} write_n & = & OQ\alpha \\ \lambda_{\mathbf{var}^\pi_\tau} ok & = & PA\alpha \\ \lambda_{\mathbf{var}^\pi_\tau} q & = & OQ\pi \\ \lambda_{\mathbf{var}^\pi_\tau} n & = & PA\pi \\ \star & \vdash_{\mathbf{var}^\pi_\tau} & write_n \\ \star & \vdash_{\mathbf{var}^\pi_\tau} & q \\ write_n & \vdash_{\mathbf{var}^\pi_\tau} & ok \\ q & \vdash_{\mathbf{var}^\pi_\tau} & n \\ m \sim_{\mathbf{var}^\pi_\tau} m' & \Leftrightarrow & \star \vdash_{\mathbf{var}^\pi_\tau} m' \wedge \star \vdash_{\mathbf{var}^\pi_\tau} m' \end{array}$$

- and the active variable arena $\mathbf{var}^\alpha_\tau$

$$\begin{array}{rcl} M_{\mathbf{var}} & = & \{ok, q\} \cup \tau \cup \{write_n | n \in \tau\} \\ \lambda_{\mathbf{var}^\alpha_\tau} write_n, q & = & OQ\alpha \\ \lambda_{\mathbf{var}^\alpha_\tau} ok, n & = & PA\alpha \\ \star & \vdash_{\mathbf{var}^\alpha_\tau} & write_n \end{array}$$

$$\begin{aligned} \star &\vdash_{\mathbf{var}^\alpha_\tau} \mathsf{q} \\ \mathsf{write}_n &\vdash_{\mathbf{var}^\alpha_\tau} \mathsf{ok} \\ \mathsf{q} &\vdash_{\mathbf{var}^\alpha_\tau} \mathsf{n} \\ \mathsf{m} \sim_{\mathbf{var}^\alpha_\tau} \mathsf{m}' &\Leftrightarrow \star \vdash_{\mathbf{var}^\alpha_\tau} \mathsf{m}' \wedge \star \vdash_{\mathbf{var}^\alpha_\tau} \mathsf{m}' \end{aligned}$$

For the purposes of our illustrative languages we will be particularly interested in variable arenas for the basic data type of natural numbers. We will simply write $\mathbf{var}^\pi$ for $\mathbf{var}^\pi_{\mathbb{N}}$ and $\mathbf{var}^\alpha$ for $\mathbf{var}^\alpha_{\mathbb{N}}$.

We will now define some strategies that we will use to model the constructs in the languages. It is important to note that these strategies all have their analogues in $\mathbf{A}^T_{\mathcal{L}}$ and we will define them in terms of the single-threaded plays that their analogues contain. It is simple to check that these single-threaded plays obey the thread-rules **T1-6** and by lemma 266 we may apply the $\mathsf{weave}(-)$ operator to our sets of single-threaded sequences to yield well defined SCI strategies.

Definition 285 Each of these strategies is defined in terms of a strategy in $\mathbf{A}^T_{\mathcal{L}}$ which has been defined in chapter 4. We call each strategy from $\mathbf{A}^T_{\mathcal{L}}$ that we use in each definition the *analogue* of the strategy in $\mathbf{C}$ that we are defining.

$$\begin{aligned} \mathbf{n}_\pi : \mathbf{N}_\pi &= \mathsf{weave}(\mathbf{n}_{\mathsf{threads}}) \\ &\quad \text{for each natural number } n. \\ \mathbf{n}_\alpha : \mathbf{N}_\alpha &= \mathsf{weave}(\mathbf{n}_{\mathsf{threads}}) \\ &\quad \text{for each natural number } n. \\ \mathbf{s}_\pi : \mathbf{N}_\pi \to \mathbf{N}_\pi &= \mathsf{weave}(\mathbf{s}_{\mathsf{threads}}). \\ \mathbf{s}_\pi : \mathbf{N}_\alpha \to \mathbf{N}_\alpha &= \mathsf{weave}(\mathbf{s}_{\mathsf{threads}}). \\ \mathbf{p}_\pi : \mathbf{N}_\pi \to \mathbf{N}_\pi &= \mathsf{weave}(\mathbf{p}_{\mathsf{threads}}). \\ \mathbf{p}_\alpha : \mathbf{N}_\alpha \to \mathbf{N}_\alpha &= \mathsf{weave}(\mathbf{p}_{\mathsf{threads}}). \\ \mathbf{ifz}_{\pi\mathsf{A}} : \mathbf{N}_\pi \times \mathsf{A} \times \mathsf{A} \to \mathsf{A} &= \mathsf{weave}(\mathbf{ifz}_{\mathsf{A}\mathsf{threads}}) \\ &\quad \text{for each flat SCI arena } \mathsf{A}. \\ \mathbf{ifz}_{\alpha\mathsf{A}} : \mathbf{N}_\alpha \times \mathsf{A} \times \mathsf{A} \to \mathsf{A} &= \mathsf{weave}(\mathbf{ifz}_{\mathsf{A}\mathsf{threads}}) \\ &\quad \text{for each flat active SCI arena } \mathsf{A}. \\ \mathbf{assign}_\pi : \mathbf{var}^\pi \times \mathbf{N}_\pi \to \mathbf{com} &= \mathsf{weave}(\mathbf{assign}_{\mathsf{threads}}). \\ \mathbf{assign}_\alpha : \mathbf{var}^\alpha \times \mathbf{N}_\alpha \to \mathbf{com} &= \mathsf{weave}(\mathbf{assign}_{\mathsf{threads}}). \\ \mathbf{deref}_\pi : \mathbf{var}^\pi \to \mathbf{N}_\pi &= \mathsf{weave}(\mathbf{deref}_{\mathsf{threads}}). \\ \mathbf{deref}_\alpha : \mathbf{var}^\alpha \to \mathbf{N}_\alpha &= \mathsf{weave}(\mathbf{deref}_{\mathsf{threads}}). \\ \mathbf{mkvar}_\pi : (\mathbf{N}_\pi \multimap \mathbf{com}) \times \mathbf{N}_\pi \to \mathbf{var}^\pi &= \mathsf{weave}(\mathbf{mkvar}_{\mathsf{threads}}). \\ \mathbf{mkvar}_\alpha : (\mathbf{N}_\alpha \multimap \mathbf{com}) \times \mathbf{N}_\alpha \to \mathbf{var}^\alpha &= \mathsf{weave}(\mathbf{mkvar}_{\mathsf{threads}}). \\ \mathbf{seq}_{\alpha\mathsf{A}} : \mathbf{com} \times \mathsf{A} \to \mathsf{A} &= \mathsf{weave}(\mathbf{seq}_{\mathsf{threads}}). \\ &\quad \text{for each flat active SCI arena } \mathsf{A}. \\ \mathbf{while}_\alpha : \mathbf{N}_\alpha \times \mathbf{com} \to \mathbf{com} &= \mathsf{weave}(\mathbf{while}_{\mathsf{threads}}). \\ \mathbf{while}_\pi : \mathbf{N}_\pi \times \mathbf{com} \to \mathbf{com} &= \mathsf{weave}(\mathbf{while}_{\mathsf{threads}}). \end{aligned}$$

$$\begin{aligned}
\mathbf{new}_{\pi} : (\mathbf{var}^{\pi} \multimap \mathbf{com}) \to \mathbf{com} &= \mathsf{weave}(\mathbf{new}_{\mathsf{threads}}).\\
\mathbf{new}_{\alpha} : (\mathbf{var}^{\alpha} \multimap \mathbf{com}) \to \mathbf{com} &= \mathsf{weave}(\mathbf{new}_{\mathsf{threads}}).\\
\mathbf{do}_{\pi} : \mathsf{P}(\mathbf{var}^{\pi} \multimap \mathbf{com}) \to \mathbf{N}_{\pi} &= \mathsf{weave}(\mathbf{do}_{\mathsf{threads}}).\\
\mathbf{do}_{\alpha} : (\mathbf{var}^{\alpha} \multimap \mathbf{com}) \to \mathbf{N}_{\alpha} &= \mathsf{weave}(\mathbf{do}_{\mathsf{threads}}).\\
\mathbf{rec}_{\pi\mathsf{A}} : \mathsf{P}(\mathsf{A} \multimap \mathsf{A}) \to \mathsf{A} &= \mathsf{weave}(\mathbf{rec}_{\mathsf{A}\,\mathsf{threads}})\\
&\quad \text{for each arena } \mathsf{A}.
\end{aligned}$$

We should note some facts about these definitions:

- We sometimes define more than one strategy from the same analogue. This is due to the different base types that we are going to have in our example languages. For example in $\mathbf{SCI}_b$ our natural number types are active whereas in **SCIR** they are passive. We do not bother to define versions of the strategies for passive command types as we do not include them in any of our languages as they do not seem to have an obvious practical purpose.
- Note that we only have a passive analogue of the **rec** strategy.
- Note the arena for which $\mathbf{do}_{\pi}$ is a strategy.
- It is simple to show that for any SCI strategy $c : \mathsf{A}$, defined here, it follows that

  $$c : \mathsf{A} = c' \cap \mathcal{H}_{\mathsf{A}}$$

  where c' is the analogue of c in $\mathbf{A}^{T}_{\mathcal{L}}$.

Lemma 286 Given negative SCI arenas A, B and C and strategies for the underlying QA arenas $\sigma \subseteq \mathcal{L}_{\mathsf{A}\to\mathsf{B}}$ and $\tau \subseteq \mathcal{L}_{\mathsf{B}\to\mathsf{C}}$ such that $\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}}$ is an SCI strategy for the SCI arena $\mathsf{A} \to \mathsf{B}$ and $\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}$ is an SCI strategy for the arena $\mathsf{B} \to \mathsf{C}$ it follows that:

$$(\sigma;\tau) \cap \mathcal{H}_{\mathsf{A}\to\mathsf{C}} = (\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}});(\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}).$$

Note that the composition on the left is performed in **A** whereas the one on the right is performed in **C**.

Proof First we show that

$$(\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}});(\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}) \subseteq (\sigma;\tau) \cap \mathcal{H}_{\mathsf{A}\to\mathsf{C}}.$$

It should be clear that

$$(\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}});(\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}) \subseteq (\sigma;\tau)$$

and we already know from lemma 238 that

$$(\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}});(\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}) \subseteq \mathcal{H}_{\mathsf{A}\to\mathsf{C}}$$

as $\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}}$ and $\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}$ are SCI strategies.

We now show that

$$(\sigma;\tau) \cap \mathcal{H}_{\mathsf{A}\to\mathsf{C}} \subseteq (\sigma \cap \mathcal{H}_{\mathsf{A}\to\mathsf{B}});(\tau \cap \mathcal{H}_{\mathsf{B}\to\mathsf{C}}).$$

Consider any sequence $s \in (\sigma;\tau) \cap \mathcal{H}_{A\to C}$ and a sequence $u \in \sigma\|\tau$ that witnesses s. It will suffice to show that $u \upharpoonright A,B \in \mathcal{H}_{A\to B}$ and $u \upharpoonright B,C \in \mathcal{H}_{B\to C}$. We prove by induction on the length of an arbitrary prefix $t \sqsubseteq u$ that $t \upharpoonright A,B \in \mathcal{H}_{A\to B}$ and $t \upharpoonright B,C \in \mathcal{H}_{B\to C}$.

Base Case If $t = \varepsilon$ then the proof is simple.

Inductive Step Suppose $t = t' \cdot m$. If $m \in t \upharpoonright A$ then by inductive hypothesis we know that $t' \upharpoonright B,C \in \mathcal{H}_{B\to C}$ and hence $t' \cdot m \upharpoonright B,C \in \mathcal{H}_{B\to C}$. Furthermore we can apply the inductive hypothesis $t' \upharpoonright A,B \in \mathcal{H}_{A\to B}$ and as $u \upharpoonright A,C \in \mathcal{H}_{A\to C}$ it follows from the bias of the referee $\mathcal{L}$ that $u \upharpoonright A \in \mathcal{L}_A$ and again by the bias of the referee $\mathcal{L}$ that $t' \cdot m \upharpoonright A,B \in \mathcal{L}_{A\to B}$. It is now straightforward to check that $t' \cdot m \upharpoonright A,B \in \mathcal{H}_{A\to B}$ as no opponent nesting occurs in $u \upharpoonright A,C \in \mathcal{H}_{A\to C}$. Our proof is similar if $m \in t \upharpoonright C$. Lastly we must consider the case when $m \in t \upharpoonright B$. We assume that m is an opponent move in σ and the proof is similar when m is an opponent move in τ. By inductive hypothesis we know that $t' \upharpoonright B,C \in \mathcal{H}_{B\to C}$ and hence $t' \cdot m \upharpoonright B,C \in \mathcal{H}_{B\to C}$. We know that $\tau \cap \mathcal{H}_{B\to C}$ is an SCI strategy hence $t' \cdot m \upharpoonright B,C \in I_{B\to C}$ hence $t' \cdot m \upharpoonright B$ respects both player and opponent nesting. By inductive hypothesis we know $t' \upharpoonright A,B \in \mathcal{H}_{A\to B}$ and hence $t' \cdot m \upharpoonright A,B \in \mathcal{H}_{A\to B}$. ■

6.6.1 The Denotational Semantics of PCF

The interpretation of **PCF** in **C** has very much in common with the interpretation of **PCF** in the innocent subcategory of $\mathbf{A}^T_{\mathcal{L}}$. We will use only the passive subcategory **P** to interpret **PCF**.

6.6.2 The Interpretation of Types

A **PCF** type A will be modelled in **C** by a negative SCI arena $[\![A]\!]_P$ in which all moves are passive. We start by defining the arena corresponding to natural number type which in our illustrative language will be the sole base type **N**. We define $[\![\mathbf{N}]\!]_P = \mathbf{N}_\pi$. The interpretation of higher types is defined inductively:

- $[\![A \times B]\!]_P = [\![A]\!]_P \times [\![B]\!]_P$.
- $[\![A \Rightarrow B]\!]_P = [\![A]\!]_P \multimap [\![B]\!]_P$.

6.6.3 The Interpretation of Typing Judgements

A typed term

$$x_1 : A_1 \ldots x_n : A_n \vdash_P M : A$$

will be modelled by a morphism

$$[\![A_1]\!]_P \times \ldots \times [\![A_n]\!]_P \to [\![A]\!]_P$$

and a closed term $\vdash_P M : A$ will be modelled by a strategy for the arena $1 \to [\![A]\!]_P$. We should note here that the semantics of a typing judgement will always be an arrow in **P** and that every move in the strategy will be passive.

6.6.4 Interpreting the λ-calculus

The λ-calculus part of **PCF** is interpreted using the Cartesian closed structure demonstrated in lemma 276. Identifiers are interpreted as projections:

$$[\![x_1 : A_1, \ldots, x_n : A_n \vdash_P x_i : A_i]\!]_P = \pi_i : [\![A_1]\!]_P \times \ldots \times [\![A_n]\!]_P \to [\![A_i]\!]_P.$$

Abstraction is modelled by currying:

$$[\![\Gamma \vdash_P M : A \Rightarrow B]\!]_P = \Lambda([\![\Gamma, x : A \vdash_P M : B]\!]_P) : [\![\Gamma]\!]_P \to [\![A \Rightarrow B]\!]_P.$$

Application is modelled by composition with the evaluation map:

$$[\![\Gamma \vdash_P MN : B]\!]_P = \langle [\![\Gamma \vdash_P M : A \Rightarrow B]\!]_P, [\![\Gamma \vdash_P N : A]\!]_P \rangle ; \mathsf{eval}_{[\![A \Rightarrow B]\!]_P}.$$

Pairing and projection are modelled using the categorical product:

$$[\![\Gamma \vdash_P \langle M, N \rangle : A \times B]\!]_P = \langle [\![\Gamma \vdash_P M : A]\!]_P, [\![\Gamma \vdash_P N : B]\!]_P \rangle.$$

$$[\![\Gamma \vdash_P \pi_i N : A]\!]_P = [\![\Gamma \vdash_P N : A \times B]\!]_P ; \pi_i.$$

6.6.5 Interpreting PCF Terms

$$\begin{array}{rcl}
[\![\Gamma \vdash_P n]\!]_P & = & \mathbf{n}_\pi \\
[\![\Gamma \vdash_P \mathsf{succ}\, n]\!]_P & = & [\![\Gamma \vdash_P \mathsf{n}]\!]_P ; \mathbf{s}_\pi \\
[\![\Gamma \vdash_P \mathsf{pred}\, n]\!]_P & = & [\![\Gamma \vdash_P \mathsf{n}]\!]_P ; \mathbf{p}_\pi \\
[\![\Gamma \vdash_P \mathsf{if}\, L = 0\, \mathsf{then}\, M\, \mathsf{else}\, N]\!]_P & = & \langle [\![\Gamma \vdash_P M]\!]_P, [\![\Gamma \vdash_P M]\!]_P, [\![\Gamma \vdash_P N]\!]_P \rangle ; \mathbf{ifz}_\pi
\end{array}$$

As with the model for $\mathbf{IA}_a$ in chapter 4, to interpret terms of the form $\Gamma \vdash_P \Upsilon\, M : \theta$ we appeal to the cpo-enriched nature of our category. Given a term $\Gamma \vdash_P M : A \Rightarrow A$ we can uncurry the interpretation to yield a strategy:

$$\sigma : [\![\Gamma]\!]_P \times [\![A]\!]_P \to M : [\![A]\!]_P.$$

We can now define the following chain of strategies:

$$\begin{array}{rcl}
\sigma_0 & = & \bot \\
\sigma_{n+1} & = & \langle \mathbf{id}_{[\![\Gamma]\!]_P}, \sigma_n \rangle ; \sigma
\end{array}$$

and we interpret $\Gamma \vdash_P \Upsilon\, M : \theta$ as the least upper bound of this chain.

6.6.6 Soundness and Adequacy

Lemma 287 Given a **PCF** judgement $\Gamma \vdash_P M : A$ it follows that

$$[\![\Gamma \vdash_P M : A]\!]_P = [\![\Gamma \vdash_A M : A]\!]_A$$

Proof The proof is by a simple induction on the derivation of $\Gamma \vdash_P M : A$ using the observation

that there are no active moves in the strategy ■

Lemma 288 Our model is operationally sound:

$$M\Downarrow_P V \Rightarrow [\![M]\!]_P = [\![V]\!]_P.$$

Lemma 289 Our model is computationally adequate:

$$[\![M]\!]_P \neq \perp \Rightarrow M\Downarrow_P.$$

Proof The proof follows from the computational adequacy of the model $\mathbf{A}^T_{\mathcal{L}}$ and from lemma 287. ■

As for the model in $\mathbf{A}^T_{\mathcal{L}}$, operational soundness and computational adequacy together imply equational soundness.

6.7 The Denotational Semantics of $\mathbf{SCI}_b$

6.7.1 The Interpretation of Types

As usual, an $\mathbf{SCI}_b$ type A will be modelled in $\mathbf{C}$ by a negative SCI arena $[\![A]\!]_B$. In this case all moves are active. We start by defining the arenas corresponding to base types:

$$\begin{aligned} [\![\mathbf{N}]\!]_B &= \mathbf{N}_\alpha \\ [\![\mathbf{var}]\!]_B &= \mathbf{var}^\alpha \\ [\![\mathbf{com}]\!]_B &= \mathbf{com}. \end{aligned}$$

As with **PCF**, the interpretation of higher types is defined inductively:

$$[\![A \times B]\!]_B = [\![A]\!]_B \times [\![B]\!]_B \text{ and } [\![A \multimap B]\!]_B = [\![A]\!]_B \multimap [\![B]\!]_B.$$

Note that by lemma 270 we have all the necessary exponentials.

6.7.2 The Interpretation of Typing Judgements

A typed term

$$x_1 : A_1 \ldots x_n : A_n \vdash_B M : A$$

will be modelled by an SCI strategy for the SCI arena

$$[\![A_1]\!]_B \otimes \ldots \otimes [\![A_n]\!]_B \to [\![A]\!]_B$$

and a closed term $\vdash_B M : A$ will be modelled by an SCI strategy for the arena $1 \to [\![A]\!]_B$.

6.7.3 Interpreting the Affine λ-calculus

The λ-calculus part of $\mathbf{SCI}_b$ is interpreted using the symmetric monoidal structure and the exponentials. Identifiers are interpreted as projections:

$$[\![x_1 : A_1, \ldots, x_n : A_n \vdash_B x_i : A_i]\!]_B = \pi_i : [\![A_1]\!]_B \otimes \ldots \otimes [\![A_n]\!]_B \to [\![A_i]\!]_B.$$

Abstraction is modelled by currying:

$$[\![\Gamma \vdash_B M : A \multimap B]\!]_B = \Lambda([\![\Gamma, x : A \vdash_B M : B]\!]_B) : [\![\Gamma]\!]_B \to [\![A \multimap B]\!]_B.$$

Application is modelled by composition with the evaluation map:

$$[\![\Gamma, \Delta \vdash_B MN : B]\!]_B = [\![\Gamma \vdash_B M : A \multimap B]\!]_B \otimes [\![\Delta \vdash_B N : A]\!]_B ; \mathsf{eval}_{[\![A \multimap B]\!]_B}.$$

Pairing is modelled using the pairing map:

$$[\![\Gamma \vdash_B \langle M, N \rangle : A \times B]\!]_B = \langle [\![\Gamma \vdash_B M : A]\!]_B, [\![\Gamma \vdash_B N : B]\!]_B \rangle.$$

Projection is interpreted by composition with the projection map:

$$[\![\Gamma \vdash_B \pi_i N : A]\!]_B = [\![\Gamma \vdash_B N : A \times B]\!]_B ; \pi_i.$$

6.7.4 Interpreting the $\mathbf{SCI}_b$ Language Constructs

The semantics of terms the Algol like constructs are defined as follows:

$$\begin{aligned}
[\![\Gamma \vdash_B n]\!]_B &= \mathbf{n}_\alpha \\
[\![\Gamma \vdash_B \mathsf{succ}\, n]\!]_B &= [\![\Gamma \vdash_B \mathsf{n}]\!]_B ; \mathbf{s}_\alpha \\
[\![\Gamma \vdash_B \mathsf{pred}\, n]\!]_B &= [\![\Gamma \vdash_B \mathsf{n}]\!]_B ; \mathbf{p}_\alpha \\
[\![\Gamma \vdash_B M := N]\!]_B &= \langle [\![\Gamma \vdash_B M]\!]_B, [\![\Gamma \vdash_B N]\!]_B \rangle ; \mathbf{assign}_\alpha \\
[\![\Gamma \vdash_B !M]\!]_B &= [\![\Gamma \vdash_B M]\!]_B ; \mathbf{deref}_\alpha \\
[\![\Gamma \vdash_B \mathsf{mkvar}\, M\, N]\!]_B &= \langle [\![\Gamma \vdash_B M]\!]_B, [\![\Gamma \vdash_B N]\!]_B \rangle ; \mathbf{mkvar}_\alpha \\
[\![\Gamma \vdash_B M; N]\!]_B &= \langle [\![\Gamma \vdash_B M]\!]_B, [\![\Gamma \vdash_B N]\!]_B \rangle ; \mathbf{seq}_\alpha \\
[\![\Gamma \vdash_B \mathsf{while}\, M = 0\, \mathsf{do}\, N]\!]_B &= \langle [\![\Gamma \vdash_B M]\!]_B, [\![\Gamma \vdash_B N]\!]_B \rangle ; \mathbf{while}_\alpha \\
[\![\Gamma \vdash_B \mathsf{if}\, L = 0\, \mathsf{then}\, M\, \mathsf{else}\, N]\!]_B &= \langle [\![\Gamma \vdash_B M]\!]_B, [\![\Gamma \vdash_B M]\!]_B, [\![\Gamma \vdash_B N]\!]_B \rangle ; \mathbf{ifz}_\alpha \\
[\![\Gamma \vdash_B \mathsf{new}\, x := 0\, \mathsf{in}\, M]\!]_B &= [\![\Gamma \vdash_B \lambda x.M]\!]_B ; \mathbf{new}_\alpha \\
[\![\Gamma \vdash_B \mathsf{do}\, x := 0\, \mathsf{then}\, M]\!]_B &= [\![\Gamma \vdash_B \lambda x.M]\!]_B ; \mathbf{do}_\alpha
\end{aligned}$$

6.7.5 Soundness and Adequacy

As with our interpretation of **PCF** in **C**, we can use the soundness and adequacy of the model of $\mathbf{IA}_a$ in chapter 4. However our proof must be a little more subtle as the nesting constraints placed upon opponent in **C** imply that the strategy used to interpret an $\mathbf{SCI}_b$ term in **C** will not be equal to the strategy used to interpret the same term in $\mathbf{A}^T_{\mathcal{L}}$.

Lemma 290 Given a $\mathbf{SCI}_b$ judgement $\Gamma \vdash_B M : A$ it follows that

$$[\![\Gamma \vdash_B M : A]\!]_P = [\![\Gamma \vdash_A M : A]\!]_A \cap \mathcal{H}_{\mathsf{A}}.$$

Proof The proof is by a simple induction on the derivation of $\Gamma \vdash_P M : A$ making use of

lemma 286. ■

Lemma 291 Our model is operationally sound:

$$M\Downarrow_B V \Rightarrow [\![M]\!]_B = [\![V]\!]_B.$$

Proof From the definition of the $\Downarrow_B$ we know that if $M\Downarrow_B V$ then $M\Downarrow_A V$ and $[\![M]\!]_A = [\![V]\!]_A$ from the soundness of the interpretation of $\mathbf{IA}_a$ in $\mathbf{A}^T_{\mathcal{L}}$. By lemma 290 we therefore have $[\![M]\!]_B = [\![V]\!]_B$. ■

Lemma 292 Our model is computationally adequate:

$$[\![M]\!]_B \neq \perp \Rightarrow M\Downarrow_B.$$

Proof If $[\![M]\!]_B \neq \perp$ then obviously $[\![M]\!]_A \neq \perp$ by lemma 290 and hence $M\Downarrow_A$ by the adequacy of the model of $\mathbf{IA}_a$ in $\mathbf{A}^T_{\mathcal{L}}$ and hence by definition it follows that $M\Downarrow_B$. ■

As for the model in $\mathbf{A}^T_{\mathcal{L}}$, operational soundness and computational adequacy together imply that we have an equationally sound model of $\mathbf{SCI}_b$.

6.8 The Denotational Semantics of SCIR

6.8.1 The Interpretation of Types

As usual, an **SCIR** type A will be modelled in $\mathbf{C}$ by a negative SCI arena $[\![A]\!]_S$. In this case moves may be either active or passive. We start by defining the arenas corresponding to base types:

$$\begin{aligned} [\![\mathbf{N}]\!]_S &= \mathbf{N}_\pi \\ [\![\mathbf{var}]\!]_S &= \mathbf{var}^\pi \\ [\![\mathbf{com}]\!]_S &= \mathbf{com}. \end{aligned}$$

As with **PCF**, the interpretation of higher types is defined inductively:

$$[\![A \times B]\!]_S = [\![A]\!]_S \times [\![B]\!]_S \text{ and } [\![A \multimap B]\!]_S = [\![A]\!]_S \multimap [\![B]\!]_S.$$

Note that we know from lemma 270 that the necessary exponentials exist.

6.8.2 The Interpretation of Typing Judgements

A typed term

$$x_1 : A_1 \ldots x_{k-1} : A_{k-1} \mid x_k : A_k \ldots x_n : A_n \vdash_S M : A$$

will be modelled by an SCI strategy for the SCI arena

$$\mathsf{S}[\![A_1]\!]_S \otimes \ldots \mathsf{S} \otimes [\![A_{k-1}]\!]_S \otimes [\![A_k]\!]_S \ldots [\![A_n]\!]_S \to [\![A]\!]_S$$

and a closed term $\vdash_S M : A$ will be modelled by an SCI strategy for the arena $1 \to [\![A]\!]_S$. We should remind ourselves here that our semantics is constructed by induction on the typing derivation so we need a rule for constructing the semantics of a term corresponding to each typing rule.

6.8.3 Interpreting the Affine λ-calculus

The λ-calculus part of **SCIR** is interpreted using the symmetric monoidal closed structure. Identifiers are interpreted as identities:

$$[\![- \mid x : A \vdash_S x : A]\!]_S = \mathsf{id}_{[\![A]\!]_S}.$$

Abstraction is modelled by currying:

$$[\![\Gamma \mid \Delta \vdash_S M : A \multimap B]\!]_S = \Lambda([\![\Gamma \mid \Delta, x : A \vdash_S M : B]\!]_S) : [\![\Gamma \mid \Delta]\!]_S \to [\![A \multimap B]\!]_S.$$

Application is modelled by composition with the evaluation map:

$$[\![\Gamma, \Gamma' \mid \Delta, \Delta' \vdash_S MN : B]\!]_S = [\![\Gamma \mid \Delta \vdash_S M : A \multimap B]\!]_S \otimes [\![\Gamma' \mid \Delta' \vdash_S N : A]\!]_S; \mathsf{eval}_{[\![A \multimap B]\!]_S}.$$

Pairing and projection are modelled using the categorical product:

$$[\![\Gamma \vdash_S \langle M, N \rangle : A \times B]\!]_S = \langle [\![\Gamma \vdash_S M : A]\!]_S, [\![\Gamma \vdash_S N : B]\!]_S \rangle.$$

$$[\![\Gamma \vdash_S \pi_i N : A]\!]_S = [\![\Gamma \vdash_S N : A \times B]\!]_S; \pi_i.$$

6.8.4 Interpreting the SCIR Structural Rules

We use the reflective and coreflective subcategorical structure of **C** to model the **SCIR** structural rules.

Activation is modelled by precomposition with the unit of the adjunction $\mathsf{S} \dashv J$:

$$[\![\Gamma \mid x : A, \Delta \vdash_S M : B]\!]_S = (\mathbf{id}_\Gamma \otimes \eta^S_{\mathsf{A}} \otimes \mathbf{id}_\Delta); [\![\Gamma, x : A \mid \Delta \vdash_S M : B]\!]_S.$$

Passification is modelled by precomposition with the section map:

$$[\![\Gamma, x : A \mid \Delta \vdash_S M : B]\!]_S = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]_S} \otimes \alpha\mathsf{A} \otimes \mathbf{id}_{[\![\Delta]\!]_S}); [\![\Gamma \mid x : A, \Delta \vdash_S M : B]\!]_S.$$

Dereliction is modelled by composition with the counit of the adjunction $J \dashv \mathsf{P}$:

$$[\![\Gamma \mid \Delta \vdash_S \mathsf{derelict}\, M : A]\!]_S = [\![\Gamma \mid\vdash_S M : \mathsf{P}B]\!]_S; \varepsilon^P_{\mathsf{A}}.$$

Promotion is modelled by the adjunction $J \dashv \mathsf{P}$:

$$[\![\Gamma \mid\vdash_S \mathsf{promote}\, M : \mathsf{P}A]\!]_S = \mathsf{P}([\![\Gamma \mid\vdash_S M : A]\!]_S).$$

Weakening is interpreted using the fact that the terminal object is the unit of the tensor.

$$[\![\Gamma, \Gamma' \mid \Delta, \Delta' \vdash_S M : A]\!]_S = \mathsf{id}_{\mathsf{S}[\![\Gamma]\!]_S} \otimes ! \otimes \mathsf{id}_{[\![\Delta]\!]_S} \otimes !; [\![\Gamma \mid \Delta \vdash_S M : A]\!]_S.$$

Finally contraction is modelled by precomposition with the contraction map $\Delta : \mathsf{SA} \to \mathsf{SA} \otimes \mathsf{SA}$:

$$[\![\Gamma, z : A \mid \Delta \vdash_S M : B]\!]_S = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]_S} \otimes \Delta \otimes \mathbf{id}_{[\![\Delta]\!]_S}); [\![\Gamma, x : A, y : A \mid \Delta \vdash_S M : B]\!]_S.$$

We know that the contraction map exists because $\mathsf{SA} \otimes \mathsf{SA} = \mathsf{SA} \times \mathsf{SA}$.

6.8.5 Interpreting the SCIR Language Constructs

The semantics of terms the Algol like constructs of **SCIR** are defined as follows:

$$\begin{aligned}
[\![\Gamma \vdash_S n]\!]_S &= \mathbf{n}_\pi \\
[\![\Gamma \vdash_S \mathsf{succ}\ n]\!]_S &= [\![\Gamma \vdash_S \mathsf{n}]\!]_S ; \mathbf{s}_\pi \\
[\![\Gamma \vdash_S \mathsf{pred}\ n]\!]_S &= [\![\Gamma \vdash_S \mathsf{n}]\!]_S ; \mathbf{p}_\pi \\
[\![\Gamma \vdash_S M := N]\!]_S &= \langle [\![\Gamma \vdash_S M]\!]_S, [\![\Gamma \vdash_S N]\!]_S \rangle ; \mathbf{assign}_\pi \\
[\![\Gamma \vdash_S !M]\!]_S &= [\![\Gamma \vdash_S M]\!]_S ; \mathbf{deref}_\pi \\
[\![\Gamma \vdash_S \mathsf{mkvar}\ M\ N]\!]_S &= \langle [\![\Gamma \vdash_S M]\!]_S, [\![\Gamma \vdash_S N]\!]_S \rangle ; \mathbf{mkvar}_\pi \\
[\![\Gamma \vdash_S M;N]\!]_S &= \langle [\![\Gamma \vdash_S M]\!]_S, [\![\Gamma \vdash_S N]\!]_S \rangle ; \mathbf{seq}_\alpha \\
[\![\Gamma \vdash_S \mathsf{while}\ M = 0\ \mathsf{do}\ N]\!]_S &= \langle [\![\Gamma \vdash_S M]\!]_S, [\![\Gamma \vdash_S N]\!]_S \rangle ; \mathbf{while}_\pi \\
[\![\Gamma \vdash_S \mathsf{if}\ L = 0\ \mathsf{then}\ M\ \mathsf{else}\ N]\!]_S &= \langle [\![\Gamma \vdash_S M]\!]_S, [\![\Gamma \vdash_S M]\!]_S, [\![\Gamma \vdash_S N]\!]_S \rangle ; \mathbf{ifz}_\pi \\
[\![\Gamma \vdash_S \mathsf{new}\ x := 0\ \mathsf{in}\ M]\!]_S &= [\![\Gamma \vdash_S \lambda x.M]\!]_S ; \mathbf{new}_\pi \\
[\![\Gamma \vdash_S \mathsf{do}\ x := 0\ \mathsf{then}\ M]\!]_S &= [\![\Gamma \vdash_S \lambda x.M]\!]_S ; \mathbf{do}_\pi
\end{aligned}$$

6.8.6 Soundness and Adequacy

As with our interpretation of $\mathbf{SCI}_b$ in **C**, we can use the soundness and adequacy of the model of $\mathbf{IA}_a$ in chapter 4. As **SCIR** is not an extension of $\mathbf{IA}_a$ we make use of the translation from **SCIR** terms to $\mathbf{IA}_a$ terms that we defined in chapter 4.

Lemma 293 Given an **SCIR** judgement $\Gamma \mid \Delta \vdash_S M : A$ it follows that

$$[\![\Gamma \mid \Delta \vdash_S M : A]\!]_S = [\![(\Gamma \mid \Delta)^* \vdash_A M^* : A^*]\!]_A \cap \mathcal{H}_{\mathsf{A}}.$$

Proof The proof is by a simple induction on the derivation of $\Gamma \mid \Delta \vdash_S M : A$ making use of lemma 286 ■

Lemma 294 Our model is operationally sound:

$$M \Downarrow_S V \Rightarrow [\![M]\!]_B = [\![V]\!]_B.$$

Proof From lemma 216 we know that if $M \Downarrow_S V$ then $M^* \Downarrow_A V^*$ and we know that $[\![M]\!]_A = [\![V]\!]_A$ from the soundness of the interpretation of $\mathbf{IA}_a$ in $\mathbf{A}^T_{\mathcal{L}}$. By lemma 293 we therefore have $[\![M]\!]_S = [\![V]\!]_S$. ■

Lemma 295 Our model is computationally adequate:

$$[\![M]\!]_B \neq \bot \Rightarrow M \Downarrow_S.$$

Proof If $[\![M]\!]_S \neq \bot$ then obviously $[\![M^*]\!]_A \neq \bot$ by lemma 293 and hence $M^* \Downarrow_A$ by the adequacy of the model of $\mathbf{IA}_a$ in $\mathbf{A}^T_{\mathcal{L}}$ and hence by lemma 216 it follows that $M \Downarrow_S$. ■

As for the model in $\mathbf{A}^T_{\mathcal{L}}$, operational soundness and computational adequacy together imply that we have an equationally sound model of **SCIR**.

Chapter 7

Definability

7.1 Why Definability?

We have so far shown that we can give a sound and adequate interpretation of **PCF**, $\mathbf{SCI}_b$ and **SCIR** in **C**. However, it is straightforward to show that we also have a sound and adequate interpretation of these languages in $\mathbf{A}^T_{\mathcal{L}}$. What then was the motivation behind the more complicated definitions and structure that we have imposed upon the model **C**?

It is certainly the case that more observationally equivalent terms of both $\mathbf{SCI}_b$ and **SCIR** are equated in our model than would be in $\mathbf{A}^T_{\mathcal{L}}$, our model is more abstract, but note that this extra equivalence is not necessarily on terms that we would first imagine equating. We give the following $\mathbf{SCI}_b$ terms as an example:

$$f : (\mathbf{com} \multimap \mathbf{com}) \multimap \mathbf{com} \vdash_B \mathsf{new}\, x := 0 \,\mathsf{in}\, f(\lambda y^{\mathbf{com}}.\mathsf{if}\, x = 0 \,\mathsf{then}\quad x := 1; y; x := 0 \quad \mathsf{else} \quad \Omega)$$

and the rather simpler term

$$f : (\mathbf{com} \multimap \mathbf{com}) \multimap \mathbf{com} \vdash_B f(\lambda y^{\mathbf{com}}.y).$$

The interpretations of these terms in **C** are equal, and as such the terms must be observationally equivalent, but note that they are not observationally equivalent as terms of $\mathbf{IA}_a$. The equivalence in **C** is due to the restriction on opponent nesting: a context can only distinguish these terms by using an additive function application. An example $\mathbf{IA}_a$ value that may be substituted for f which distinguishes these terms is

$$\lambda g^{\mathbf{com} \multimap \mathbf{com}}.g(g\,\mathsf{skip}).$$

Note that the resultant term is not typeable in $\mathbf{SCI}_b$.

However, some of the more straightforward equivalences, those that lie at the very heart of the syntactic control of interference, are *not* equated in our model. Consider as an example the observational equivalence of the $\mathbf{SCI}_b$ terms:

$$(x : \mathbf{com}, y : \mathbf{com} \vdash_B x; y) \simeq (x : \mathbf{com}, y : \mathbf{com} \vdash_B y; x).$$

We readily see that these terms are not equivalent in our model, however there exist models of $\mathbf{SCI}_b$ where the interpretation of these terms is equivalent, for example the object spaces model [46] that was later given an alternative formulation and shown to be fully abstract in [37].

The important property that our model does possess, however, is one that is not shared by the models of [46, 37]: namely definability. Any morphism (obeying certain restrictions that we will describe later) between the semantics of a context and the semantics of a type of one of our target languages will be the denotation of a term in that language. As with our model of $\mathbf{IA}_a$ in chapter 4, we will show in section 7.10 how this property leads to full abstraction. This indirect method of using definability to achieve full abstraction seems in some ways unsatisfactory, but due to a result in [33] we know that no fully abstract model of finitary **PCF** is effectively presentable, and we will show in section 7.10 that **SCIR** is a conservative extension of **PCF**, hence no fully abstract model of **SCIR** is effectively presentable.

7.2 No Adequate Bireflective Model of SCIR has the Definability Property

In this section we will show that any bireflective model must inevitably have undefinable morphisms. We show that the unit of the adjunction $J \dashv \mathsf{P}$ is not generally definable in any adequate model of **SCIR**.

Suppose we consider any term $- \mid x : \mathbf{com} \vdash_S M : \mathsf{P}\mathbf{com}$. It follows from the activity condition that player can play no active initial move in

$$[\![- \mid x : \mathbf{com} \vdash_S M : \mathsf{P}\mathbf{com}]\!]_S$$

therefore the strategy contains no move from the subarena $[\![\mathbf{com}]\!]_S$ and it is straightforward to show that for any M

$$[\![- \mid \vdash_S \mathsf{skip} : \mathbf{com}]\!]_S ; [\![- \mid x : \mathbf{com} \vdash_S M : \mathsf{P}\mathbf{com}]\!]_S$$

is always equal to

$$[\![- \mid - \vdash_S \Omega : \mathbf{com}]\!]_S ; [\![- \mid x : \mathbf{com} \vdash_S M : \mathsf{P}\mathbf{com}]\!]_S.$$

It therefore follows that

$$[\![- \mid - \vdash_S M[\mathsf{skip}/x] : \mathsf{P}\mathbf{com}]\!]_S = [\![- \mid - \vdash_S M[\Omega/x] : \mathsf{P}\mathbf{com}]\!]_S$$

and hence

$$[\![- \mid - \vdash_S M[\mathsf{skip}/x] : \mathsf{P}\mathbf{com}]\!]_S ; \varepsilon^P_{\mathbf{com}} = [\![- \mid - \vdash_S M[\Omega/x] : \mathsf{P}\mathbf{com}]\!]_S ; \varepsilon^P_{\mathbf{com}}.$$

We can now appeal to the observational soundness of our model to see that the following holds for any term $\mid x : \mathbf{com} \vdash_S M : \mathsf{P}\mathbf{com}$:

$$- \mid - \vdash_S M[\mathsf{skip}/x] : \mathsf{P}\mathbf{com}\Downarrow_S \Leftrightarrow - \mid - \vdash_S M[\Omega/x] : \mathsf{P}\mathbf{com}\Downarrow_S.$$

Now suppose we have some bireflective semantics $[\![-]\!]^\dagger$ and suppose that there exists a term $\mid x : \mathbf{com} \vdash_S U : \mathsf{P}\mathbf{com}$ with the semantics of the unit of the adjunction $J \dashv \mathsf{P}$:

$$[\![- \mid x : \mathbf{com} \vdash_S U : \mathsf{P}\mathbf{com}]\!]^\dagger = \eta^P_{\mathbf{com}}.$$

In a bireflective semantics we have $\eta^P_{\mathbf{com}};\varepsilon^P_{\mathbf{com}} = \mathbf{id}_{\mathbf{com}}$ hence:

$$[\![- \mid - \vdash_S \mathsf{derelict}\, U[\mathsf{skip}/x] : \mathbf{com}]\!]^\dagger = [\![- \mid - \vdash_S \mathsf{skip}]\!]^\dagger$$

and

$$[\![- \mid - \vdash_S \mathsf{derelict}\, U[\Omega/x] : \mathbf{com}]\!]^\dagger = [\![- \mid - \vdash_S \Omega]\!]^\dagger$$

and hence $[\![- \mid - \vdash_S \mathsf{skip}]\!]^\dagger = [\![- \mid - \vdash_S \Omega]\!]^\dagger$ and the semantics is inadequate.

7.3 Simple Arenas

Fortunately the interpretation of judgements in our model requires only a particular kind of SCI arena for which we can prove some special properties which we require for our definability proofs.

Definition 296 (Simple Arenas) We say that an arena A is simple if and only if it satisfies the following:

- $\sim_{\mathsf{A}}$ is an equivalence relation.
- A has a finite number of $\sim$ equivalence classes.
- For all moves $\mathsf{m}, \mathsf{m}', \mathsf{m}'' \in \mathsf{M_A}$ it follows that if $\mathsf{m}, \mathsf{m}' \vdash \mathsf{m}''$ then $\mathsf{m} \sim \mathsf{m}'$.
- For all moves $\mathsf{m}, \mathsf{m}', \mathsf{m}''$ in $\mathsf{M_A}$ it follows that if $\mathsf{m} \sim \mathsf{m}'$ and $\mathsf{m} \vdash \mathsf{m}''$ then $\mathsf{m}' \vdash \mathsf{m}''$.

Given a question $\mathsf{m} \in \mathsf{M_A}$ we write $[\mathsf{m}]$ for the $\sim$ equivalence class of moves containing m.

It is simple to check by structural induction that for any type A in the language **PCF**, (respectively $\mathbf{SCI}_b$, **SCIR**) it follows that $[\![A]\!]_P$ (respectively $[\![A]\!]_B$, $[\![A]\!]_S$) is a simple arena. Furthermore, the semantics of a term judgement for **PCF**, (respectively $\mathbf{SCI}_b$, **SCIR**) is always a strategy for a simple arena.

Definition 297 (Signature) Given a simple arena A and odd-length sequences $\mathsf{s} \cdot \mathsf{m} \in \mathcal{H}_{\mathsf{A}}$ we define the signature of $\mathsf{s} \cdot \mathsf{m}$ as follows:

- If m is a question then the signature is a triple $\langle \mathsf{m}, V_j, V_a \rangle$ where

  1. $V_j = \lceil \varepsilon \rceil_{\prec}$ if m is initial.
  2. Otherwise $V_j = \lceil \mathsf{s}_{\leq \mathsf{j}} \rceil_{\prec}$ where $\mathsf{j} \curvearrowleft \mathsf{m}$.

  1. $V_a = \lceil \varepsilon \rceil_{\prec}$ if there is no answer a such that $\mathsf{a} \prec \mathsf{m}$.
  2. Otherwise $V_a = \lceil \mathsf{s}_{\leq \mathsf{a}} \rceil_{\prec}$ where a is the greatest answer such that $\mathsf{a} \prec \mathsf{m}$.

- If m is an answer to a question to a question justified by a j that enables opponent questions in $\sim$ equivalence classes $c_1, \ldots, c_k$ then the signature is a $k+2$-tuple $\langle \mathsf{m}, \lceil \mathsf{s}_{\leq \mathsf{j}} \rceil_{\prec}, V_1, \ldots, V_k \rangle$ where for each $1 \leq i \leq k$

  1. $V_i = \lceil \varepsilon \rceil_{\prec}$ if there is no answer a to a question $\mathsf{q} \in c_i$ such that $\mathsf{a} \prec \mathsf{m}$.
  2. Otherwise $V_a = \lceil \mathsf{s}_{\leq \mathsf{a}} \rceil_{\prec}$ where a is the greatest answer to a question $\mathsf{q} \in c_i$ such that $\mathsf{a} \prec \mathsf{m}$.

Lemma 298 Given a simple arena A and odd-length sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{H}_{\mathsf{A}}$ it follows that $\lceil \mathsf{s} \rceil_{\prec} = \lceil \mathsf{s}' \rceil_{\prec}$ if and only if the sequences have the same signature.

Proof The proof is a simple induction on the length of s. ■

Lemma 299 Given a simple arena A and a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ with distinct occurrences of opponent questions $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ that are either both initial or that share a justifier we have the following.

- $\mathsf{m} \overset{*}{\prec} \mathsf{m}' \Rightarrow \mathsf{m} \sim \mathsf{m}'$
- $\mathsf{m} \overset{*}{\prec} \mathsf{m}' \Rightarrow \lambda^{\alpha\pi}\mathsf{m} = \alpha$

Proof

We prove these statements by induction on the length of s.

Base Case If $\mathsf{s} = \varepsilon$ then the lemma is trivial.

Inductive Step Let $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}'$. We must have some move $\mathsf{k} \in \mathsf{s}$ such that $\mathsf{m} \overset{*}{\prec} \mathsf{k} \prec \mathsf{m}'$. As $\mathsf{m} \overset{*}{\prec} \mathsf{k}$ we cannot have $\mathsf{k} \curvearrowleft \mathsf{m}'$ therefore k is the answer to some active question q such that $\mathsf{q} \leftrightsquigarrow \mathsf{m}'$. By lemma 245 we have $\mathsf{q} \overset{*}{\prec\!\succ} \mathsf{m}$ and we have one of the following three cases.

1. If $\mathsf{q} = \mathsf{m}$ then our lemma is satisfied.
2. If $\mathsf{m} < \mathsf{q}$ we can apply the inductive hypothesis to $\mathsf{s}_{\leq \mathsf{q}}$ to yield both $\mathsf{q} \sim \mathsf{m}$ and m is active and, as $\sim$ is an equivalence relation, we also have $\mathsf{m} \sim \mathsf{m}'$.
3. If $\mathsf{q} < \mathsf{m}$ we can apply the inductive hypothesis to $\mathsf{s}_{\leq \mathsf{m}}$ to yield $\mathsf{q} \sim \mathsf{m}$ but this case is absurd as the nesting condition ensures that q is answered, by k, before m is played and this contradicts $\mathsf{m} \overset{*}{\prec} \mathsf{k}$.

■

Lemma 300 Given a simple arena A and a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ with moves $\mathsf{q}_P, \mathsf{m}, \mathsf{q}_O, \mathsf{q}'_O \in \mathsf{s}$ such that we have the following,

A1 $\mathsf{q}_P < \mathsf{m}$,

A2 q_O and q'_O are opponent questions, either both initial or justified by the same move, and are distinct occurrences,

A3 q_O is open in $\lceil \mathsf{s}_{\leq \mathsf{q}_P} \rceil$ and $\mathsf{q}'_O \in \lceil \mathsf{s}_{\leq \mathsf{m}} \rceil$,

A4 q_P is a player question either initial or justified by a move preceding q_O.

A5 There are no passive opponent moves in $\lceil \mathsf{s}_{\leq \mathsf{q}_P} \rceil$ strictly between q_O and q_P.

it follows that

$$\mathsf{q}_O \overset{*}{\prec} \mathsf{q}'_O \Leftrightarrow \mathsf{q}_P \overset{*}{\prec} \mathsf{m}.$$

Proof First we assume $\mathsf{q}_O \overset{*}{\prec} \mathsf{q}'_O$ and prove $\mathsf{q}_P \overset{*}{\prec} \mathsf{m}$. By lemma 299 we have $\mathsf{q}_O \sim \mathsf{q}'_O$ and q_O active.

The nesting condition ensures that q_O is answered before q'_O is played. Let this answer be a_P. We now have the following:

$$\begin{array}{lcll} q_O & \stackrel{*}{\prec} & q_P & \text{by lemma 246.} \\ q_P & \stackrel{*}{\prec} & a_P & \text{by lemma 248.} \\ a_P & \prec & q'_O & \text{by definition.} \\ q'_O & \stackrel{*}{\prec} & m & \text{by lemma 246.} \\ q_P & \stackrel{*}{\prec} & m & \text{by transitivity.} \end{array}$$

We now assume $q_P \stackrel{*}{\prec} m$ and prove $q_O \stackrel{*}{\prec} q'_O$ by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is trivial.

Inductive Step Let $s = s' \cdot m$.

case: If m is a player move then $q_P \stackrel{*}{\prec} m$ if and only if $q_P \stackrel{*}{\prec} m^-$ and we simply apply the inductive hypothesis to s' to yield $q_O \stackrel{*}{\prec} q'_O$.

case: If $m = q'_O$ then as $q_P \stackrel{*}{\prec} m$ there must be some player answer a''_P to some active question q''_O so that $q_P \stackrel{*}{\prec} a''_P \prec m$. If $q''_O = q_O$ then we apply lemma 244 to yield $q_O \stackrel{*}{\prec} a''_P$ and by transitivity $q_O \stackrel{*}{\prec} q'_O$. Otherwise we apply the inductive hypothesis to yield $q_O \stackrel{*}{\prec} q''_O$, hence $q_O \stackrel{*}{\prec} q'_O$.

case: Otherwise we must have a move j such that $j \curvearrowleft m$ and $q'_O < j$. As $q_P \stackrel{*}{\prec} m$ there must be some move k such that $q_P \stackrel{*}{\prec} k \prec m$. Either $k = j$ or else k must answer an active opponent question q''_O such that $j \curvearrowleft q''_O \curvearrowleft k$. Either way visibility ensures that we have $q'_O \in \lceil s_{\leq k} \rceil$ so we can apply the inductive hypothesis to $s_{\leq k}$. ■

7.4 Minimal Sequences

Definition 301 We say that a sequence $s \cdot m \in \mathcal{H}_A$ is *minimal* if and only if for all $m' \in s \cdot m$ we have $m' \stackrel{*}{\prec} m$. For the sake of completeness we say that ε is minimal.

Given a sequence $s \in \mathcal{H}_A$ we write $\mathbf{min}(s)$ for that subsequence of s containing exactly those moves from $\lceil s \rceil_{\prec}$.

Lemma 302 Given a sequence $s \in \mathcal{H}_A$ it follows that $\mathbf{min}(s) \in \mathcal{J}_A$.

Proof The sequence starts with an initial opponent question and alternation is respected by inspection of the definition of $\lceil - \rceil_{\prec}$. Every non-initial move has a justifier by lemma 245. ■

Lemma 303 Given a sequence $s \in \mathcal{H}_A$ and moves $m, m' \in \mathbf{min}(s)$ such that $m \in \lfloor s_{\leq m'} \rfloor$ it follows that the subsequence of moves from s that lie inclusively between m and m' and are also present in $\mathbf{min}(s)$ is either empty or starts and finishes with moves n and n' such that $n \in \lfloor s_{\leq n'} \rfloor$.

Proof Our proof is by induction on the number of moves ks such that $m \leq k < m'$.

Base Case For our base case we have $m = m'$ and the lemma is trivial.

Inductive Step

case: Suppose m' is an opponent move. It follows that $m \in \lfloor s_{\leq m'^-} \rfloor$. Suppose there are no moves inclusively between m and m'^- that are also found in $\mathbf{min}(s)$ the lemma becomes simple. Either

$\mathsf{m}' \notin \mathbf{min}(\mathsf{s})$ and our lemma is satisfied or else $\mathsf{m}' \in \mathbf{min}(\mathsf{s})$ and we trivially have $\mathsf{m}' \in \lfloor \mathsf{s}_{\leq \mathsf{m}'} \rfloor$. Alternatively, let n, k' be the least and greatest moves from $\mathbf{min}(\mathsf{s})$ lying inclusively between m and m'^{-}. By inductive hypothesis we know that $\mathsf{n} \in \lfloor \mathsf{s}_{\leq \mathsf{k}'} \rfloor$. If m' is not in $\mathbf{min}(\mathsf{s})$ then this suffices to prove the lemma. Otherwise we simply note that $\mathbf{min}(\mathsf{s})$ alternates by lemma 302 and k' must be a player move we then inspect the definition of $\lfloor - \rfloor$ to yield $\mathsf{n} \in \lfloor \mathsf{s}_{\leq \mathsf{m}'} \rfloor$.

case: Now suppose that m' is a player move. Let $\mathsf{j} \curvearrowleft \mathsf{m}'$ and by the definition of $\lfloor - \rfloor$ we know that $\mathsf{m} \in \lfloor \mathsf{s}_{\leq \mathsf{j}} \rfloor$.

Let us consider the case when $\mathsf{m} \neq \mathsf{j}$. The number of moves between m and j is strictly less than between m and m' and so is the sequence strictly between j and m' so we can apply the inductive hypothesis twice. It is simple to show that the first move in $\mathbf{min}(\mathsf{s})$, if it exists, that lies inclusively between j and m' is an opponent move so it is simple to prove the lemma by inspection of the definition of $\lfloor - \rfloor$.

Finally we consider the case when $\mathsf{m} \neq \mathsf{j}$. Suppose $\mathsf{m}' \in \mathbf{min}(\mathsf{s})$ then by lemma 245 we also have $\mathsf{j} \in \mathbf{min}(\mathsf{s})$ and it follows from the definition of $\lfloor - \rfloor$ that $\mathsf{j} \in \lfloor \mathsf{s}_{\leq \mathsf{m}'} \rfloor$. If $\mathsf{m}' \notin \mathbf{min}(\mathsf{s})$ then by the visibility condition we know that $\lceil \mathsf{s} \rceil$ is of the form:

$$\cdots \mathsf{j} \cdot p_0 \overset{\frown}{\cdot} o_0 \cdots p_n \overset{\frown}{\cdot} o_n \cdot \mathsf{m}'.$$

If for some $0 \leq i \leq n$ we have the move $o_i \in \mathbf{min}(\mathsf{s})$ then we should note that $\mathsf{j} \in \mathbf{min}(\mathsf{s})$ and also $p_j, o_j \in \mathbf{min}(\mathsf{s})$ for all $j \leq i$ by lemma 246. We can then apply the inductive hypothesis to the sequence of moves inclusively between o_i and m' and the lemma follows simply. Alternatively if there is no move $o_i \in \mathbf{min}(\mathsf{s})$ we note that by the inductive hypothesis it follows that between each such pair p_i, o_i there are either no moves from $\mathbf{min}(\mathsf{s})$ or some sequence beginning and ending with moves n_i and n'_i such that $\mathsf{n}_i \in \lfloor \mathsf{s}_{\leq \mathsf{n}'_i} \rfloor$. It is simple to show that n_i must be an opponent move and n_i must be a player move. An inner induction on n shows that for each i, j such that $0 \leq i \leq j \leq n$ we have $\mathsf{n}_i \in \lfloor \mathsf{s}_{\leq \mathsf{n}'_j} \rfloor$. Hence our lemma holds. ■

Lemma 304 Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ and any move $\mathsf{m} \in \mathsf{s}$ it follows that $\lceil \mathsf{s}_{\leq \mathsf{m}} \rceil = \lceil \mathbf{min}(\mathsf{s}_{\leq \mathsf{m}}) \rceil$.

Proof The proof follows simply from lemma 245 and the definition of $\lceil - \rceil$. ■

Lemma 305 Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ and moves $\mathsf{m}, \mathsf{m}' \in \lceil \mathsf{s} \rceil_{\prec}$ it follows that:

- $\mathsf{m} \prec_{\mathsf{s}} \mathsf{m}' \Leftrightarrow \mathsf{m} \prec_{\mathbf{min}(\mathsf{s})} \mathsf{m}'$
- $\mathsf{m} \overset{*}{\prec}_{\mathsf{s}} \mathsf{m}' \Leftrightarrow \mathsf{m} \overset{*}{\prec}_{\mathbf{min}(\mathsf{s})} \mathsf{m}'$.

Proof The first item follows directly from the definitions of $\prec$ and $\mathbf{min}(-)$. The second item is proved using an induction on the number of moves between m and m' and uses the first item. ■

Lemma 306 Given a single threaded sequence $\mathsf{s} \in I_{\mathsf{A}}$ it follows that $\mathbf{min}(\mathsf{s}) \in I_{\mathsf{A}}$.

Proof We know from lemma 302 that $\mathbf{min}(\mathsf{s}) \in \mathcal{J}_{\mathsf{A}}$.

We will now prove that $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$ by induction on the length of $\mathbf{min}(\mathsf{s})$. Let $\mathsf{A} = \mathsf{A}_{\mathsf{l}} \to \mathsf{A}_{\mathsf{r}}$.

Base Case If $\mathbf{min}(\mathsf{s}) = \varepsilon$ then $\mathbf{min}(\mathsf{s}) \in \mathcal{S}_{\mathsf{A}}$.

Inductive Step Suppose $\mathbf{min}(\mathsf{s}) = \mathsf{s}' \cdot \mathsf{m}$.

case: If m is a player move then we apply the inductive hypothesis to s' and hence $\mathsf{s}' \cdot \mathsf{m}$ respects the switching condition.

case: Now suppose m is an opponent move. Let $\mathsf{m} \in \mathsf{s} \upharpoonright \mathsf{A}_r$ as the proof is similar when $\mathsf{m} \in \mathsf{s} \upharpoonright \mathsf{A}_l$. If m is initial then it must be the first move in the sequence as $\mathsf{s}' \cdot \mathsf{m}$ is single threaded and hence $\mathsf{s} \in \mathcal{S}_{\mathsf{A}}$. Otherwise let $\mathsf{j} \curvearrowleft \mathsf{m}$. We know by lemma 303 that $\mathsf{j} \in \lfloor \mathbf{min}(\mathsf{s}_{\leq \mathsf{m}}) \rfloor$ so we inspect the form of $\lfloor \mathbf{min}(\mathsf{s}_{\leq \mathsf{m}}) \rfloor$:

$$\cdots \mathsf{j} \cdot o_0 \curvearrowleft p_0 \cdots o_n \curvearrowleft p_n \cdot \mathsf{m}.$$

By inductive hypothesis we know that each pair of moves o_i, p_i must be from the $\mathsf{s} \upharpoonright \mathsf{A}_r$ and hence s respects the switching condition.

To show that $\mathsf{s} \in \mathcal{V}_{\mathsf{A}}$ we simply appeal to lemmas 303 and 304.

To see that s is well bracketed we note that by lemmas 245 and 250 it follows that any question and answer in s are either both included in or both absent from $\mathbf{min}(\mathsf{s})$.

We can use similar reasoning to show that $\mathbf{min}(\mathsf{s})$ respects the nesting condition.

The activity condition is obviously satisfied as $\mathbf{min}(\mathsf{s})$ is a subsequence of s.

To show that $\mathbf{min}(\mathsf{s})$ respects the SCI condition we first prove the following two statements. Given a sequence $\mathsf{s} \in \mathcal{H}_{\mathsf{A}}$ and moves $\mathsf{m}, \mathsf{m}' \in \lceil \mathsf{s} \rceil_{\prec}$ it follows that:

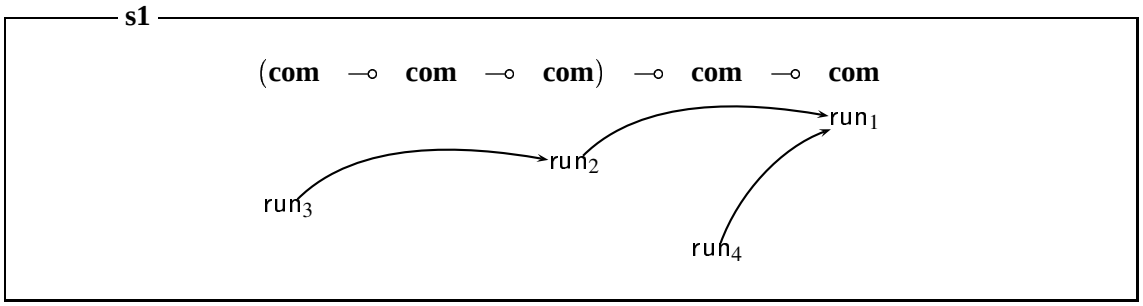
- $\mathsf{m} \prec_{\mathsf{s}} \mathsf{m}' \Leftrightarrow \mathsf{m} \prec_{\mathbf{min}(\mathsf{s})} \mathsf{m}'$
- $\mathsf{m} \overset{*}{\prec}_{\mathsf{s}} \mathsf{m}' \Leftrightarrow \mathsf{m} \overset{*}{\prec}_{\mathbf{min}(\mathsf{s})} \mathsf{m}'$.

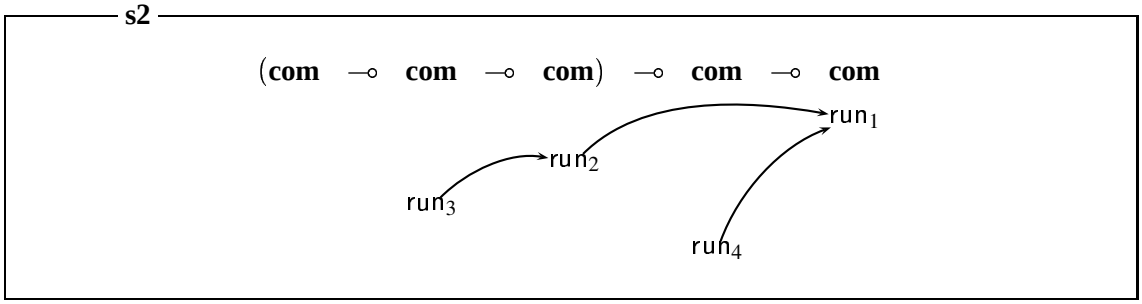
The first item follows directly from the definitions of $\prec$ and $\mathbf{min}(-)$. The second item is proved using an induction on the number of moves between m and m' and uses the first item. ■

As a corollary it follows from SCI innocence that any SCI strategy containing a single threaded sequence s must also contain the sequence $\mathbf{min}(\mathsf{s})$.

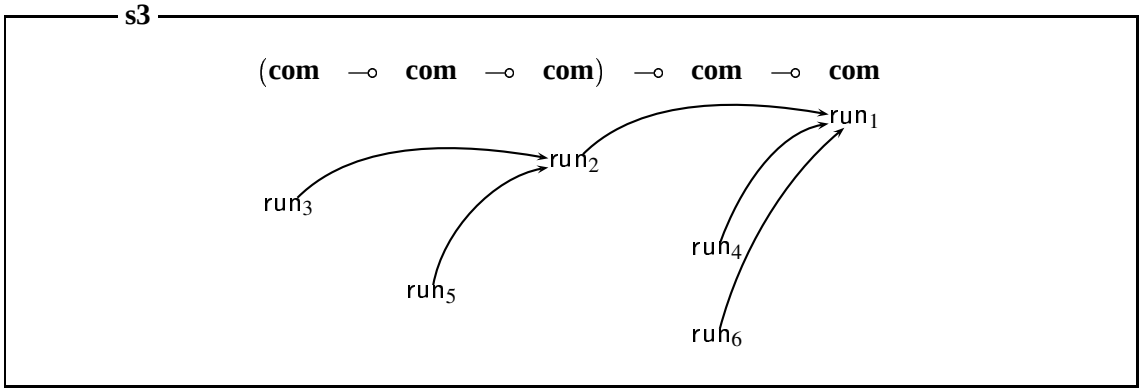
7.5 Complete Strategies

We do not have the usual definability result when we model our interference controlled languages in **C**. Consider a strategy whose single threaded plays are all prefixes of the following pair of plays.





Such a strategy is present in **C** but corresponds, intuitively, to an untypeable term of the form $\lambda f.\lambda x.f(x;\Omega)(x;\Omega)$. One possible way of circumventing this problem might be to construct a model in which we allow opponent to violate visibility. In this case the coherence condition would ensure that the presence of the above plays in a strategy would imply the inclusion of the following play, which would break the nesting condition, but the correct way to proceed in formulating such a model is unclear.



Definition 307 We say that a sequence $s \in \mathcal{J}_A$ is complete if and only if every question in s is answered. Given an SCI strategy $\sigma : A$ we say that a sequence $s \in \sigma$ is completable if and only if there exists some complete sequence $s' \in \sigma$ such that $s \sqsubseteq s'$. We say the strategy σ is complete if and only if every sequence in σ is completable.

Happily, it is the case that any pair of terms with semantics which coincide on complete sequences are observationally equivalent. This indicates three possible paths that we could bear fruit:

- We could collapse our model and work in a quotient of **C** in which strategies are equated if they possess exactly the same complete plays. However, we should note that there is further work to show that the collapsed category has the structure that we have demonstrated is possessed by **C**.
- We could modify our definition of the category so that strategies only contain sequences that are prefixes of complete sequences in the strategy and composition is performed as in **C** but is followed by an operation that discards sequences that are not prefixes of complete plays in the composite strategy. However, we should note that as our definitions stand, the result of discarding such uncompletable sequences from an SCI strategy is not guaranteed to result in an SCI strategy.
- We could weaken the definability result and show that for any suitable strategy σ we can define a term the semantics of which possess exactly the same complete plays as σ.

We choose the third approach and we show in section 7.10 that this approach suffices to yield full abstraction. When considering complete sequences, the following lemma affords us the intuition that given a strategy $\sigma : A \to B$ disjoint resources on the right of an arrow access disjoint active resources on the left.

Lemma 308 Given an SCI strategy $\sigma : A$ with single threaded sequences $q_O \cdot s, q'_O \cdot s' \in \sigma$ where $q_O \cdot s$ is complete, and initial player questions $q_P \in q_O \cdot s$ and $q'_P \in q'_O \cdot s'$, it follows that if q_P is active and $q_P \sim q'_P$ then it follows that $q_O \sim q'_O$.

Proof It is simple to show that $q_O \cdot s \cdot q'_O \cdot s' \in \mathcal{H}_A$ and by thread independence it follows that $q_O \cdot s \cdot q'_O \cdot s' \in \sigma$. By the SCI condition it must be the case that $q_P \overset{*}{\prec} q'_P$ and hence by lemma 246 $q_P \overset{*}{\prec} q'_O$ and therefore it must be the case that $q_O \sim q'_O$. ■

7.5.1 Active Visibility

The definition of active visibility that we will shortly provide is somewhat similar to the definitions of player rigidity, player bracketing and player visibility from chapter 4. As such it is not hard to prove that it is preserved by composition and is a property possessed by the copycat strategies. In earlier variations of our model it was required of any strategy in the model that its constituent sequences obeyed this condition. However we shall see that it is already implicit in the definitions of the strategies in which we are most interested. The activity condition might be regarded as a weakening of the activity condition in the fully abstract model of Idealized Algol with Passive Expressions found in [10]. The condition was proposed in its current form in personal communication with Guy McCusker. It formed part of an unpublished games model for **ILCR**, a language due to Yang and Reddy that utilises the elegant zones found in the **SCIR** type system to control statefulness but not interference [54].

Definition 309 (Active Player View) The active player view of a justified sequence s, notated $\lceil s \rceil_\alpha$, is that subsequence of s defined inductively as follows:

- $\lceil \varepsilon \rceil_\alpha = \varepsilon$.
- $\lceil t \cdot m \rceil_\alpha = \lceil t \rceil_\alpha \cdot m$ where $\lambda^{OP} m = P$.
- $\lceil t \cdot m \rceil_\alpha = m$ where m is an initial opponent move or $\lambda m = OQ\pi$.
- $\lceil t \cdot j \overset{\frown}{\cdot} t' \cdot m \rceil_\alpha = \lceil t \rceil_\alpha \cdot j \overset{\frown}{\cdot} m$ otherwise.

Definition 310 (Active Visibility) A justified sequence s satisfies active visibility if and only if, for all $t \cdot m \sqsubseteq s$ we have the justifier of m occurring in $\lceil t \rceil_\alpha$ if $\lambda^{OP\alpha\pi} m = P\alpha$ and m is not initial. Furthermore, if $\lambda^{OP\alpha\pi} m = P\alpha$ and m is initial then all opponent questions in $\lceil s_{\leq m} \rceil$ must be active.

We say that an SCI strategy $\sigma : A$ respects active visibility if and only if every sequence in σ respects active visibility.

Lemma 311 Given a complete SCI strategy σ for a simple arena A it follows that all completable plays in σ respect active visibility.

Proof The proof is by induction on the length of an arbitrary sequence $s \in \sigma$.

Base Case If $s = \varepsilon$ then the lemma is trivially satisfied.

Inductive Step We can apply the inductive hypothesis and need only check the case when the last move in s is an active player question. Let $s = t \cdot q_O \cdot t' \cdot q_P$ such that opponent question $q_O \in \lceil t \cdot q_O \cdot t' \cdot q_P \rceil$ where q_P is an active player question that is either initial or is justified by some move that lies before q_O. We will show that q_O is active. It suffices to consider only the case when $t \cdot q_O \cdot t' \cdot q_P$ is minimal as we can otherwise apply our inductive hypothesis to $\mathbf{min}(s)$ using lemmas 306 and 304.

Let us now consider an extension of s: a sequence $t \cdot q_O \cdot t' \cdot q_P \cdot u \cdot a_P \in \sigma$ which ends in a move answering q_O. We know that such a sequence exists as σ is a complete strategy. We now construct the following sequence:

$$t \cdot q_O \cdot t' \cdot q_P \cdot u \cdot a_P \cdot q_O^* \cdot t'^* \cdot q_P*$$

where the subsequence $q_O^* \cdot t'^* \cdot q_P^*$ is simply a copy of $q_O \cdot t' \cdot q_P$. We give justifiers to moves in this subsequence as follows. The move q_O^* receives the same justifier as q_O. Clearly this is in the opponent view. All opponent moves in t' are justified within t' by lemma 247 as s is minimal. It is then straightforward to show that the player view at every move in $q_O \cdot t' \cdot q_P$ is equal to the player view of its copy in $q_O^* \cdot t'^* \cdot q_P^*$. It is straightforward to show that the concatenated sequence is in $\mathcal{H}_A$ and also that for any move $m \in q_O \cdot t' \cdot q_P$. It is also straightforward to show that the SCI view at m is equal to the SCI view at the copy of m in $q_O^* \cdot t'^* \cdot q_P^*$. Therefore it follows that

$$t \cdot q_O \cdot t' \cdot q_P \cdot u \cdot a_P \cdot q_O^* \cdot t'^* \cdot q_P* \in \sigma.$$

By the bracketing condition it follows that q_P must have been answered by some move $a_O \in u$ and it must be the case that:

$a_O \overset{*}{\prec} q_P*$	by the SCI condition.
$q_P \overset{*}{\prec} q_P*$	by lemma 246
$q_O \overset{*}{\prec} q_P*$	by lemma 246
$q_O \overset{*}{\prec} q_O*$	by lemma 246
$\lambda^{\alpha\pi} q_O = \alpha$	by lemma 299

■

Lemma 312 Given a sequence $s \in \mathcal{H}_A$ that respects active visibility and moves $q, a, m \in s$ where:

- Player move a answers q.
- $q \leq m \leq a$.
- The move m is an active player question that is either initial or has a justifier that lies in $s_{\leq q}$.

Then it follows that $q \overset{*}{\prec} m \Rightarrow m \overset{*}{\prec} a$.

Proof We carry out induction on the length of $s_{\leq a}$.

Base Case If $s = \varepsilon$ then the proof is trivial.

Inductive Step We inspect $\lceil s_{\leq a} \rceil$, which by well bracketing and visibility has the following form:

$$\cdots q \cdot q_P \cdot a_O \cdots q_P \cdot a_O \cdot a.$$

If m is in this view then we know that $m \overset{*}{\prec} a$ by lemma 246. Otherwise it must lie between question and its answer that lie in this view. Let us call them q'_P and a'_O and remind ourselves of the situation graphically.

$$\cdots q \cdots q'_P \cdots m \cdots a'_O \cdots a$$

Now we note that any consecutive pair of moves, k and k^+ in $\lceil s_{\leq a} \rceil$, that lie inclusively between q and m have the property that $k \in \lceil s_{\leq m} \rceil \Rightarrow k^+ \in \lceil s_{\leq m} \rceil$, by lemma 240 when k is a player move and by the definition of player view when k is an opponent move. Note from lemma 247 that we must have $q \in \lceil s_{\leq m} \rceil$ and a simple induction on the length of $s_{\leq m}$ therefore yields $q'_P \in \lceil s_{\leq m} \rceil$.

We now inspect the move immediately following q'_P in $\lceil s_{\leq m} \rceil$. Clearly it is justified by q'_P and it must be a question as q'_P is answered after m. Let us call this move q_O. We know that s obeys active visibility so it must be that $\lambda q_O = OQ\alpha$. It is also unanswered in $s_{\leq m}$ by lemma 241. Because of well bracketing it must be the case that q_O is answered in $s_{\leq a}$, let this answer be a_P so we can apply our inductive hypothesis to $s_{\leq a_P}$ to yield $m \overset{*}{\prec} a_P$ and the following situation.

$$\cdots q \cdots q'_P \cdots q_O \cdots m \cdots a_P \cdots a'_O \cdots a$$

We have $a_P \prec a'_O$ by definition and we know, also, that $a'_O \overset{*}{\prec} a$ by lemma 246 so we have $m \overset{*}{\prec} a$ by transitivity. ■

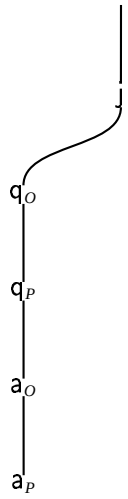
Lemma 313 Given an SCI strategy σ for a simple arena A and sequences $s, s' \in \sigma$ such that

- s is completable and $\lceil s \rceil$ is of the form $t \cdot j \cdot q_O \cdot u \cdot q_P$.
- $\lceil s' \rceil$ is of the form $t' \cdot j' \cdot q'_O \cdot u' \cdot q'_P$.
- $\lceil t \cdot j \rceil_\prec = \lceil t' \cdot j' \rceil_\prec$ rendered by a bijection f.
- $q_P \sim q'_P$.
- q_P is active.
- q_P and q'_P are either both initial or $n \curvearrowleft q_P \Leftrightarrow fn \curvearrowleft q'_P$.

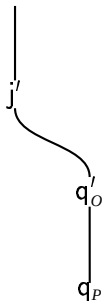
It must be the case that $q_O \sim q'_O$.

Proof The idea behind our proof is as follows. We first consider our sequence s then we consider

some extension of s from σ where both q_O and q_P are answered and then take the minimal sequence containing only those moves in the SCI view to result in a sequence which is also in σ. We consider the following graphic representation of the SCI view.

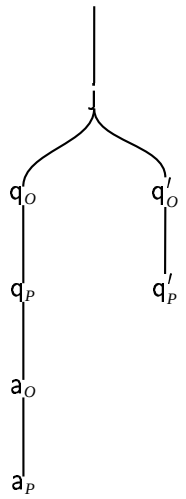


Next we consider the minimal sequence $\mathbf{min}(s')$ and find it to be of the following form:

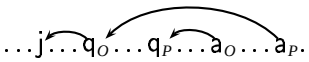


It is now possible to graft the second part of this sequence, starting immediately after j', onto the first. The resultant sequence must also be in σ. However if $q_O \not\sim q'_O$ the resultant sequence violates

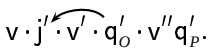
the SCI condition as $a_O \overset{*}{\not\prec} q'_P$:



We now show the proof more formally. First we take any extension $u \cdot a_P \in \sigma$ where a_P answers q_O. We know that such a sequence exists as we required that s be completable. From well-bracketing it follows that q_P must be answered by some move $a_O \in u \cdot a_P$. We know from lemma 311 that this sequence respects active visibility and by lemma 304 we know that $\mathbf{min}(u \cdot a_P)$ therefore respects active visibility also. By lemma 306 it follows that $\mathbf{min}(u \cdot a_P) \in \sigma$. By lemma 312 we know that $q_P \in \mathbf{min}(u \cdot a_P) \in \sigma$ and by lemma 246 we know that j and q_O are also in $q_P \in \mathbf{min}(u \cdot a_P)$ so, to recap, the sequence must be of the form:



We now consider the form of $\mathbf{min}(s')$ and find it to be of the form:



In a similar style to lemma 311 we can show that the sequence $u \cdot a_P \cdot v' q'_O \cdot v'' q'_P$ must also be in σ. By the SCI condition we require that $a_O \overset{*}{\prec} q'_P$ hence by lemma 246 and transitivity it follows that $q_O \overset{*}{\prec} q'_P$ and by lemma 246 again we have $q_O \overset{*}{\prec} q'_O$ and hence $q_O \sim q'_O$ by lemma 299. ■

7.6 Innocent SCI Strategies

Definition 314 We say that a sequence $s \in \mathcal{J}_A$ is single viewed if $s = \lceil s \rceil$. We write σ_{views} for the set of single viewed sequences in a strategy σ.

Lemma 315 Given an innocent SCI strategy $\sigma : A$ we have $s \in \sigma \Rightarrow \lceil s \rceil \in \sigma$.

Proof We know from lemma 252 that $\lceil s \rceil \in \mathcal{H}_A$ we carry out a simple induction on the length of s. ■

Definition 316 Given an innocent SCI strategy $\sigma : A$ we write σ_{comp} for the set of completable plays in σ.

Lemma 317 Given sequences $s, s' \in \mathcal{H}_A$ it follows that:

$$(\lceil s \rceil_{\prec} = \lceil s' \rceil_{\prec}) \Rightarrow (\lceil s \rceil = \lceil s' \rceil)$$

Proof This can be proved by a simple induction on the length of s. ■

As a corollary it follows that any innocent strategy $\sigma \subseteq I_A$ is an SCI strategy, or in other words σ respects SCI innocence.

Definition 318 Given a sequence $s \in I_A$ we say that an extension $s \cdot t \in I_A$ is a quick completion if and only if $s \cdot t$ is complete and there is no opponent question in t.

Lemma 319 Given an innocent SCI strategy $\sigma : A$ and a completable sequence $s \in \sigma$ it follows that there exists a quick completion of s in σ.

Proof Let $s \cdot t$ be some complete sequence in σ. We prove that s has a quick completion by induction on the length of t.

Base Case If $t = \varepsilon$ then s is a quick completion of itself.

Inductive Step

case: Suppose there is no open player question in $\lfloor s \rfloor$. It is simple to show that s is a quick completion of itself.

case: Let $s = s' \cdot q_P \cdot s''$, where q_P is the most recent unanswered question in $\lfloor s \rfloor$, and let $t = t' \cdot a_O \cdot m \cdot t''$ where a_O answers q_P in $s \cdot t$.

It is simple to show that $s' \cdot q_P \cdot s'' \cdot a_O \cdot m \cdot t'' \in \sigma$. We now apply the inductive hypothesis and assume some quick completion of $s' \cdot q_P \cdot s'' \cdot a_O \cdot m$. ■

Lemma 320 Given an innocent SCI strategy $\sigma : A$ it follows that the set

$$S = \{s \mid \exists t \in \sigma_{\mathsf{comp}}.s \sqsubseteq_{\mathsf{even}} t\}$$

is an innocent strategy for the arena A.

Proof Clearly the set is a prefix closed, deterministic subset of I_A. We need only check that the resultant strategy is innocent as by lemma 317 we know that innocence implies SCI innocence.

Suppose we take any sequences $s \cdot m \cdot n, s' \in S$ such that there exists an extension $s' \cdot m'$ for which $\lceil s \cdot m \rceil = \lceil s' \cdot m' \rceil$. We know from the innocence of σ that $s' \cdot m' \cdot n' \in \sigma$ such that $\lceil s \cdot m \cdot n \rceil = \lceil s' \cdot m' \cdot n' \rceil$ so it is sufficient for us to show that $s' \cdot m' \cdot n'$ is completable.

case: Suppose there is no open player question in $\lceil s' \cdot m' \cdot n' \rceil$. In this case it is simple to show that $s' \cdot m' \cdot n'$ is complete.

case: Suppose that the most recent open player question in $\lceil s' \cdot m' \cdot n' \rceil$ lies in the sequence s'. We know that there is some quick completion $s' \cdot t \cdot a_O \cdot t'$ of s' where a_O answers q_P by lemma 319. It is simple to show that $s' \cdot m' \cdot n' \cdot a_O \cdot t' \in \sigma$ and that this sequence is complete.

case: Finally suppose that the most recent open player question in $\lceil s' \cdot m' \cdot n' \rceil$ is the move n'. We know that there is some quick completion $s \cdot m \cdot n \cdot t \cdot t'$ of $s \cdot m \cdot n$ by lemma 319 such that:

- Every opponent move in t is an answer to some question in the sequence $\mathsf{n} \cdot \mathsf{t}$ and every question in $\mathsf{n} \cdot \mathsf{t}$ is answered.
- Every player question in $\mathsf{n} \cdot \mathsf{t}$ is answered in $\mathsf{n} \cdot \mathsf{t}$.
- The sequence $\mathsf{n} \cdot \mathsf{t}$ ends in a player answer.

It is simple to show that the sequence $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \cdot \mathsf{t} \in \sigma$. We know that the sequence s' is completable and hence by lemma 319 it has a quick completion $\mathsf{s}' \cdot \mathsf{u}$. It is now straightforward to show that $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \cdot \mathsf{t} \cdot \mathsf{u} \in \sigma$ and that it is a complete sequence. ■

Innocent strategies are completely determined by the single viewed sequences they contain. We show that the function $-_{\mathsf{views}}$ is injective by defining its inverse.

Definition 321 Given a set $S \subseteq I_{\mathsf{A}}$ of single viewed sequences we define the set $\mathsf{strategize}(S)$ as the least set such that:

- $\varepsilon \in \mathsf{strategize}(S)$.
- Given $\mathsf{s} \in \mathsf{strategize}(S)$ and a sequence $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in I_{\mathsf{A}}$ such that $\lceil \mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \rceil \in S$ then it follows that $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in \mathsf{strategize}(S)$.

Lemma 322 Given an SCI strategy $\sigma : \mathsf{A}$ we have

$$\mathsf{strategize}(\sigma_{\mathsf{views}}) = \sigma.$$

Proof First we show that for all $\mathsf{s} \in \mathsf{strategize}(\sigma_{\mathsf{views}})$ we have $\mathsf{s} \in \sigma$ by induction on the length of s.

Base Case We know that $\varepsilon \in \mathsf{strategize}(\sigma_{\mathsf{views}})$ and $\varepsilon \in \sigma$.

Inductive Step Let $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{n}$. From the definition of $\mathsf{strategize}(-)$ we must have $\mathsf{s}' \in \mathsf{strategize}(\sigma_{\mathsf{views}})$ and a sequence $\mathsf{t} \cdot \mathsf{m} \cdot \mathsf{n} \in \sigma_{\mathsf{views}}$ such that $\mathsf{s}' \cdot \mathsf{m}$ is in I_{A} and $\mathsf{t} \cdot \mathsf{m} = \mathsf{s}' \cdot \mathsf{m}$. By inductive hypothesis we have $\mathsf{s}' \in \sigma$ hence by the innocence of σ we have $\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{n} \in \sigma$.

We now show that for all $\mathsf{s} \in \sigma$ we have $\mathsf{s} \in \mathsf{strategize}(\sigma_{\mathsf{views}})$ by induction on the length of s.

Base Case Once again $\varepsilon \in \mathsf{strategize}(\sigma_{\mathsf{views}})$ and $\varepsilon \in \sigma$.

Inductive Step Let $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{n}$. By induction hypothesis we have $\mathsf{s}' \in \mathsf{strategize}(\sigma_{\mathsf{views}})$ and clearly we have $\mathsf{s}' \cdot \mathsf{m} \in I_{\mathsf{A}}$ and by lemma 315 $\lceil \mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{n} \rceil \in \sigma_{\mathsf{views}}$ hence by innocence we have $\mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{n} \in \mathsf{strategize}(\sigma_{\mathsf{views}})$. ■

The idea is that when applied to a certain kind of set S, it follows that the set $\mathsf{strategize}(S)$ is the least innocent SCI strategy containing S. We now examine some conditions that we can place on a set of single viewed sequences S so that $\mathsf{strategize}(S)$ is an innocent SCI strategy. Note that we do not claim that this is the weakest set of conditions that have this property.

Definition 323 (View Rules) Given a set of single viewed sequences $T \subseteq \mathcal{L}_{\mathsf{A}}$ we term the following conditions the view rules.

V1 $S \subseteq I_{\mathsf{A}}$.

V2 $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in S \Rightarrow \mathsf{s} \in S$.

V3 For all sequences $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{n}, \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \in S$ such that $\mathsf{s} \cdot \mathsf{m} = \mathsf{s}' \cdot \mathsf{m}'$ we have $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{n} = \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$.

V4 Each sequence $\mathsf{s} \in S$ respects active visibility.

V5 For all sequences $\mathsf{s} \cdot \mathsf{q}_O \cdot \mathsf{t} \cdot \mathsf{q}_P, \mathsf{s} \cdot \mathsf{q}'_O \cdot \mathsf{t}' \cdot \mathsf{q}'_P \in S$ where:

- q_O and q'_O are opponent questions.
- $\mathsf{q}_P \sim \mathsf{q}'_P$.
- q_P and q'_P are either both initial or both justified by the same move in s.
- q_P is active.

it follows that $\mathsf{q}_O \sim \mathsf{q}'_O$.

We can observe some similarity between the view rules **V1-5** and the thread rules **T1-6**. The first three view rules are the same as the first three thread rules. View rule **V4** might intuitively be seen as internalizing the activity condition and view rule **V5** internalizes thread rule **T5**.

Lemma 324 Given a set $S \subseteq \mathcal{L}_{\mathsf{A}}$ of even length single-viewed strategies such that S obeys the view rules **V1-6** then it follows that $\mathsf{strategize}(S)$ is an innocent SCI strategy for arena A.

Proof It is straightforward to show that $\mathsf{strategize}(S)$ is an even prefix closed subset of I_{A} from the definition of $\mathsf{strategize}(-)$. Determinism follows from view rule **V3**. We need now only prove innocence as by lemma 317 SCI innocence is implied. Suppose we have sequences $\mathsf{s} \operatorname{cot} \mathsf{m} \cdot \mathsf{n}, \mathsf{s}' \in \mathsf{strategize}(S)$ and an extension $\mathsf{s}' \cdot \mathsf{m}' \in I_{\mathsf{A}}$ such that $\lceil \mathsf{s}' \cdot \mathsf{m}' \rceil = \lceil \mathsf{s} \cdot \mathsf{m} \rceil$ then we must show that $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \in \mathsf{strategize}(S)$. Inspection of the definition of $\mathsf{strategize}(-)$ clearly shows that $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \in \mathsf{strategize}(S)$ if and only if $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \in I_{\mathsf{A}}$.

We know that $\mathsf{s}' \cdot \mathsf{m}' \in I_{\mathsf{A}}$ and also $\lceil \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \rceil = \lceil \mathsf{s} \cdot \mathsf{m} \cdot \mathsf{n} \rceil$ and therefore it is straightforward to show that $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ is in $\mathcal{H}_{\mathsf{A}}$ and respects the activity condition. We must simply check that $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ respects player nesting and the SCI condition.

case: Suppose n' is a player answer to a question q_O. Clearly $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ respects player nesting as $\mathsf{s}' \cdot \mathsf{m}' \in I_{\mathsf{A}}$. Suppose there is some move $\mathsf{k} \in \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ such that $\mathsf{k} \lhd \mathsf{n}'$. From the definition of $\lhd$ we know that we must have one of the following:

- $\mathsf{k} \curvearrowleft \mathsf{n}'$ and $\mathsf{k} \in \lceil \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}' \rceil$ by the visibility condition and hence $\mathsf{k} \overset{*}{\prec} \mathsf{n}'$ by lemma 246.
- There exists some moves $\mathsf{q}_P \in \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ such that $\mathsf{q}_O \curvearrowleft \mathsf{q}_P$ and k answers active player question q_P. It is simple to check that $\mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ obeys active visibility and it follows that as $\mathsf{q}_O \in \lceil \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'_{\leq \mathsf{q}_P} \rceil$ by the visibility condition it must also be the case that $\mathsf{q}_P \overset{*}{\prec} \mathsf{n}'$ by lemma 248 and hence $\mathsf{k} \overset{*}{\prec} \mathsf{n}'$ by lemma 250.

case: Now suppose n' is a player question either initial or justified by a move q_O. Suppose there is some move $\mathsf{k} \in \mathsf{s}' \cdot \mathsf{m}' \cdot \mathsf{n}'$ such that $\mathsf{k} \lhd \mathsf{n}'$. From the definition of $\lhd$ we know that we must have one of the following:

- $k \curvearrowleft n'$ and $k \in \lceil s' \cdot m' \cdot n' \rceil$ by the visibility condition and hence $k \overset{*}{\prec} n'$ by lemma 246.
- There exists some active player question $q_P \in s' \cdot m' \cdot n'$ such that $q_P \leftrightsquigarrow n'$ and k answers q_P. Let us now examine the player views $\lceil s' \cdot m' \cdot n'_{\leq q_P} \rceil$ and $\lceil s' \cdot m' \cdot n' \rceil$. These two views trivially have a longest prefix in common — albeit possibly an empty one. Furthermore, inspection of the definition of player view implies that this prefix does not end with an opponent move. Let us call this prefix t and let the move immediately after this prefix in $\lceil s' \cdot m' \cdot n'_{\leq q_P} \rceil$ be q_O — note that it must be a question or else it would also be in $\lceil s' \cdot m' \cdot n' \rceil$ by lemma 240. If the move immediately following t in $\lceil s' \cdot m' \cdot n' \rceil$ is an answer then lemma 248 ensures that $q_P \overset{*}{\prec} n'$ and hence $k \overset{*}{\prec} n'$ by lemma 250. Otherwise let q'_O be the move immediately following the prefix t in $\lceil s' \cdot m' \cdot n' \rceil$. By view rule **V5** we know that $q_O \sim q'_O$ and by active visibility we know that q_O is active. By the nesting condition we know that q_O is answered by some move *pa* before q'_O is played. We therefore have:

$$\begin{array}{rcl} q_O & \overset{*}{\prec} & q_P \text{ by lemma 244} \\ q_P & \overset{*}{\prec} & k \text{ by lemma 244} \\ q_O & \overset{*}{\prec} & k \text{ by transitivity} \\ k & \overset{*}{\prec} & a_P \text{ by lemma 248} \\ a_P & \prec & q'_O \text{ by definition} \\ q'_O & \overset{*}{\prec} & n' \text{ by lemma 246} \\ k & \overset{*}{\prec} & n' \text{ by transitivity} \end{array}$$

Finally we are left to show that the nesting condition is obeyed. Our proof is somewhat similar to our proof of the SCI condition. Suppose that there exists some active player question $q_P \in s' \cdot m' \cdot n'$ such that $q_P \leftrightsquigarrow n'$. Again we examine the player views $\lceil s' \cdot m' \cdot n'_{\leq q_P} \rceil$ and $\lceil s' \cdot m' \cdot n' \rceil$ and let t be the greatest prefix common to both views and let the move immediately after this prefix in $\lceil s' \cdot m' \cdot n'_{\leq q_P} \rceil$ be q_O. If the move immediately following t in $\lceil s' \cdot m' \cdot n' \rceil$ is an answer then well-bracketing assures us that q_P has been answered. Otherwise let q'_O be the move immediately following the prefix t in $\lceil s' \cdot m' \cdot n' \rceil$. By view rule **V5** we know that $q_O \sim q'_O$ and by active visibility we know that q_O is active. By the nesting condition applied to $s' \cdot m$ we know that q_O is answered by some move *pa* before q'_O is played and by well-bracketing it follows that q_P is also answered.

■

Lemma 325 Given a complete innocent strategy σ for a simple arena A it follows that σ_{views} obeys view rules **V1-6**.

Proof

V1 $\sigma_{\mathsf{views}} \subseteq I_A$ by lemma 252.

V2 $s \cdot m \cdot m' \in \sigma_{\mathsf{views}} \Rightarrow s \in S$ by the prefix closure of σ.

V3 For all sequences $s \cdot m \cdot n, s' \cdot m' \cdot n' \in \sigma_{\mathsf{views}}$ such that $s \cdot m = s' \cdot m'$ we have $s \cdot m \cdot n = s' \cdot m' \cdot n'$ by the determinism of σ.

V4 Each sequence $s \in \sigma_{\mathsf{views}}$ respects active visibility by lemma 311.

V5 For all sequences $s \cdot q_O \cdot t \cdot q_P, s \cdot q'_O \cdot t' \cdot q'_P \in \sigma_{\mathsf{views}}$ where:

- q_O and q'_O are opponent questions.
- $\mathsf{q}_P \sim \mathsf{q}'_P$.
- q_P and q'_P are either both initial or both justified by the same move in s.
- q_P is active.

it follows that $\mathsf{q}_O \sim \mathsf{q}'_O$ by lemma 313.

■

7.7 Innocent Definability

Our definability result will be for *compact* strategies only — we will later show that this suffices to yield full abstraction. SCI-strategies for an arena A ordered by inclusion form a DCPO. The compact elements in this order are those in which player makes a response to a finite number of SCI-views of single threads. The innocent strategies also form a DCPO and an innocent strategy is innocently compact if player only makes a response to a finite number of player-views.

In [28, 9] it is noted that the definability result does not quite hold for the language **PCF**; the language must be augmented with k-ary conditional statements. This addition does not affect the full abstraction result as this is a *conservative extension* of **PCF**: observationally equivalent terms of **PCF** remain equivalent when contexts are permitted to contain such case constructs. Accordingly our proofs also make use of this family of language constructs that are not present in the languages **PCF**, $\mathbf{SCI}_b$ or **SCIR**. We will add k-ary conditional statements with the following syntax:

$$\mathsf{case}_k\ M\ \mathsf{of}\ N_1 \ldots\ N_k,$$

with the understanding that each of the subterms N_i are of the same base type and M is of natural number type. The idea of the operational semantics of such a statement is that M is first evaluated to a natural number and then the program branches depending on this value.

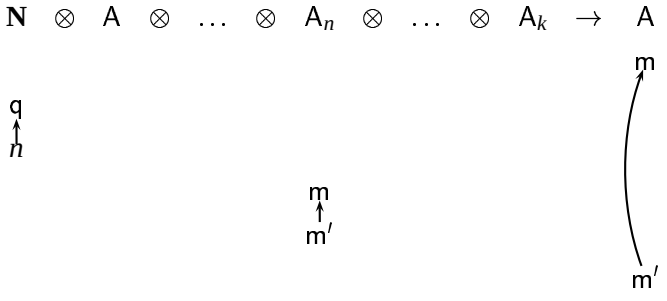
The typing rule for the construct is as follows:

$$\frac{\Gamma \vdash M : \mathbf{N} \quad \Gamma \vdash N_1 : \tau \quad \ldots \quad \Gamma \vdash N_k : \tau}{\Gamma \vdash \mathsf{case}_k\ M\ \mathsf{of}\ N_1 \ldots\ N_k : \tau}$$

The operational semantics for the construct is as follows:

$$\frac{s, M \Downarrow \mathsf{s}', i \qquad s', M_{i+1} \Downarrow s'', V}{s, \mathsf{case}_k\ M\ \mathsf{of}\ N_1 \ldots\ N_k \Downarrow s'', V}\ i \in \{0, \ldots, k-1\}$$

For each flat arena A and for each natural number k we define the strategy $\mathbf{case} : \mathbf{N} \otimes \mathsf{A}_1 \ldots \otimes \mathsf{A}_k \to \mathsf{A}$ which has complete, single threaded plays of the following form:



We should note that adding k-ary conditional statements to the languages $\mathbf{SCI}_b$ and **SCIR** is not problematic as k-ary conditional statements are definable in these languages. We can view such syntax as sugar for terms using local store and nested binary conditionals. The syntax for terms using the k-ary syntax can be desugared inductively on k. Consider the term

$$\mathsf{case}_k\ M\ \mathsf{of}\ N_1 \ldots\ N_k$$

where each of the N_i are of **com** type: when $k = 0$ we define $\mathsf{case}_k\ M\ \mathsf{of} -$ to be the term Ω and when $k = k' + 1$ we define

$$\mathsf{case}_k\ M\ \mathsf{of}\ N_1 \ldots\ N_k$$

to be the term

$$\mathsf{new}\ x := 0\ \mathsf{in}\ (x := M; \mathsf{if}\ x = 0\ \mathsf{then}\ N_1\ \mathsf{else}\ (\mathsf{case}_k\ (\mathsf{pred}\ x)\ N_2 \ldots\ N_k))$$

We begin our definability proofs by showing that the innocently compact strategies consisting of complete plays are definable.

Proposition 326

- Given a complete, innocent SCI-strategy, with finite view function,

  $$\sigma : [\![A_1]\!]_B \otimes \ldots \otimes [\![A_n]\!]_B \to [\![A]\!]_B$$

  such that A and each of the A_i are $\mathbf{SCI}_b$ types, we can define a term

  $$x_1 : A_1 \ldots x_n : A_n \vdash_B M : A$$

  with semantics σ that does not use of the new and do constructs.

- Given a complete, innocent SCI-strategy, with finite view function,

  $$\sigma : \mathsf{S}([\![A_1]\!]_S \otimes \ldots \otimes [\![A_n]\!]_S) \otimes [\![A_{n+1}]\!]_S \otimes \ldots \otimes [\![A_m]\!]_S \to [\![A]\!]_S$$

  such that A and each of the A_i are **SCIR** types, we can define a term

  $$x_1 : A_1 \ldots x_n : A_n | x_{n+1} : A_{n+1} \ldots x_m : A_m \vdash_S M : A$$

  with semantics σ.

- Given an innocent SCI-strategy, with a finite view function,

  $$\sigma : [\![A_1]\!]_P \otimes \ldots \otimes [\![A_n]\!]_P \to [\![A]\!]_P$$

  such that A and each of the A_i are **PCF** types, we can define a term

  $$x_1 : A_1 \ldots x_n : A_n \vdash_P M : A$$

  of the language **PCF**+case with semantics σ.

In this section we only prove the second of these statements as the proofs of the other two statements are similar and somewhat simpler. Proof of the third statement has already been alluded to in chapter 4 and follows the proofs given in [7, 36, 2] and does not rely on the completeness of the strategy σ. We should note here that we do not explicitly use the new and do constructs except in desugaring the case statements.

Definition 327 (Singly answered/Multiply answered) Given an arena A we say that a question $\mathsf{q} \in \mathsf{M_A}$ is singly answered if and only if for any answers $\mathsf{a}, \mathsf{a}' \in \mathsf{M_A}$ such that $\mathsf{q} \vdash \mathsf{a}, \mathsf{a}'$ we have $\mathsf{a} = \mathsf{a}'$. Otherwise we say that q is multiply answered. In the arenas that interpret **SCIR** judgements we will have run and write_n as singly answered questions and q as the only multiply answered questions.

Definition 328 (Linear Applicative Strategies) Given an innocent SCI-strategy

$$\sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \to [\![\tau]\!]_S$$

such that τ is an **SCIR** base type and Γ, Δ are **SCIR** contexts, we say that σ is linear applicative in $A \in \Gamma, \Delta$ if and only if for all sequences $\mathsf{s} \cdot \mathsf{m} \cdot \mathsf{m}' \in \sigma$ it follows that:

1. m is an initial move in $[\![\tau]\!]_S$ if and only if m' is an initial move in $[\![A]\!]_S$.
2. m' is an answer to an initial move in $[\![\tau]\!]_S$ if and only if m is an answer to an initial move in $[\![A]\!]_S$.

In other words the strategy only asks an initial question in A immediately after asking an initial question in τ, and only answers it immediately before answering the initial move in τ.

Lemma 329 Suppose we are given an **SCIR** type A and **SCIR** contexts Γ, Δ and a complete, innocent SCI strategy

$$\sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![A]\!]_S \to [\![\mathbf{N}]\!]_S$$

with finite view function that is linear applicative in A, where the initial move in $[\![A]\!]_S$ is a many answered q move.

Under the assumption that for any innocent strategy

$$\sigma' : \mathsf{S}[\![\Gamma']\!]_S \otimes [\![\Delta']\!]_S \to [\![A']\!]_S,$$

where σ' has a strictly smaller view function that σ, there exists some term $\Gamma' | \Delta' \vdash_S M : A'$ such that

$$[\![\Gamma' | \Delta' \vdash_S M : A']\!]_S = \sigma' : \mathsf{S}[\![\Gamma']\!]_S \otimes [\![\Delta']\!]_S \to [\![A']\!]_S$$

it follows that there exists some term $\Gamma | \Delta, x : A \vdash N : \mathbf{N}$ such that

$$[\![\Gamma | \Delta, x : A \vdash N : \tau]\!]_S = \sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes A \to [\![\mathbf{N}]\!]_S.$$

Proof Our proof is by structural induction on the type A.

Base Case Suppose A is a base type. By assumption it must be the case that either $A = \mathbf{N}$ of $A = \mathbf{var}$. In either case every single threaded sequence in σ is either of the form $\mathsf{q} \cdot \mathsf{q}'$ or of the form:

$$\mathsf{q} \cdot \mathsf{q}' \cdot n' \cdot n$$

where n' and n are natural numbers. By the innocence of σ there must be some function $f : \mathbb{N} \mapsto \mathbb{N} \cup \{\Omega\}$ defined as follows:

$$\begin{aligned} fn &= n' \text{ if there exists a sequence } \mathsf{q} \cdot \mathsf{q}' \cdot \mathsf{n} \cdot \mathsf{n}' \in \sigma \\ fn &= \Omega \text{ otherwise.} \end{aligned}$$

As σ has a finite view function there must be some k such that for any n it follows that $n > k \Rightarrow fn = \Omega$. Suppose $A = \mathbf{N}$, we can then construct a term with semantics σ as follows:

$$\Gamma|\Delta, x : A \vdash_S \mathsf{case}_k\, x \,\mathsf{of}\; f0 \ldots\; fk.$$

Similarly, if $A = \mathbf{var}$, we can then construct a term with semantics σ as follows:

$$\Gamma|\Delta, x : A \vdash_S \mathsf{case}_k\, !x \,\mathsf{of}\; f0 \ldots\; fk.$$

Inductive Step

case: Let $A = PB$. First we show that the set σ can be transformed into an innocent strategy for the game $\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{N}]\!]_S$ that is linear applicative in B. By lemma 325 it follows that σ_{views} obeys view rules **V1-5**. Clearly the set σ_{views} can also be considered as a set of sequences of moves from the arena $\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{N}]\!]_S$: by original assumption our initial question in $[\![PB]\!]_S$ is many answered and hence passive when considered as an initial move in $[\![B]\!]_S$. We verify that the set σ_{views} still obeys view rules **V1-5** when considered as a set of sequences for the altered arena. We can now apply lemma 324 to give an innocent strategy $\mathsf{strategize}(\sigma_{\mathsf{views}}) : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{N}]\!]_S$ and inspection confirms that the strategy is linear applicative.

By inductive hypothesis we have some term $\Gamma|\Delta, x : B \vdash_S N : \tau$ such that

$$[\![\Gamma \mid \Delta, x : B \vdash N : \mathbf{N}]\!]_S = \mathsf{strategize}(\sigma_{\mathsf{views}}) : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes B \to \mathbf{N}.$$

It is straightforward to show that

$$\sigma = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S} \otimes \mathbf{der}_B); \mathsf{strategize}(\sigma_{\mathsf{views}})$$

and hence we have

$$[\![\Gamma \mid \Delta, x : A \vdash_S N[\mathbf{der}x/x] : \tau]\!]_S = \sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes A \to \mathbf{N}.$$

case: Let $A = B \times C$. Note that as σ is linear applicative, only one initial move is ever played in $[\![B \times C]\!]_S$, so let us assume that this is a move from $[\![B]\!]_S$ (our proof is similar when it is from $[\![C]\!]_S$). Once again we can transform the set σ into an innocent strategy for the arena $\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{N}]\!]_S$ that is linear applicative strategy in $[\![B]\!]_S$. By lemma 325 we have σ_{views} obeys view rules **V1-5**. Clearly the set σ_{views} can be transformed into a set of sequences, σ'_{views}, of moves from the arena $\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{N}]\!]_S$: the only difference being the lack of tagging of moves that was required to form a disjoint union to construct the product type. It is trivial to show that the set σ'_{views} obeys view rules **V1-5**. We can now apply lemma 324 to give an innocent strategy $\mathsf{strategize}(\sigma'_{\mathsf{views}}) : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{N}]\!]_S$ and inspection confirms that the strategy is linear applicative.

By inductive hypothesis we have some term $\Gamma \mid \Delta, x : B \vdash_S N : \mathbf{N}$ such that

$$[\![\Gamma \mid \Delta, x : B \vdash_S N : \mathbf{N}]\!]_S = \mathsf{strategize}(\sigma'_{\mathsf{views}}) : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes B \to \mathbf{N}.$$

It is straightforward to show that

$$\sigma = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S} \otimes \pi_1); \mathsf{strategize}(\sigma'_{\mathsf{views}})$$

and hence we have

$$[\![\Gamma \mid \Delta, x : A \vdash_S N[\pi_1(x)/x] : \mathbf{N}]\!]_S = \sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes A \to \mathbf{N}.$$

case: The case when $A = B \multimap C$ requires more thought. First we note that no active initial move is played from $[\![\Delta]\!]_S$ so it is straightforward to transform our strategy σ into a strategy:

$$\sigma' : \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta]\!]_S \otimes \mathsf{S}[\![A]\!]_S \to [\![\mathbf{N}]\!]_S$$

containing the exactly the same sequences.

Let T be that subset of σ'_{views} containing exactly the sequences of the form

$$\mathsf{q} \overset{\curvearrowleft}{\cdot} \mathsf{q}' \overset{\curvearrowright}{\cdot} \mathsf{m} \cdot \mathsf{s}$$

where m is a move from $[\![B]\!]_S$. Let S be the complement, $\sigma_{\mathsf{views}} \setminus T$. Note that the visibility condition ensures that no sequence in S contains a move from $[\![B]\!]_S$.

We now transform σ' into a strategy:

$$\sigma'' : \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta]\!]_S \otimes \mathsf{S}[\![\Delta]\!]_S \otimes \mathsf{S}[\![A]\!]_S \to [\![\mathbf{N}]\!]_S$$

such that sequences in T are unchanged but sequences in S use moves from the copied contexts, $[\![\Gamma]\!]'_S$ and $[\![\Delta]\!]'_S$. We are going to define a term with semantics σ'' and it is straightforward to see how it is then possible to subsequently define a term with semantics σ from σ' using contraction and activation rules.

It is straightforward to check that T obeys view rules**V1-5** furthermore, the definition of T assures us that no move from $[\![B]\!]_S$ is played in any sequence in this strategy so we can show that the following strategy is well defined:

$$\mathsf{strategize}(T) : \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta]\!]_S \otimes \mathsf{S}[\![C]\!]_S \to [\![\mathbf{N}]\!]_S$$

and by inductive hypothesis we have some term N such that

$$[\![\Gamma, \Delta, x : C \mid - \vdash_S N : \mathrm{exp}]\!]_S = \mathsf{strategize}(T).$$

Now we consider the set S and recall that it contains only sequences of the form

$$\mathsf{q} \overset{\curvearrowleft}{\cdot} \mathsf{q}' \overset{\curvearrowright}{\cdot} \mathsf{m} \cdot \mathsf{s}$$

where m is a move from $[\![B]\!]_S$. As σ obeys view rules **V1-5** it is straightforward to show that S obeys view rules **V1** and **V3-5**, but not **V2**: S is not even-length prefix closed as each sequence commences with two questions. However it is straightforward to show that removing the first two moves from each sequence in S in conjunction with some re-tagging leaves us with a set S' of sequences of moves from the game $\mathsf{S}[\![\Gamma']\!]_S \otimes \mathsf{S}[\![\Delta']\!]_S \to [\![B]\!]_S$. Furthermore, it is straightforward to show that S' obeys view rules **V1-5** and hence we can construct the innocent strategy

$$\mathsf{strategize}(S') : \mathsf{S}[\![\Gamma']\!]_S \otimes \mathsf{S}[\![\Delta']\!]_S \to [\![B]\!]_S.$$

We see that $\mathsf{strategize}(S')$ necessarily has a smaller view function than σ and hence by original assumption we have some term M such that

$$[\![\Gamma, \Delta \mid - \vdash_S M : B]\!]_S = \mathsf{strategize}(S').$$

We can now use the application rule to yield

$$[\![\Gamma', \Delta' \mid f : B \multimap C \vdash_S x(M) : C]\!]_S = (\mathbf{id}_{B \multimap C} \otimes \mathsf{strategize}(S')); \mathbf{eval}_{B \multimap C}$$

We can now use substitution to give a term

$$\Gamma, \Gamma', \Delta, \Delta' \mid f : B \multimap C \vdash_S N[f(M)/x]$$

has the strategy σ' as its semantics as required. ■

Lemma 330 Suppose we are given **SCIR** type A and **SCIR** contexts Γ, Δ and a complete, innocent SCI strategy

$$\sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![A]\!]_S \to [\![\mathbf{com}]\!]_S$$

that is linear applicative in $[\![A]\!]_S$ and has a finite view function.

Under the assumption that for any innocent strategy

$$\sigma' : \mathsf{S}[\![\Gamma']\!]_S \otimes [\![\Delta']\!]_S \to [\![A']\!]_S,$$

where σ' has a strictly smaller view function that σ, there exists some term $\Gamma' | \Delta' \vdash_S M : A'$ such that

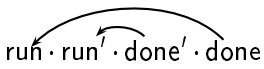
$$[\![\Gamma' | \Delta' \vdash_S M : A']\!]_S = \sigma' : \mathsf{S}[\![\Gamma']\!]_S \otimes [\![\Delta']\!]_S \to [\![A']\!]_S$$

it follows that there exists some term $\Gamma | \Delta, x : A \vdash N : \mathbf{com}$ such that

$$[\![\Gamma | \Delta, x : A \vdash_S N : \tau]\!]_S = \sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes A \to [\![\mathbf{com}]\!]_S.$$

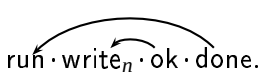
Proof Our proof is by structural induction on the type A. If the sole initial move from $[\![A]\!]_S$ is many answered then our proof is much the same as lemma 329 so we will assume that it is single-valued — an active or passive write or run move.

Base Case Suppose $A = \mathbf{com}$. In this case every single threaded sequence in σ is either of the form $\mathsf{run} \cdot \mathsf{run}'$ or:



and we can define a term $x : \mathbf{com} \vdash_S x$ with semantics σ.

Similarly, if $A = \mathbf{var}$, it is the case that every single threaded sequence in σ is either of the form $\mathsf{run} \cdot \mathsf{write}_n$ or:



and we can define a term $x : \mathbf{var} \vdash_S x := n$ with semantics σ.

Inductive Step

case: Suppose $A = PB$ and our proof is similar to that for lemma 329. First we show that the set σ can be transformed into an innocent strategy for the game $\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\tau]\!]_S$ that is linear

applicative in B. By lemma 325 it follows that σ_{views} obeys view rules **V1-5**. Clearly the set σ_{views} can also be considered as a set of sequences of moves from the game $\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\tau]\!]_S$: the sole difference being that the initial move in $[\![B]\!]_S$ is now possibly labelled active. We verify that the set σ_{views} still obeys view rules **V1-5**, even with this possible change in labelling. The only possible headache might be that the sequences no longer obey the activity condition but this is avoided by our insistence that the morphism points to $[\![\mathbf{com}]\!]_S$. We can now apply lemma 324 to give an innocent strategy $\mathsf{strategize}(\sigma_{\mathsf{views}}) : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![\mathbf{com}]\!]_S$ and inspection confirms that the strategy is linear applicative.

By inductive hypothesis we have some term $\Gamma|\Delta, x : B \vdash_S N : \tau$ such that

$$[\![\Gamma \mid \Delta, x : B \vdash N : \tau]\!]_S = \mathsf{strategize}(\sigma_{\mathsf{views}}) : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes B \to \tau.$$

It is straightforward to show that

$$\sigma = (\mathbf{id}_{\mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S} \otimes \mathbf{der}_B); \mathsf{strategize}(\sigma_{\mathsf{views}})$$

and hence we have

$$[\![\Gamma \mid \Delta, x : A \vdash_S N[\mathbf{der}x/x] : \mathbf{com}]\!]_S = \sigma.$$

case: If $A = B \times C$ then our proof is similar to that in lemma 329.

case: Again the case when $A = B \multimap C$ requires more thought. Let T be that subset of σ_{views} containing exactly the sequences of the form

$$\mathsf{run} \cdot \mathsf{run}' \cdot \mathsf{m} \cdot \mathsf{s}$$

where m is a move from $[\![B]\!]_S$. Let S be the complement, $\sigma_{\mathsf{views}} \setminus T$. Note that the visibility condition ensures that no sequence in S contains a move from $[\![B]\!]_S$.

Recall that $[\![\Delta]\!]_S$ is of the form $[\![A_1]\!]_S \otimes \cdots \otimes [\![A_n]\!]_S$. Now we allocate each of the A_i to three contexts, Δ_P, Δ_S and Δ_T, as follows:

- $A_i \in \Delta_P$ if and only if no active initial move from $[\![A_i]\!]_S$ is played in σ.
- $A_i \in \Delta_S$ if and only if there exists some sequence $\mathsf{s} \cdot \mathsf{q}_P \in S$ where q_P is an active initial move from $[\![A_i]\!]_S$.
- $A_i \in \Delta_T$ if and only if there exists some sequence $\mathsf{s} \cdot \mathsf{q}_P \in T$ where q_P is an active initial move from $[\![A_i]\!]_S$.

First note that Δ_P is disjoint from both Δ_S and Δ_T. Once again lemma 325 assures us that σ_{views} obeys the view rules**V1-5**. Note that view rule **V5** implies that Δ_S and Δ_T are also disjoint. This point is very important, it is how we will be able ensure that the terms we construct in our definability proof can be typed using a multiplicative function application rule.

Clearly $[\![\Delta]\!]_S \simeq [\![\Delta_P]\!]_S \otimes [\![\Delta_S]\!]_S \otimes [\![\Delta_T]\!]_S$. A simple re-tagging of moves formed by disjoint unions in conjunction with a passification of $[\![\Delta_P]\!]_S$ gives us a strategy

$$\sigma' : \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Gamma]\!]'_S \otimes \mathsf{S}[\![\Delta_P]\!]_S \otimes \mathsf{S}[\![\Delta'_P]\!]_S \otimes [\![\Delta_S]\!]_S \otimes [\![\Delta_T]\!]_S \otimes [\![B \multimap C]\!]_S \to [\![\mathbf{com}]\!]_S$$

where we now have two copies of $[\![\Delta_P]\!]_S$, and $[\![\Gamma]\!]_S$, and we create sets S' and T' by an appropriate re-tagging of the moves of the sequences in S and T respectively so that T' uses moves from $[\![\Gamma]\!]_S$ and $[\![\Delta_P]\!]_S$ and S' uses moves from the copies $[\![\Gamma']\!]_S$ and $[\![\Delta'_P]\!]_S$. We are going to define a term with semantics σ' and from this it is then straightforward to use the contraction and activation rules to construct a term with semantics σ.

It is straightforward to check that T' obeys view rules**V1-5** and hence we have a linear applicative strategy $\mathsf{strategize}(T')$ for the arena

$$\mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Gamma']\!]_S \otimes \mathsf{S}[\![\Delta_P]\!]_S \otimes \mathsf{S}[\![\Delta'_P]\!]_S \otimes [\![\Delta_T]\!]_S \otimes [\![\Delta_S]\!]_S \otimes [\![B \multimap C]\!]_S \to [\![\mathbf{com}]\!]_S.$$

The definition of T' assures us that no move in either $[\![\Delta_S]\!]_S$, $\mathsf{S}[\![\Delta'_P]\!]_S$ or B is played in any sequence in this strategy so we can re-tag the moves to give a strategy

$$\mathsf{strategize}(T')' : \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta_P]\!]_S \otimes [\![\Delta_T]\!]_S \otimes [\![C]\!]_S \to [\![\mathbf{com}]\!]_S$$

and by inductive hypothesis we have some term N such that

$$[\![\Gamma, \Delta_P \mid \Delta_T, x : C \vdash_S N : \tau]\!]_S = \mathsf{strategize}(T')'$$

Now we consider the set S' and recall that it contains only sequences of the form

$$\mathsf{q} \cdot \mathsf{q}' \cdot \mathsf{m} \cdot \mathsf{s}$$

where m is a move from $[\![B]\!]_S$. As in lemma 329 σ obeys view rules **V1-5** and it is straightforward to show that S' obeys view rules **V1** and **V3-5**, but not **V2**: S' is not even-length prefix closed as each sequence commences with two questions. However it is straightforward to show that removing the first two moves from each sequence in S' in conjunction with some re-tagging leaves us with a set S'' of sequences of moves from the game $\mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta_P]\!]'_S \otimes [\![\Delta_S]\!]_S \to [\![B]\!]_S$. Furthermore, it is straightforward to show that S'' obeys view rules **V1-5** and hence we can construct the innocent strategy

$$\mathsf{strategize}(S'') : \mathsf{S}[\![\Gamma]\!]'_S \otimes \mathsf{S}[\![\Delta_P]\!]'_S \otimes [\![\Delta_S]\!]_S \to [\![B]\!]_S.$$

We see that $\mathsf{strategize}(S'')$ is necessarily a smaller view function than σ and hence by original assumption we have some term M such that

$$[\![\Gamma', \Delta'_P | \Delta_S \vdash_S M : B]\!]_S = \mathsf{strategize}(S'').$$

We can now use the application rule to yield

$$[\![\Gamma, \Delta_P \mid \Delta_S, f : B \multimap C \vdash_S f(M) : C]\!]_S = (\mathbf{id}_{B \multimap C} \otimes \mathsf{strategize}(S'')); \mathbf{eval}_{B \multimap C}.$$

Now we can construct the following term with the required semantics σ'':

$$\Gamma, \Gamma', \Delta_P, \Delta'_P \mid \Delta_S, \Delta_T, f : B \multimap C, \vdash_S N[f(M)/x].$$

■

Lemma 331 Suppose we are given a complete, innocent SCI-strategy, with finite view function,

$$\sigma : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \to [\![\tau]\!]_S$$

such that Γ, Δ are **SCIR** contexts and τ is an **SCIR** base type **N** or **com**, we can define a term

$$\Gamma \mid \Delta \vdash_S M : \tau$$

with semantics σ.

Proof We consider the case where $\tau = \mathbf{N}$; the case where $\tau = \mathbf{com}$ is similar and uses lemma 330. Our proof is by induction on the size of the view function, similar to that in [7]. As in [7] we make use of a *case* construct.

Base Case For our base case we have $\sigma = \{\varepsilon\}$ and for the corresponding term we have $M = \Omega$.

Inductive Step For the inductive step we note that all player views of sequences in σ commence with an opponent question q. We first consider the simple case where player answers q immediately with a and this yields the semantics of some constant: if $\mathsf{a} = \mathsf{done}$ we have $\tau = \mathbf{com}$ and $M = \mathsf{skip}$ otherwise $\tau = \mathbf{N}$ and $M = \mathsf{a}$.

We next consider the case when player responds to q with a question q' in arena $[\![A]\!]_S$. We will assume that q' is a many answered question — the proof is similar if it is a run or a write. Odd length player views of sequences in σ are therefore of one of two forms:

1. $\mathsf{q} \cdot \mathsf{q}' \cdot \mathsf{k} \cdot \mathsf{s}$ where natural number k answers q'. For each move k we make a set T_k of such views. It is straightforward to show that each T_k respects view rules **V1** and **V3-5**. However it is clearly not prefix closed. However if we remove the moves $\mathsf{q}' \cdot \mathsf{k}$ from each sequence in T_k and add the empty sequence we end up with a set T_k' that respects view rules **V1-5**. We can therefore produce complete, innocent strategy $\mathsf{strategize}(T_k')$ for each k. The view function of each σ_k is strictly smaller than that of σ so by the inductive hypothesis we have a term $\Gamma \mid \Delta \vdash_S N_k : \mathbf{N}$ such that $\mathsf{strategize}(T_k') = [\![\Gamma \mid \Delta \vdash_S N_k : \mathbf{N}]\!]_S$. Note also that as the view function is finite, there exists some L for which for all $k \geq L$ we have $\sigma_k = \emptyset$.

2. $\mathsf{q} \cdot \mathsf{q}' \cdot \mathsf{q}'' \cdot \mathsf{s}$ where q'' is justified by q' and is initial in arena $[\![A_j]\!]_S$ where A_j is an **SCIR** type. Let the set of such sequences be S. It is simple to check that these sequences respect view rules **1-5** but it is not the case that the subsequent application of $\mathsf{strategize}(-)$ will yield a complete strategy. We add to the set views of the form $\mathsf{q} \cdot \mathsf{q}' \cdot \mathsf{k} \cdot \mathsf{k}$ for each $0 \leq k \leq L$ to yield the set S'.

   By the activity condition we know that no active move is played in $[\![A_j]\!]_S$ so we can change the tagging of moves in the members of S' to create a set, S'', of single view sequences for the arena:

   $$\mathsf{S}[\![A_j]\!]_S \otimes \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta]\!]_S \to \mathbf{N}.$$

   We now see that $\mathsf{strategize}(S'')$ is a complete linear applicative strategy for the arena

   $$\mathsf{S}[\![A_j]\!]_S \otimes \mathsf{S}[\![\Gamma]\!]_S \otimes \mathsf{S}[\![\Delta]\!]_S \to \mathbf{N}$$

   so by induction hypothesis and lemma 329 we have a term with semantics equal to $\mathsf{strategize}(S'')$. After suitable application of the contraction rule to the copies of A_j and activation of A_j and Δ we have a term $\Gamma \mid \Delta \vdash_S M : \mathbf{N}$ such that

   $$[\![\Gamma \mid \Delta \vdash_S M : \mathbf{N}]\!]_S = \mathsf{strategize}(S').$$

It is now possible to show that the following term has semantics σ:

$$\Gamma \mid \Delta \vdash_S \mathsf{case}_L\ M\ \mathsf{of}\ 0\ N_0 \ldots\ i\ N_i\ \ldots\ > L\ \Omega.$$

We should add the caveat here that when $\tau = \mathbf{com}$ we cannot necessarily split off an extra copy of the arena $[\![A_j]\!]_S$ and then later use the contraction rule, as it may be the case that active initial moves are played from the arena $[\![A_j]\!]_S$ but the nesting condition assures us that no further initial move is played while an active move is played from $[\![A_j]\!]_S$. ■

Theorem 332 Given a complete, innocent SCI-strategy, with finite view function,

$$\sigma : \mathsf{S}([\![\Gamma]\!]_S) \otimes [\![\Delta]\!]_S \to [\![A]\!]_S$$

such that A is an **SCIR** type and Γ, Δ are SCIR contexts, we can define a term

$$\Gamma|\Delta \vdash_S M : A$$

with semantics σ.

Proof Our proof is by structural induction on the type A.

Base Case If A is **N** or **com** then we simply apply lemma 331. If $A = \mathbf{var}$ then the proof is more delicate. We first compose σ with the retraction of mkvar

$$r = [\![-|x : \mathbf{var} \vdash_S \langle \lambda y^{\mathbf{N}}.x := y, !x \rangle]\!]_S$$

to yield a strategy $\sigma; r : (\mathbf{N} \multimap \mathbf{com}) \times \mathbf{N}$. We can uncurry $\sigma; r; \pi_1$, then apply lemma 331 to find a term M_1:

$$[\![\Gamma|\Delta, y : \mathbf{N} \vdash_S M_1 : \mathbf{N}]\!]_S = \mathsf{uncurry}(\sigma; r; \pi_2).$$

and we can now curry the term again to yield so that:

$$[\![\Gamma|\Delta \vdash_S \lambda y^{\mathbf{N}}.M_1 : \mathbf{N}]\!]_S = \sigma; r; \pi_2.$$

We can now use lemma 331 to yield a term M_2 such that:

$$[\![\Gamma|\Delta \vdash_S M_2 : \mathbf{N}]\!]_S = \sigma; r; \pi_2.$$

Now we can now pair our strategies and compose with $\mathbf{mkvar}_\pi$ to yield:

$$[\![\Gamma|\Delta \vdash_S \mathsf{mkvar}\ \lambda y^{\mathbf{N}}.M_1\ M_2 : \mathbf{var}]\!]_S = \sigma.$$

Inductive Step

case: Suppose $A = \mathsf{P}A'$. By the activity condition we know that no active initial player question is asked in the strategy. It is therefore straightforward to show that the set σ is also a strategy for the arena:

$$\mathsf{S}[\![\Gamma \otimes \Delta]\!]_S \to [\![\mathsf{P}A']\!]_S.$$

It is now simple to check that the set σ has the same single threaded plays as a strategy for the arena

$$\mathsf{S}[\![\Gamma \otimes \Delta]\!]_S \to [\![A']\!]_S$$

and by inductive hypothesis there must be some term $\Gamma, \Delta \mid - \vdash_S M : A'$ such that

$$[\![\Gamma, \Delta \mid - \vdash_S M : A']\!]_S = \mathsf{weave}(\sigma_{\mathsf{threads}}).$$

We can now apply the promotion rule followed any necessary applications of the activation rule to yield the following term with semantics σ:

$$\Gamma \mid \Delta \vdash_S \mathbf{promote}(M) : PA'.$$

case: Suppose $A = B \times C$. By the universal property of products we have strategies

$$\sigma_L : \mathsf{S}\Gamma \otimes \Delta \to [\![B]\!]_S$$

and

$$\sigma_R : \mathsf{S}\Gamma \otimes \Delta \to [\![C]\!]_S$$

such that $\langle \sigma_L, \sigma_R \rangle = \sigma$ and by induction hypothesis these strategies are the denotation of some terms

$$[\![\Gamma \mid \Delta \vdash_S M_L : B]\!]_S = \sigma_L$$

and

$$[\![\Gamma \mid \Delta \vdash_S M_R : C]\!]_S = \sigma_R$$

respectively. We can now construct the term

$$\Gamma \mid \Delta \vdash_S \langle M_L, M_R \rangle : B \times C$$

with semantics σ.

case: Finally, suppose $A = B \multimap C$. We uncurry σ to give a strategy

$$\sigma' : \mathsf{S}[\![\Gamma]\!]_S \otimes [\![\Delta]\!]_S \otimes [\![B]\!]_S \to [\![C]\!]_S$$

which, by inductive hypothesis, is the semantics of some term

$$\Gamma \mid \Delta, x : B \vdash_S M : C$$

so we simply apply the abstraction rule to give the following term with semantics σ:

$$\Gamma \mid \Delta \vdash_S \lambda x.M : B \multimap C.$$

■

7.8 Definability

In this section we show how we can extend the definability results from the innocent subcategory of **C** to the category **C** itself. We use a different argument for each of the languages **PCF**, $\mathbf{SCI}_b$ and **SCIR**. The argument for **PCF** is particularly simple, whilst the arguments for $\mathbf{SCI}_b$ and **SCIR** are more complicated but follow the style of the factorization from [7] that we recalled in chapter 4.

7.8.1 Definability for PCF

Lemma 333 Given a sequences $\mathsf{s}, \mathsf{s}' \in \mathcal{H}_{\mathsf{A}}$ containing only passive moves it follows that:

$$\lceil \mathsf{s} \rceil = \lceil \mathsf{s}' \rceil \Leftrightarrow \lceil \mathsf{s} \rceil_{\prec} = \lceil \mathsf{s}' \rceil_{\prec}.$$

Proof The proof is a simple induction on the length of s. ■

This yields the following corollary: given an SCI-strategy,

$$\sigma : [\![A_1]\!]_P \otimes \ldots \otimes [\![A_n]\!]_P \to [\![A]\!]_P$$

such that A and each of the A_i are **PCF** types it follows that σ is innocent.

We can now infer the following definability theorem from proposition 326 and lemma 333.

Theorem 334 Given a compact SCI-strategy,

$$\sigma : [\![A_1]\!]_P \otimes, \ldots, \otimes [\![A_n]\!]_P \to [\![A]\!]_P$$

such that A and each of the A_i are **PCF** types, we can define a term

$$x_1 : A_1 \ldots x_n : A_n \vdash_P M : A$$

with semantics σ.

7.8.2 Definability for $\mathbf{SCI}_b$

In this section we will be interested in defining terms for strategies for arenas of the form

$$[\![A_1]\!]_B \otimes \ldots \otimes [\![A_n]\!]_B \to [\![A]\!]_B$$

where $A_1, \ldots, A_n, A$ are $\mathbf{SCI}_b$ types. We can simplify our reasoning by identifying certain properties possessed by arenas of this kind. We say that a simple arena is *basic* if and only if it contains only active moves. Of course, given $\mathbf{SCI}_b$ types $A_1, \ldots, A_n, A$ it follows that an arena of the above form is basic.

Lemma 335 Given a basic arena A and sequence $\mathsf{s} \in I_{\mathsf{A}}$ and moves $\mathsf{m}, \mathsf{m}' \in \mathsf{s}$ with $\mathsf{m} < \mathsf{m}'$ such that $\mathsf{m} \sim \mathsf{m}'$ it follows that:

1. $\mathsf{m} \overset{*}{\prec} \mathsf{m}'$.
2. m is answered before m' is played.

Proof This is proved by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is vacuously satisfied.

Inductive Step Suppose $s = s' \cdot n$. We first apply the inductive hypothesis to s' and now we only have to show that for any $m \in s$ such that $m \sim n$ we have $m \overset{*}{\preceq} n$ and also m is answered in s'. Suppose n is initial then from the definition of a basic arena we must also have m as initial and hence $m \leftrightsquigarrow n$ and by the nesting condition there exists some move $a \in s_{<n}$ that answers m. By the SCI condition if n is a player move, or by construction otherwise, it follows that $a \overset{*}{\preceq} n$ and hence $m \overset{*}{\preceq} n$ by lemma 244 and transitivity.

If n is not initial then m is not initial, again by virtue of the definition of a basic arena, so let $j \curvearrowleft m$ and $j' \curvearrowleft n$. Now as A is basic we also have $j \sim j'$. By lemma 239 we know that j' is open in s and hence by inductive hypothesis we know that j is answered by some move a before j' is played. By well bracketing it follows that m must be answered before a is played. We now have the following:

$$\begin{array}{rcl} j & \overset{*}{\preceq} & m \text{ by lemma 244} \\ m & \overset{*}{\preceq} & a \text{ by lemma 312} \\ j & \overset{*}{\preceq} & j' \text{ by inductive hypothesis} \\ a & \overset{*}{\preceq} & j' \text{ by the lemma 250} \\ j' & \overset{*}{\preceq} & n \text{ by lemma 244} \\ m & \overset{*}{\preceq} & n \text{ by transitivity.} \end{array}$$

■

Lemma 336 Given a basic arena A with sequence $s \in I_A$ and moves $m, m' \in s$ we have

$$m \vdash m' \Rightarrow m \overset{*}{\prec\!\succ} m'$$

Proof Clearly $m \neq m'$ as this contradicts the alternation of $\vdash$.

Let us consider the case when $m < m'$. As m' is not initial we let $j \curvearrowleft m'$ then we can apply lemma 244 to yield $j \overset{*}{\preceq} m'$. Furthermore, from the definition of a basic arena we have $m \sim j$. If $m \leq j$ then we know that $m \overset{*}{\preceq} j$ from lemma 335 so we can apply transitivity to yield $m \overset{*}{\preceq} m'$. Alternatively, we might imagine $j < m$ but in this case we can apply lemma 335 to show that j must be answered before m is played, and by lemma 239 we know that j could not justify m' as this move is also played after j has been answered.

We now consider the case when $m' < m$. Once again as m' is not initial we let $j \curvearrowleft m'$ and apply lemma 244 to yield $j \overset{*}{\preceq} m'$. From the definition of a basic arena we have $m \sim j$ so by lemma 335 we know that j must be answered before m is played. Let this answer be a. So by lemma 312 we have $m' \overset{*}{\preceq} a$. Furthermore, from lemma 335 we have $j \overset{*}{\preceq} m$ and hence $a \overset{*}{\preceq} m$ by lemma 312. We can now happily apply transitivity to yield $m' \overset{*}{\preceq} m$. ■

Lemma 337 Given an SCI strategy σ for a basic arena A and sequences $s, s' \in \sigma$ such that

- s is completable and $\lceil s \rceil$ is of the form $t \cdot j \cdot q_O \cdot u \cdot m$ where q_O is an opponent question.
- $\lceil s' \rceil$ is of the form $t' \cdot j' \cdot q'_O \cdot u' \cdot m'$ where q'_O is an opponent question.

- $\lceil t \cdot j \rceil_{\preceq} = \lceil t' \cdot j' \rceil_{\preceq}$.

It follows that if $m \sim m'$ or $m \vdash m'$ or $m' \vdash m$ then it must be the case that $q_O \sim q'_O$.

Proof We will assume that $q_O \not\sim q'_O$ and show that it cannot be the case that $m \sim m'$ or $m \vdash m'$ or $m' \vdash m$.

The proof is similar to that for lemma 313. First we take any extension $u \cdot a_P \in \sigma$ where a_P answers q_O. We know that such a sequence exists as we required that s be completable. By lemma 306 it follows that $\mathbf{min}(u \cdot a_P) \in \sigma$. By lemma 312 we know that $m \in \mathbf{min}(u \cdot a_P)$ and by lemma 246 we know that j and q_O are also in $\mathbf{min}(u \cdot a_P)$ so, to recap, the sequence must be of the form:

$$\ldots j \ldots q_O \ldots m \ldots a_P.$$

We now consider the form of $\mathbf{min}(s')$ and find it to be of the form:

$$v \cdot j' \cdot v' \cdot q'_O \cdot v'' \cdot m'.$$

In a similar style to the proof of lemma 311 we can show that the sequence

$$u \cdot a_P \cdot v' \cdot q'_O \cdot v'' \cdot m'$$

must also be in σ. As $q_O \not\sim q'_O$ it follows that $q_O \overset{*}{\not\leftrightarrow} q'_O$ by lemma 299. Hence by lemma 246 $m \overset{*}{\not\leftrightarrow} m'$ and by lemmas 335 and 336 we know that it cannot be the case that $m \sim m'$ or $m \vdash m'$ or $m' \vdash m$. ■

Given a simple arena A, we assume some injective function, *code*, from SCI views of single-threaded plays from I_A to the natural numbers. Note that the existence of such an encoding is guaranteed by the countability of M_A. Given $s \in I_A$ we write $\overline{s}$ as notation for $\mathrm{code}(\lceil \mathbf{thread}(s) \rceil_{\preceq})$. For simplicity, we will insist here that $\overline{\varepsilon} = 0$. We are now going to define a transformation on strategies such that uses this encoding so that player can infer the SCI-view of the current thread solely using information in the player view, and thus simulate player's response from the original strategy whilst playing as a function of the player view.

Definition 338 (sim) Given a basic arena A, with $\sim$ equivalence classes of opponent moves $c_1, \ldots, c_n$, we inductively define a function **sim** from single threaded plays in I_A to single viewed sequences of moves from the arena

$$\mathbf{var}_1^{\alpha} \otimes \ldots \otimes \mathbf{var}_n^{\alpha} \to A.$$

We will simultaneously define a mapping $\mathsf{protector}$ from questions in $\mathbf{sim}(s) \upharpoonright \mathbf{var}_1^{\alpha} \otimes \ldots \otimes \mathbf{var}_n^{\alpha}$ to moves in s. In our definition we will subscript each question from $\mathbf{var}_1^{\alpha} \otimes \ldots \otimes \mathbf{var}_n^{\alpha}$ with a $\sim$ equivalence class of opponent moves.

1. As our basis we have $\mathbf{sim}(\varepsilon) = \varepsilon$.

   The definition of $\mathbf{sim}(s)$ when $s = s' \cdot m$ is dependent on λm, and is defined by cases.

2. If $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$ and m is an initial opponent question then $\mathbf{sim}(\mathsf{s})$ is defined as follows:

$$\begin{array}{ccccc} \mathbf{var}_1^\alpha & \otimes \ldots \otimes & \mathbf{var}_n^\alpha & \rightarrow & \mathsf{A} \\ & & & \vdots & \\ & & & \mathbf{sim}(\mathsf{s}'_{\leq \mathsf{j}}) & \\ & & & \vdots & \\ & & & & \mathsf{m} \\ \mathsf{read}_{[\mathsf{m}]} & & & & \\ \overset{\bullet}{\overset{\bullet}{\varepsilon}} & & & & \end{array}$$

In this case we define $\mathsf{protector}(\mathsf{read}_{[\mathsf{m}]}) = \mathsf{m}$.

3. If $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$ and m is an opponent question justified by move j and n is the greatest move in s such that $\mathsf{n} \lhd \mathsf{m}$ then $\mathbf{sim}(\mathsf{s})$ is defined as follows:

$$\begin{array}{ccccc} \mathbf{var}_1^\alpha & \otimes \ldots \otimes & \mathbf{var}_n^\alpha & \rightarrow & \mathsf{A} \\ & & & \vdots & \\ & & & \mathbf{sim}(\mathsf{s}'_{\leq \mathsf{j}}) & \\ & & & \vdots & \\ & & & & \mathsf{m} \\ \mathsf{read}_{[\mathsf{m}]} & & & & \\ \overset{\bullet}{\overline{\mathsf{s}_{\leq \mathsf{n}}}} & & & & \end{array}$$

In this case we define $\mathsf{protector}(\mathsf{read}_{[\mathsf{m}]}) = \mathsf{m}$.

4. If $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$ and m is a player answer to a question q_O then we define $\mathbf{sim}(\mathsf{s})$ as follows:

$$\begin{array}{ccccc} \mathbf{var}_1^\alpha & \otimes \ldots \otimes & \mathbf{var}_n^\alpha & \rightarrow & \mathsf{A} \\ & & & \vdots & \\ & & & \mathbf{sim}(\mathsf{s}') & \\ & & & \vdots & \\ \mathsf{write}(\overline{\mathsf{s}})_{[\mathsf{q}_O]} & & & & \\ \overset{\bullet}{\mathsf{ok}} & & & & \\ & & & & \mathsf{m} \end{array}$$

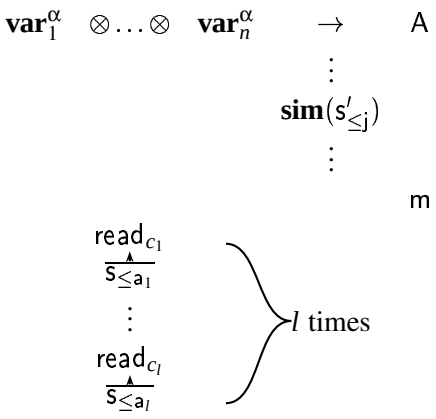
In this case $\mathsf{protector}(\mathsf{write}(\overline{\mathsf{s}})_{[\mathsf{q}_O]}) = \mathsf{q}_O$.

5. If $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$ and m is a player question that enables opponent questions in $\sim$ equivalence classes $c'_1, \ldots, c'_l$ then we define $\mathbf{sim}(\mathsf{s})$ as follows:

$$\begin{array}{ccccc} \mathbf{var}_1 & \otimes \ldots \otimes & \mathbf{var}_n & \rightarrow & \mathsf{A} \\ & & & \vdots & \\ & & & \mathbf{sim}(\mathsf{s}') & \\ & & & \vdots & \\ \mathsf{write}(\overline{\mathsf{s}' \cdot \mathsf{m}})_{c'_1} & & & & \\ \overset{\bullet}{\mathsf{ok}} & & & & \\ \vdots & & & & \\ \mathsf{write}(\overline{\mathsf{s}' \cdot \mathsf{m}})_{c'_l} & & & & \\ \overset{\bullet}{\mathsf{ok}} & & & & \\ & & & & \mathsf{m} \end{array} \quad \Big\} \; l \text{ times}$$

In this case $\mathsf{protector}(\mathsf{write}(\overline{\mathsf{s}'\mathsf{m}})_{c'_i}) = \mathsf{m}$ for $1 \leq i \leq l$.

6. Suppose $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$ and m is an opponent answer to a player question j that enables opponent questions in $\sim$ equivalence classes $c'_1, \ldots, c'_l$ and $\mathsf{a}_1, \ldots, \mathsf{a}_l$ are either the answers to the most recent representatives from each of these classes that is justified by j or else j itself if no such representative exists. We define $\mathbf{sim}(\mathsf{s})$ as follows:



In this case $\mathsf{protector}(\mathsf{read}_{c_i}) = \mathsf{j}$ for $1 \leq i \leq l$.

Lemma 339 It is straightforward to check that the following hold for every single-threaded $\mathsf{s} \in I_\mathsf{A}$:

- $\mathbf{sim}(\mathsf{s}) \in I_{\mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha \to \mathsf{A}}$.
- $\mathbf{sim}(\mathsf{s}) \upharpoonright \mathsf{A} = \lceil \mathsf{s} \rceil$.
- $\mathbf{sim}(\mathsf{s})$ is single viewed.

Definition 340 We overload the definition of **sim** to act on strategies as follows:

$$\mathbf{sim}(\sigma) = \{\mathsf{s} \mid \exists \mathsf{t} \in \sigma_\mathsf{comp}.\mathsf{s} \sqsubseteq_\mathsf{even} \mathbf{sim}(\mathsf{t})\}.$$

Lemma 341 Given a strategy σ for a basic arena A and sequences $\mathsf{s}, \mathsf{s}' \in \mathbf{sim}(\sigma)$ with questions $\mathsf{m} \in \mathsf{s} \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n$ and $\mathsf{m}' \in \mathsf{s}' \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$ such that $\mathsf{m} \sim \mathsf{m}'$ it follows that we have one of the following:

- $\mathsf{protector}(\mathsf{m}) \sim \mathsf{protector}(\mathsf{m}')$.
- $\mathsf{protector}(\mathsf{m}) \vdash \mathsf{protector}(\mathsf{m}')$.
- $\mathsf{protector}(\mathsf{m}') \vdash \mathsf{protector}(\mathsf{m})$.

Proof Proof is by inspection of the definition of $\mathsf{protector}$. ■

Lemma 342 Given sequences $\mathsf{s}, \mathsf{s}' \in \mathsf{A}$ where A is basic it follows that:

$$(\mathbf{sim}(\mathsf{s}) = \mathbf{sim}(\mathsf{s}')) \Leftrightarrow (\lceil \mathbf{thread}(\mathsf{s}) \rceil_\prec = \lceil \mathbf{thread}(\mathsf{s}') \rceil_\prec).$$

Proof This follows from a simple induction on the length of s using lemma 298. ■

Lemma 343 Given a strategy σ for a basic arena A it follows that $\mathbf{sim}(\sigma)$ obeys view rules **V1-5**.

Proof

V1 We have already stated in lemma 339 that $\mathbf{sim}(\sigma) \subseteq I_{\mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha \to A}$.

V2 It follows directly from the definition of **sim** applied to strategies that even-length prefix closure is implied: $s \cdot m \cdot m' \in \mathbf{sim}(\sigma) \Rightarrow s \in \mathbf{sim}(\sigma)$.

V3 For all sequences $s \cdot m \cdot n, s' \cdot m' \cdot n' \in \mathbf{sim}(\sigma)$ such that $s \cdot m = s' \cdot m'$ we need to prove that $s \cdot m \cdot n = s' \cdot m' \cdot n'$. We prove this by induction on the length of $s \cdot m \cdot n$.

Base Case If $s \cdot m \cdot n = \varepsilon$ then the proof is vacuous.

Inductive Step Let k be the last opponent move from A in $s \cdot m \cdot n$ and let k' be the last opponent move from A in $s' \cdot m' \cdot n'$. Inspection of the definition of **sim** implies that there must exist sequences $t \cdot k \cdot l$ and $t' \cdot k' \cdot l'$ in σ such that $\mathbf{sim}(t \cdot k) = s \cdot m_{\leq k}$ and $\mathbf{sim}(t' \cdot k') = s' \cdot m'_{\leq k'}$. Suppose $s \cdot m = s' \cdot m'$, then we have the following:

$$\begin{array}{rcl} s \cdot m_{\leq k} & = & s' \cdot m'_{\leq k'} \text{ by inspection of the definition of } \mathbf{sim}. \\ \lceil \mathbf{thread}(t \cdot k) \rceil_{\prec} & = & \lceil \mathbf{thread}(t' \cdot k') \rceil_{\prec} \text{ by lemma 342.} \\ \lceil t \cdot k \cdot l \rceil_{\prec} & = & \lceil t' \cdot k' \cdot l' \rceil_{\prec} \text{ by SCI-innocence and thread-independence.} \end{array}$$

We can now inspect the definition of **sim** again to note that all moves following k in $s \cdot m$ are matched by moves following k' in $s' \cdot m'$ and hence $s \cdot m \cdot n = s' \cdot m' \cdot n'$.

V4 Each sequence $s \in \mathbf{sim}(\sigma)$ respects active visibility as all moves are active.

V5 Suppose we have sequences $s \cdot q_O \cdot t \cdot q_P, s \cdot q'_O \cdot t' \cdot q'_P \in \mathbf{sim}(\sigma)$ where:

- q_O and q'_O are opponent questions.
- $q_P \sim q'_P$.
- q_P and q'_P are either both initial or both justified by the same move in s.
- q_P is active — which will always be true as arena A is basic.

We need to show that it follows that $q_O \sim q'_O$. We should note, of course, that q_O and q'_O are from arena A and if they are initial then by the definition of a basic arena it immediately follows that $q_O \sim q'_O$. So let j be the last move in s — and of course it follows that $j \curvearrowleft q_O, q'_O$ as the sequences are single viewed. There must exist single-threaded sequences $u, u' \in \sigma$ such that $s \cdot q_O \cdot t \cdot q_P \sqsubseteq \mathbf{sim}(u)$ and $s \cdot q'_O \cdot t' \cdot q'_P \sqsubseteq \mathbf{sim}(u')$. We now examine the definition of **sim** and note that there is a series of $\overline{\mathsf{write}_{u_{\leq j}}} \cdot \mathsf{ok}$ moves immediately prior to j in s. We examine the series and note that by lemma 298 we know that this implies that $\lceil u_{\leq j} \rceil_{\prec} = \lceil u'_{\leq j} \rceil_{\prec}$. We now have two cases to consider:

case: Suppose q_P and q'_P are from arena A. We know from the definition of $\mathbf{sim}(\sigma)$ that u and u' are completable and it is straightforward to show that $q_O \in \lceil u_{\leq q_P} \rceil$ and $q'_O \in \lceil u'_{\leq q'_P} \rceil$ so we can apply lemma 313 to yield $q_O \sim q'_O$.

case: Now suppose that q_P and q'_P are from one of the $\mathbf{var}^\alpha$ arenas. It follows that $q_O \in s \cdot q_O \cdot t \cdot q_{P \leq \mathsf{protector}(q_P)}$ and $q'_O \in s \cdot q'_O \cdot t' \cdot q'_{P \leq \mathsf{protector}(q'_P)}$ and it is straightforward to show that $q_O \in \lceil u_{\leq \mathsf{protector}(q_P)} \rceil$ and $q'_O \in \lceil u'_{\leq \mathsf{protector}(q'_P)} \rceil$ so we can apply lemma 337 to yield $q_O \sim q'_O$.

As a corollary it follows from lemma 324 that $\mathsf{strategize}(\mathbf{sim}(\sigma))$ is an innocent SCI strategy. It is also straightforward to show that $\mathsf{strategize}(\mathbf{sim}(\sigma))$ is complete. ■

Lemma 344 Given a strategy σ for a basic arena A and a single-threaded sequence $s \in \mathsf{strategize}(\mathbf{sim}(\sigma))$ such that

$$s \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$$

is a multi-cell trace, as defined in section 206, that is either empty or ends in a move from A it follows that:

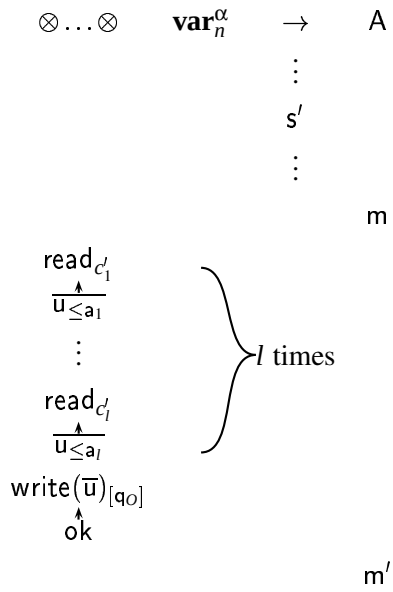
H1 $s \upharpoonright \mathsf{A} \in \sigma$.

H2 For each $\sim$ equivalence class c of opponent moves in A the most recent write move, if it exists, in $\mathbf{var}_c^\alpha$ is of the encoding $\overline{s_{\leq k}}$ where k is the greatest move that is either an answer to a move in c or enables the members of c.

Proof Our proof is by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is trivial.

Inductive Step Let m and m' be the last two moves in $s \upharpoonright \mathsf{A}$. We will consider only the case when m and m' are answers as the argument when given each of the other three possible *QA* labellings for m and m' is similar and yields no surprises. We let j justify m and enable opponent questions from equivalence classes $c'_1, \ldots, c'_l$ and we let the justifier of m' be q_O. By inspection of the definitions of $\mathsf{strategize}(-)$ and **sim** we may infer that s has the following form:



By inductive hypothesis **H1** we know that $s' \upharpoonright \mathsf{A} \in \sigma$ and as the referee I is biased we know that $s \cdot m \upharpoonright \mathsf{A} \in I_{\mathsf{A}}$. From the definition of $\mathsf{strategize}(-)$ we know that there must be some sequence $u \in \sigma$ such that $\mathbf{sim}(u) = \lceil s \rceil$ and by lemma 298 and inductive hypothesis **H2** we know that $\lceil u_{\leq m} \rceil_\prec = \lceil s \cdot m \upharpoonright \mathsf{A} \rceil_\prec$ and hence by the SCI-innocence of σ we must have

$$s \upharpoonright \mathsf{A} \in \sigma \text{ and } \lceil s \upharpoonright \mathsf{A} \rceil_\prec = \lceil u \rceil_\prec$$

so our lemma is satisfied. ■

Lemma 345 Given a strategy σ for a basic arena A and a completable single-threaded sequence $s \in \sigma$ then there exists a sequence $t \in \mathsf{strategize}(\mathbf{sim}(\sigma))$ such that

$$t \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$$

is a multi-cell trace and $s = t \upharpoonright A$.

Proof Our proof is by induction on the length of s.

Base Case If $s = \varepsilon$ then the lemma is trivial.

Inductive Step Let $s = s' \cdot m \cdot m'$. We will consider only the case when m and m' are answers as the argument for each of the other three possible *QA* labellings for m and m' is similar. We let j justify m and enable opponent questions from equivalence classes $c'_1, \ldots, c'_l$ and we let the justifier of m' be q_O. By inductive hypothesis we know that there is some single-threaded sequence $t' \in \mathsf{strategize}(\mathbf{sim}(\sigma))$ such that

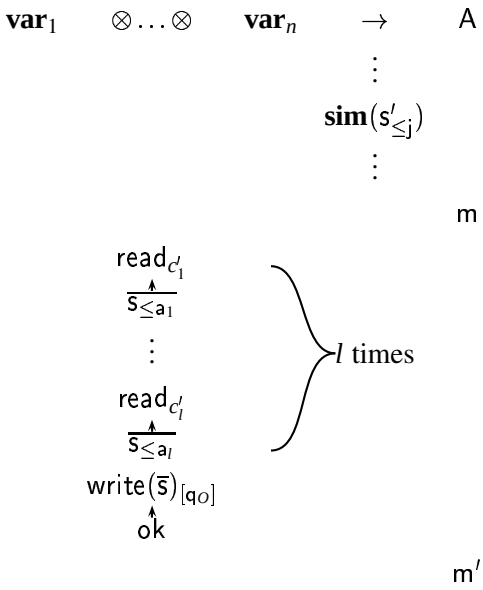
$$s' \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$$

is a multi-cell trace and $s' = t' \upharpoonright A$.

As the referee I is biased we know that

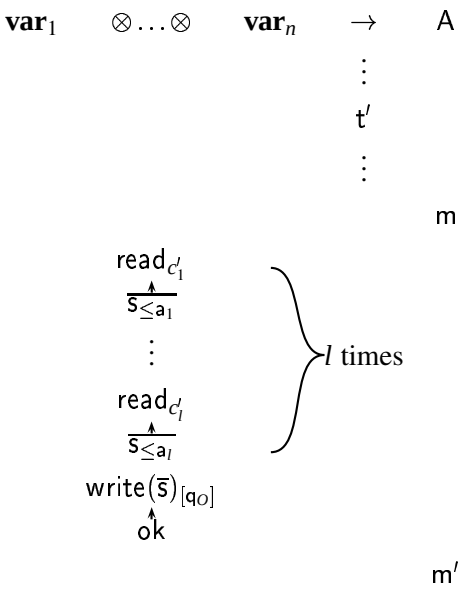
$$t' \cdot m \in I_{\mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha \to A}.$$

We see that $\mathbf{sim}(s)$ has the following form:



It therefore follows by innocence and thread-independence that the following sequence, t, is in

$\mathsf{strategize}(\mathbf{sim}(\sigma))$:



Clearly we have $\mathsf{s} = \mathsf{t} \upharpoonright \mathsf{A}$. Furthermore, using lemma 344 we can see that the read moves are consistent with the claim that component

$$\mathsf{s} \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$$

is a multi-cell trace. ■

Lemma 346 Given a negative basic arena A and a compact strategy $\sigma : \mathsf{A}$ then there exists an innocently compact strategy

$$\mathsf{strategize}(\mathbf{sim}(\sigma)) : \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha \to A$$

such that for all complete single threaded sequences $\mathsf{s} \in I_\mathsf{A}$ it follows that $\mathsf{s} \in \sigma$ if and only if there exists a sequence $\mathsf{t} \in \mathsf{strategize}(\mathbf{sim}(\sigma))$ such that

$$\mathsf{t} \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$$

is a multi-cell trace and $\mathsf{t} \upharpoonright \mathsf{A} = \mathsf{s}$.

Proof The proof follows from lemmas 344 and 345 and from the thread-independence of σ. ■

Theorem 347 (Definability for $\mathbf{SCI}_b$) Given $\mathbf{SCI}_b$ type A and any compact strategy $\sigma[\![A]\!]_B$ we can define a $\mathbf{SCI}_b$ term

$$\vdash_B M : A$$

such that

$$[\![M]\!]_{B_\mathsf{comp}} = \sigma_\mathsf{comp}.$$

Proof We know from lemma 346 that we can construct an innocently compact strategy

$$\mathsf{strategize}(\mathbf{sim}(\sigma)) : \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha \to [\![A]\!]_B$$

such that there exists a sequence $\mathsf{s} \in \mathsf{strategize}(\mathbf{sim}(\sigma))$, with

$$\mathsf{s} \upharpoonright \mathbf{var}_1^\alpha \otimes \ldots \otimes \mathbf{var}_n^\alpha$$

a multi-cell trace, if and only if $\mathsf{s} \upharpoonright [\![A]\!]_B \in \sigma$. We can apply theorem 332 to yield a term

$$x_1 : \mathbf{var}, \ldots, x_l : \mathbf{var} \vdash_B M : A$$

such that $[\![x_1 : \mathbf{var}, \ldots, x_n : \mathbf{var} \vdash_B M : A]\!]_B = \mathsf{strategize}(\mathbf{sim}(\sigma))$. We can now construct a term by binding all the identifiers in the context to yield a term

$$\mathsf{new}\ x_1 := 0\ \mathsf{in}\ (\ldots(\mathsf{new}\ x_n := 0\ \mathsf{in}\ M)\ldots).$$

It is now straightforward to show that

$$[\![\mathsf{new}\ x_1 := 0\ \mathsf{in}\ (\ldots(\mathsf{new}\ x_n := 0\ \mathsf{in}\ M)\ldots)]\!]_{B_{\mathsf{comp}}} = \sigma_{\mathsf{comp}}.$$

■

7.9 Definability for SCIR

Our proof for the definability of **SCIR** is by *factorization* and therefore there are some similarities between this proof and that for the definability of $\mathbf{IA}_a$ that we outlined in chapter 4. Also, as we shall see, our proof bears some obviously similarity to the proof of the definability of $\mathbf{SCI}_b$. Unfortunately our proof is considerably more complicated than either of these proofs.

The intuition behind our factorization is similar to that which was behind the definability proof for $\mathbf{SCI}_b$. We will provide a construction which translates an SCI strategy $\sigma : \mathsf{A}$, where A is an **SCIR** type, into an innocent strategy $\sigma' : \mathsf{O} \to \mathsf{A}$; we will define O shortly. The idea is that player uses stateful behaviour by opponent in the arena O so that an encoding of the SCI-views of the threads in σ are rendered visible in the player views of sequences in σ'. We will make use of the injection from SCI-views of single threads to natural numbers that we assumed in the proof of definability of $\mathbf{SCI}_b$. Our translation is somewhat dependent upon the basic types that we choose to include in the language, and upon whether these types are passive or active. We have already chosen to investigate the situation where we have an active unit type, **com**, a passive natural number type, $\mathbf{N}_\pi$ and a storage variable type **var** which stores these passive numbers. Perhaps we should note here that the factorization is a little easier if we include active natural number types, although we eschew this approach so as to remain true to Reynolds' ideas in [50, 47]. Given a simple arena A, we are going to define a function $\mathbf{sim}'$ from I_A to justified sequences of moves taken from the following arena

$$(\mathsf{P}((\mathbf{var}^\pi \multimap \mathbf{com}) \multimap \mathbf{com}) \otimes \mathsf{P}(\mathbf{var}^\pi \multimap \mathbf{com}) \multimap \mathbf{N}_\pi) \to \mathsf{A}.$$

In the interests of brevity, let NEW be $\mathsf{P}((\mathbf{var}^\pi \multimap \mathbf{com}) \multimap \mathbf{com})$ and DO be $\mathsf{P}(\mathbf{var}^\pi \multimap \mathbf{com}) \multimap \mathbf{N}_\pi$ and O be $\mathsf{NEW} \otimes \mathsf{DO}$. Given an **SCIR** type A we again make use of the fact that questions in M_A fall into two sets:

- Singly answered questions ie. those that enable only one answer. These will be active or passive versions of run and write.
- Multiply answered questions ie. those that enable many answers. These will be the passive q moves.

Definition 348 ($\mathbf{sim}'$, ice, oce, protector)

The mapping $\mathbf{sim}'$ will be defined inductively on the length of its argument: a sequence from I_{A}. In the inductive case, where the argument is of the form $s \cdot m$, the definition of $\mathbf{sim}'(s \cdot m)$ will depend not only on whether m is an opponent move or a player move and whether m is an answer or a question but also on whether it is active or passive and whether or not it is a multiply or singly answered question.

The idea here is that for each opponent question, $m \in s$ we are going to open a new strand in $\mathbf{sim}'(s) \restriction \mathsf{O}$ to keep track of local stateful behaviour. If m is singly valued then the new strand is in the NEW component. Otherwise it is in the DO component. For each occurrence, m of an opponent question in s we define a run move occurrence, in this new strand, which we refer to as the *inner cell justifier*, $(\mathtt{ice}(m))$. The idea will be that we will make write moves justified by $\mathtt{ice}(m)$ in $\mathbf{sim}'(s)$ to encode SCI views of single threads that occur in prefixes of s where m is the most recent unanswered opponent question in the player-view. There will therefore be a one to one correspondence between opponent questions and their inner cell justifiers.

Furthermore, for each non-initial occurrence, m' justified by j, of an opponent question we define a run move in $\mathbf{sim}'(s) \restriction \mathsf{O}$ to be its *outer cell justifier*, $\mathtt{oce}(m')$. The idea here is that whenever a player question is played in s we open up one new strand in $\mathbf{sim}'(s) \restriction \mathsf{DO}$ for each $\sim$ equivalence class of the question moves enabled by m'. We overload our notation here and let $\mathtt{oce}(j, [m'])$ and $\mathtt{oce}(m')$ denote the same move occurrence. We intend write moves justified by $\mathtt{oce}(j, [m'])$ to encode the SCI view of the current thread of a prefix, $t \cdot n$ of s such that n is the greatest active answer to a question in $[m']$ justified by j, or j itself if no such move has been played. Note that the values stored in these variables are similar to those stored in the proof of definability for $\mathbf{SCI}_b$.

Note that we are effectively going to define $\mathbf{sim}'$, ice and oce together. Furthermore, we are going to define a move $\mathsf{protector}(m)$ in s for certain read or write moves in $m \in \mathbf{sim}'(s) \restriction \mathsf{O}$, and we shall use this mapping in later proofs.

$\mathbf{sim}'$ is defined inductively by cases.

As our definition is complicated we will try to give an intuition for each of the cases as they are defined. The idea is that we are in effect defining a set of views from which we can construct an innocent strategy using $\mathsf{strategize}(-)$. We now anthropomorphize player and imagine that when she encounters a player view $\mathbf{sim}'(s)$ she may assume that all play by opponent in O has been consistent with the strategy $\mathbf{new}_\pi \otimes \mathbf{do}_\pi$. She should be able to reconstruct the SCI view of the current thread in s solely by information she sees in the player view, and thus play according to the original strategy σ.

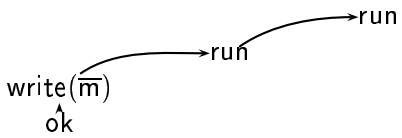
As our basis we have $s = \varepsilon$ and $\mathbf{sim}'(s) = \varepsilon$.

As we have already noted, the definition of $\mathbf{sim}'(s)$ when $s = s' \cdot m$ is dependent on the move m and is defined by cases.

1. If m is a initial singly answered opponent question then $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m}$ is defined as follows with the final run in $\mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{O}$ as $\mathtt{ice}(\mathsf{m})$.

$$(\mathsf{P}((\mathbf{var}^{\pi} \multimap \mathbf{com}) \multimap \mathbf{com}) \otimes \mathsf{DO}) \to \mathsf{A}$$

$$\mathsf{m}$$



   In each of the cases of our definition of $\mathbf{sim}'(\mathsf{s}' \cdot \mathsf{m})$ where m is a player move the sequence is prefixed by the sequence $\mathbf{sim}'(\mathsf{s}') \cdot \mathsf{read}$ where the read move is justified by the $\mathtt{ice}$ of the most recent open opponent move in $\lceil \mathsf{s}' \rceil$. We shall later see that the reply to this question is how the anthropomorphized player assesses the SCI-view of the current thread.

2. Now we specifically consider the case when m is an answer to an initial single valued opponent question, m', we have the following sequence. The penultimate done in $\mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{O}$ and its successor are justified by $\mathtt{ice}(\mathsf{m}')$ and its predecessor respectively. In the remainder of the definition of $\mathbf{sim}'(\mathsf{s}' \cdot \mathsf{m})$ we find that whenever m is a P-move answer, justified by m' then we close the thread in $\mathbf{sim}'(\mathsf{s}' \cdot \mathsf{m})$ associated with $\mathtt{ice}(\mathsf{m}')$ by answering $\mathtt{ice}(\mathsf{m}')$ and its predecessor. In the interest of brevity, we shall henceforth omit this justification information.

   The idea for the anthropomorphized player is that whenever she sees that such an $\mathtt{ice}$ is closed she knows she needs to make a response. The fact that the question is singly answered means that she simply plays the move done.

$$(\mathsf{P}((\mathbf{var}^{\pi} \multimap \mathbf{com}) \multimap \mathbf{com}) \otimes \mathsf{DO}) \to \mathsf{A}$$

$$\vdots$$

$$\mathbf{sim}'(\mathsf{s}')$$

$$\vdots$$

$$\mathsf{read}$$

$$\overline{\mathsf{s}'}$$

$$\mathsf{done}$$

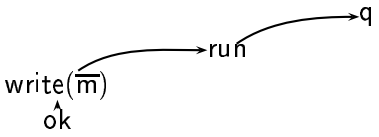
$$\mathsf{done}$$

$$\mathsf{m}$$

3. Similarly, if m is a multiply answered initial opponent question then we have the following, with run as $\mathtt{ice}(\mathsf{m})$.

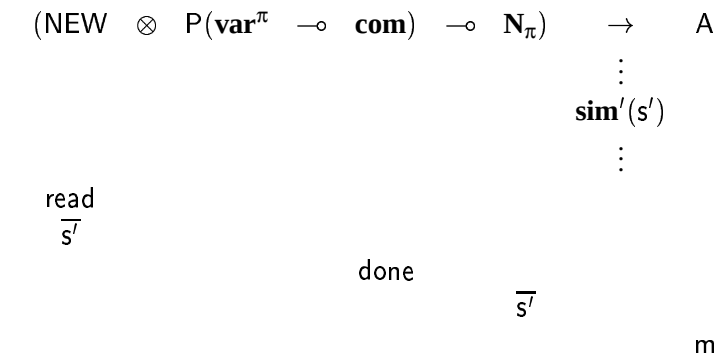
$$(\mathsf{NEW} \otimes \mathsf{P}(\mathbf{var}^{\pi} \multimap \mathbf{com}) \multimap \mathbf{N}_{\pi}) \to \mathsf{A}$$

$$\mathsf{m}$$

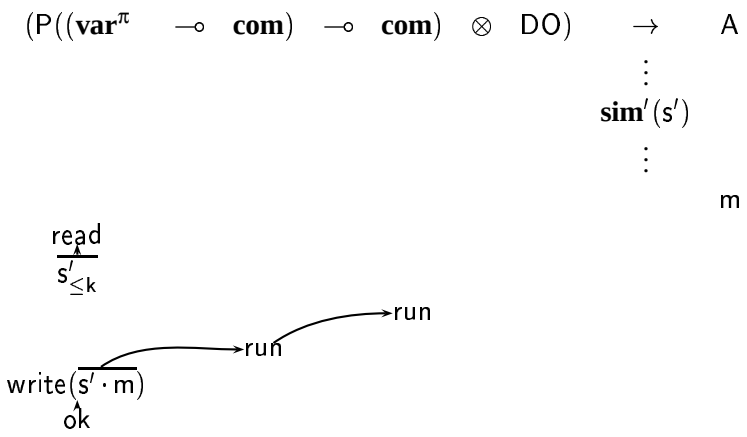


4. If m is an answer to a multiply answered initial opponent question we have the following sequence. Player sees an encoding of the SCI-view of the current thread and plays accord-

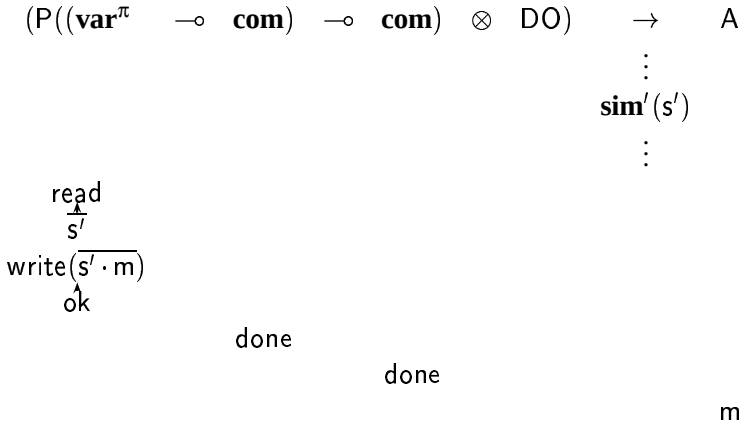
ingly.



5. If m is a singly answered opponent question, justified by j let k be the greatest move in s' such that $k \lhd m$. We have the following, with the second run as ice(m) and read justified by oce(j, [m]). We define protector(read) = m. The idea is that player uses lemma 298 and can infer the SCI-view from m and the SCI-view when k was played. Note too that player updates the inner cell justifier appropriately before playing in A.

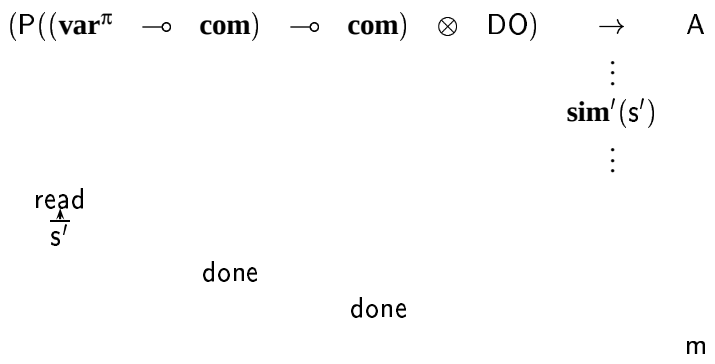


6. If m is an active player answer to a singly answered non-initial question m' then let $j \curvearrowleft m'$. Justification pointers for the moves are as follows. The read is justified by ice(m') while the write($\overline{s' \cdot m}$) is justified by oce(j, [m']). We have protector(write($\overline{s' \cdot m}$)) = m'. Note that after player infers the SCI-view of the current thread by reading from the inner cell justifier before she updates the inner cell justifier with the appropriate encoding before playing in A.

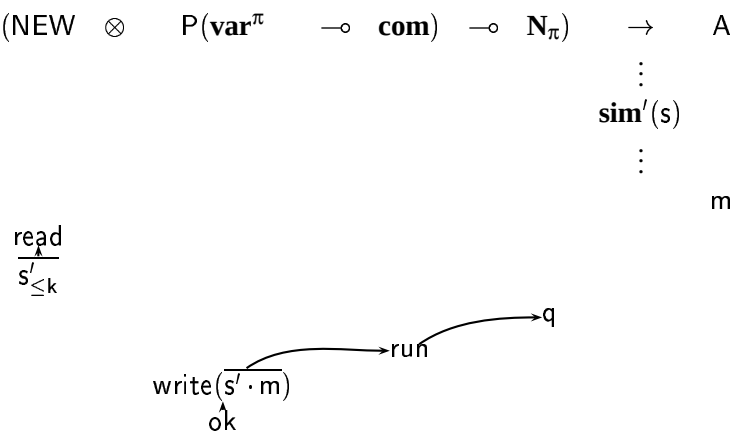


7. Similarly, if m is a passive player answer to a single valued non-initial move m'. The read move is justified by ice(m'). Notice that, unlike the previous case, player does not update

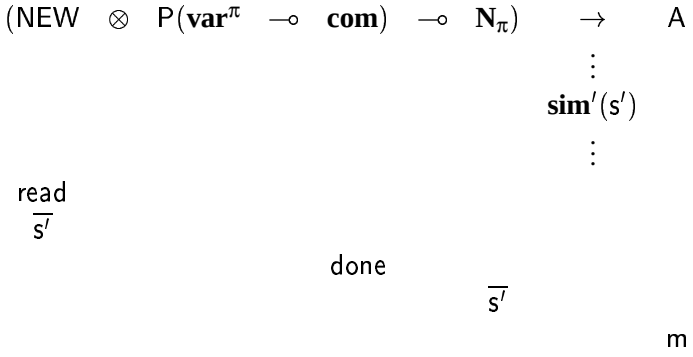
the outer cell justifier before playing in arena A.



8. If m is a many valued opponent question, justified by j, let k be the greatest move in s' such that $k \lhd m$. We have the following, with the second run as ice(m) and read justified by oce(j, [m]). We have protector(read) = m.

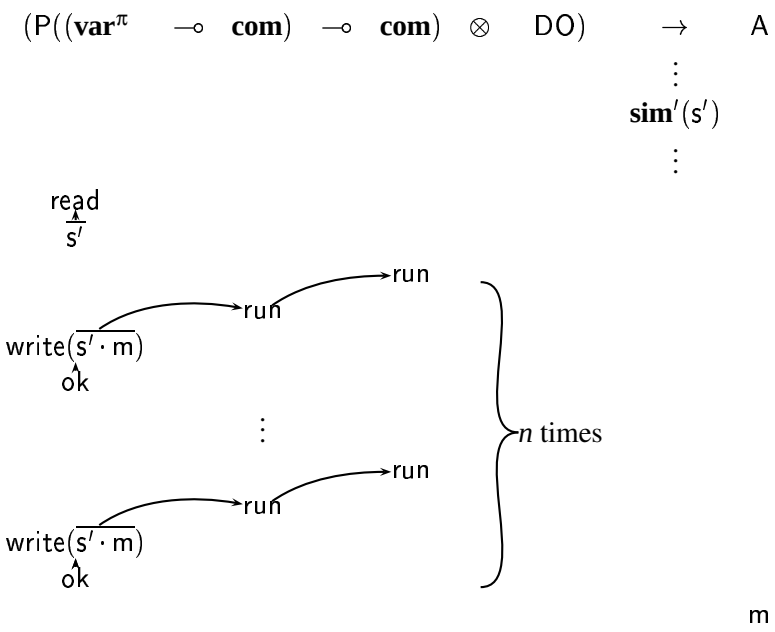


9. If m is a player answer to a many valued non-initial q then we have the following. Once again note that player cannot update the outer cell justifier.

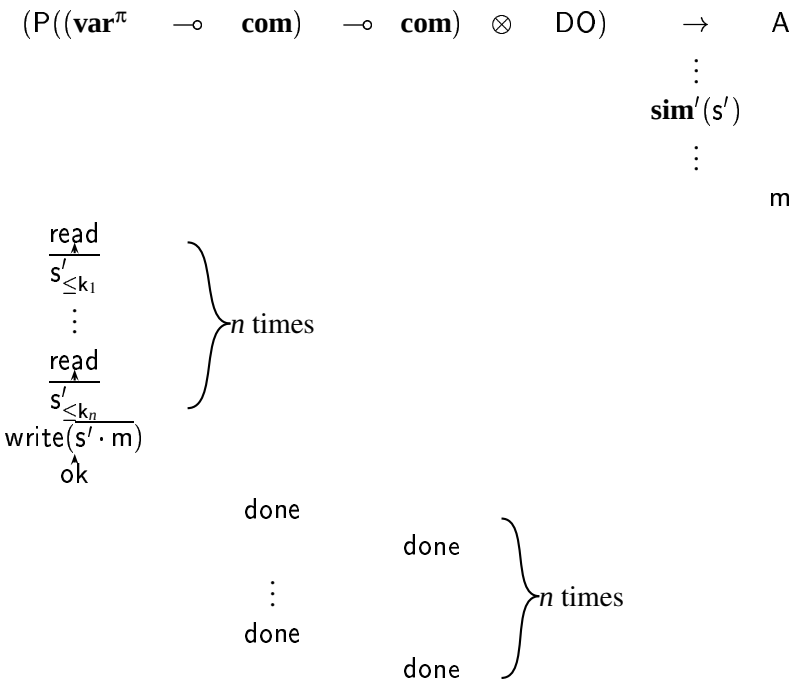


10. If m is a player question we have the following, with read being justified by $\mathtt{ice}(q_O)$ where q_O is the most recent opponent question from A in the player view. Let m enable equivalence classes $c_1, \ldots, c_n$ and we open one new strand in the subgame NEW for each such class.

Each opponent run move is then defined as $\mathtt{oce}(\mathsf{m}, c_i)$ for one of the c_i.



11. Now suppose that m is an opponent answer justified by j and that j enables equivalence classes $c_1, \ldots, c_n$. For all $1 \leq i \leq n$ let k_i be the most recent answer an to active move from c_i that is justified by j, or else j itself if such a move does not exist. We have the following with $\mathsf{write}(\overline{\mathsf{s}' \cdot \mathsf{m}})$ justified by $\mathtt{ice}(\mathsf{q}_O)$ where q_O is the most recent unanswered opponent question in the player view. The repeated $\mathsf{read} \cdot \overline{\mathsf{s}'_{\leq \mathsf{k}_i}}$ and $\mathsf{done} \cdot \mathsf{done}$ sequences belong to the threads opened in the previous case of the definition, and are played in the opposite predetermined order to maintain bracketing.



Note that we have a little work to show that **sim**(s) is well defined— in the definition we claim the existence of justifiers for all the non-initial moves. This situation is easily remedied by the following lemma.

Lemma 349 Given a sequence $s \in I_A$ we have the following:

- $s = \mathbf{sim}'(s) \upharpoonright A$.
- For any opponent question $q \in \mathbf{sim}'(s)$ we have defined a move $\mathtt{oce}(q) \in \mathbf{sim}'(s)$ if q is not initial. Furthermore q is open if and only if $\mathtt{oce}(q)$ is open.
- For any opponent question $q \in \mathbf{sim}'(s)$ we have $\mathtt{ice}(q) \in \mathbf{sim}'(s)$. Furthermore q is open if and only if $\mathtt{ice}(q)$ is open.

Proof The proof is by a simple induction on the length of s. ■

It is not at first obvious that a sequence of the form $\mathbf{sim}'(s)$ is a member of $I_{O \to A}$. We must first check some properties possessed by player views of prefixes of sequences of this form.

Lemma 350 Given a sequence $s \in I_A$ such that s respects active visibility then for all $t \sqsubseteq \mathbf{sim}'(s)$ it follows that:

H1 $\lceil \mathbf{sim}'(t) \rceil \in I_{O \to A}$.

H2 $\mathbf{sim}'(t)$ respects active visibility.

H3 A question move m from $s \upharpoonright A$ is open in $\lceil s \rceil$ if and only if it is open in $\lceil \mathbf{sim}(s) \rceil$.

Proof The proof is by induction on the length of s.

Base Case If $s = \varepsilon$ then **H1-3** are trivial.

Inductive Step The inductive case is long so we will illustrate with the case where $s = s' \cdot m$ and m is a player answer to an active opponent question, m', such that $j \curvearrowleft m'$. In this case recall that $\mathbf{sim}'(s' \cdot m)$ will be of the following form:

$$(\mathsf{P}((\mathbf{var}^{\pi} \multimap \mathbf{com}) \multimap \mathbf{com}) \otimes \mathsf{DO}) \to \mathsf{A}$$

$$\vdots$$

$$\mathbf{sim}'(s')$$

$$\vdots$$

$$\mathsf{read}$$

$$\uparrow$$

$$s'$$

$$\mathsf{write}(\overline{s' \cdot m})$$

$$\uparrow$$

$$\mathsf{ok}$$

$$\mathsf{done}$$

$$\mathsf{done}$$

$$m$$

Recall that justification pointers for the moves are as follows. The read is justified by $\mathtt{ice}(m')$ while the $\mathsf{write}(\overline{s' \cdot m})$ is justified by $\mathtt{oce}(j, \lceil m' \rceil)$.

First we prove **H3**: a question move m from $s \upharpoonright A$ is open in $\lceil s \rceil$ if and only if it is open in $\lceil \mathbf{sim}(s) \rceil$. This is fairly trivial and requires only the inductive hypothesis **H3** applied to $s_{\leq m'}$.

To prove **H1**: we will first prove that $\lceil \mathbf{sim}'(t) \rceil \in \mathcal{L}_{O \to A}$. As the sequence is single viewed it obviously respects opponent visibility. If $t \sqsubseteq \mathbf{sim}'(s')$ then we can apply the inductive hypothesis. Otherwise, we must check the case where the prefix t ends with each of the moves in the diagram. We can check that the justifiers of the read and write moves are in the player view by apply the inductive hypothesis **H3** to s' to see that $m' \in \lceil \mathbf{sim}'(s') \rceil$ and then inspecting the definition of **sim** we can check that both $\mathtt{ice}(m')$ and $\mathtt{oce}(j, \lceil m' \rceil)$ must be in this view. Similarly it is straightforward to check that the answer moves $\mathsf{done} \cdot \mathsf{done} \cdot m$ must necessarily be in response to the most recent question and hence we have well-bracketing. We can now use statement **H3** to show that $\lceil \mathbf{sim}'(t) \rceil$ respects the nesting condition and by lemma 246 it is trivial to show that $\lceil \mathbf{sim}'(t) \rceil$ respects the SCI-condition and hence $\lceil \mathbf{sim}'(t) \rceil \in I_{O \to A}$.

Finally we prove **H2**. We can apply the inductive hypothesis **H2** to $\mathbf{sim}'(s')$ and merely check that their are no passive opponent questions in $\lceil \mathbf{sim}'(s' \cdot m)_{\leq \mathsf{write}(\overline{s' \cdot m})} \rceil$ lying after $\mathtt{oce}(j, \lceil m' \rceil)$. By well-bracketing we know that the only opponent question between $\mathtt{oce}(j, \lceil m' \rceil)$ and the write move is m' and we know this to be active. This point is important as our construction would fail if such a write move were played in the case where m is a passive answer. ■

Lemma 351 Given a sequence $s \in I_A$ and an odd-length prefix $s' \cdot m \sqsubseteq \mathbf{sim}'(s)$ it follows that

$$s' \in I_{O \to A} \Rightarrow s' \cdot m \in I_{O \to A}.$$

Proof Proof is by induction on the length of s and uses the bias of I and inspection of the definition of $\mathbf{sim}'$. ■

Lemma 352 Given sequences $s, t \in A$ it follows that:

$$(\lceil \mathbf{thread}(s) \rceil_{\prec} = \lceil \mathbf{thread}(t) \rceil_{\prec}) \Rightarrow (\lceil \mathbf{sim}'(s) \rceil = \lceil \mathbf{sim}'(t) \rceil).$$

Proof The proof follows by a simple induction on the length of s using lemma 298.

■

Definition 353 We now define the following set of views:

$$\sigma^{\ddagger} = \{ \lceil s \rceil \mid \exists t \in \sigma_{\mathsf{comp}}. s \sqsubseteq_{\mathsf{even}} \mathbf{sim}'(t) \}.$$

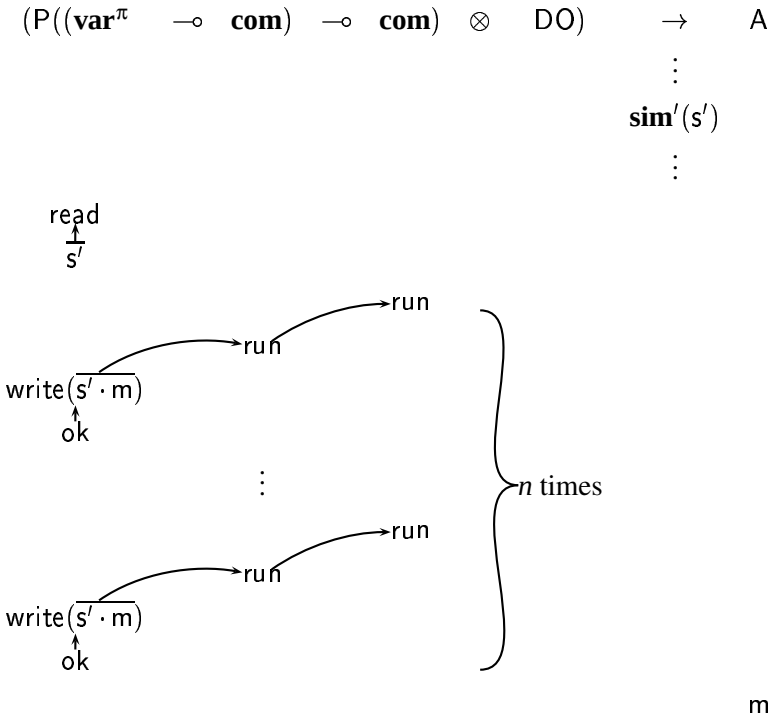
In other words we take all the complete plays in σ, apply $\mathbf{sim}'$ to each of them, and then take the player views of all their even length prefixes.

Lemma 354 Given any sequences $t \cdot n, t \cdot n' \in \sigma^{\ddagger}$ it follows that $t \cdot n = t \cdot n'$.

Proof We check by inspection of the definition of $\mathbf{sim}'$ that player's response is always a function of σ and the current player view. The proof uses lemma 298. Suppose $t \cdot n$ is the player view of some prefix of sequence $\mathbf{sim}'(s \cdot m)$ and we can prove that player responds as a function of σ and the player view by induction on the length of s.

Base Case The base case is vacuously satisfied.

Inductive Step There are many cases to consider, so we take as an example the case where m is a player question. We know that $\mathbf{sim}'(\mathsf{s}\cdot\mathsf{m})$ is of the following form:



case: If n is from $\mathbf{sim}'(\mathsf{s}')$ we can apply the inductive hypothesis to s'.

case: If n is the $\mathtt{read}$ then we note that immediately after any sequence of the form from $\mathbf{sim}'(\mathsf{s}')$ player always plays a read move justified by the most recent $\mathtt{ice}$.

case: The write moves and move m itself are obviously played as a function of the current player view and the strategy σ. ■

Lemma 355 Given a strategy $\sigma : \mathsf{A}$ it follows that $\sigma^{\ddagger}$ obeys view rules **V1-5**.

Proof

V1 We have already shown in lemma 350 that $\sigma^{\ddagger} \subseteq I_{\mathsf{O}\to\mathsf{A}}$.

V2 It follows directly from the definition of $\mathbf{sim}'$ applied to strategies that we have prefix closure: $\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{m}' \in \sigma^{\ddagger} \Rightarrow \mathsf{s} \in \sigma^{\ddagger}$.

V3 For all sequences $\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{n}, \mathsf{s}'\cdot\mathsf{m}'\cdot\mathsf{n}' \in \sigma^{\ddagger}$ such that $\mathsf{s}\cdot\mathsf{m} = \mathsf{s}'\cdot\mathsf{m}'$ it follows from lemma 354 $\mathsf{s}\cdot\mathsf{m}\cdot\mathsf{n} = \mathsf{s}'\cdot\mathsf{m}'\cdot\mathsf{n}'$.

V4 We have shown in lemma 350 that each sequence $\mathsf{s} \in \sigma^{\ddagger}$ respects active visibility.

V5 Suppose we have sequences $\mathsf{s}\cdot\mathsf{q}_O\cdot\mathsf{t}\cdot\mathsf{q}_P, \mathsf{s}\cdot\mathsf{q}'_O\cdot\mathsf{t}'\cdot\mathsf{q}'_P \in \mathbf{sim}'(\sigma)$ where:

- q_O and q'_O are opponent questions.
- $\mathsf{q}_P \sim \mathsf{q}'_P$.
- q_P and q'_P are either both initial or both justified by the same move in s.
- q_P is active.

We need to show that it follows that $\mathsf{q}_O \sim \mathsf{q}'_O$.

case: Suppose that either q_O or q'_O is from arena O. It must, of course, be an $\mathtt{ice}$ or an $\mathtt{oce}$ and as the moves preceding q_O and q'_O are the same it follows that $\mathsf{q}_O = \mathsf{q}'_O$ and hence $\mathsf{q}_O \sim \mathsf{q}'_O$.

case: Suppose that q_O, q'_O, q_P and q'_P are all from arena A. Of course, if q_O and q'_O are initial then it must be the case that $\mathsf{q}_O \sim \mathsf{q}'_O$ by the definition of a simple game. Otherwise let $\mathsf{j} \curvearrowleft \mathsf{q}_O, \mathsf{q}'_O$. It follows from the definition of $-^{\ddagger}$ that there must be some completable sequences $\mathsf{u} \cdot \mathsf{q}_P, \mathsf{u}' \cdot \mathsf{q}'_P \in \sigma$ such that

$$\mathsf{s} \cdot \mathsf{q}_O \cdot \mathsf{t} \cdot \mathsf{q}_P = \lceil \mathbf{sim}'(\mathsf{u} \cdot \mathsf{q}_P) \rceil.$$

and

$$\mathsf{s}' \cdot \mathsf{q}'_O \cdot \mathsf{t}' \cdot \mathsf{q}'_P = \lceil \mathbf{sim}'(\mathsf{u}' \cdot \mathsf{q}'_P) \rceil.$$

By lemma 350 we note that it must also be the case that $\mathsf{q}_O \in \lceil \mathsf{u} \cdot \mathsf{q}_P \rceil$ and $\mathsf{q}'_O \in \lceil \mathsf{u}' \cdot \mathsf{q}'_P \rceil$. By inspection of the definition of $-^{\ddagger}$ we see that it must be the case that $\lceil \mathsf{u}_{\leq \mathsf{j}} \rceil_{\prec} = \lceil \mathsf{u}'_{\leq \mathsf{j}} \rceil_{\prec}$ and hence by lemma 313 we know that $\mathsf{q}_O \sim \mathsf{q}'_O$.

case: Now suppose that q_O and q'_O are from arena A and q_P and q'_P are from O and are justified by an $\mathtt{ice}$. By lemma 339 we know that these moves are justified by the most recent open questions in the player view so the lemma is trivially satisfied.

case: Finally suppose that q_O and q'_O are from arena A and q_P and q'_P are from O and are justified by an $\mathtt{oce}$. Inspection of the definition of $\mathbf{sim}'$ shows us that if q_P or q'_P are generated by one of the last two cases of the definition then it is justified by the most recent unanswered question in the player view and **V5** is trivially satisfied. Otherwise inspection of the definition of $\mathbf{sim}'$ shows us that it must be the case that $\mathsf{q}_O = \mathsf{protector}(\mathsf{q}_P)$, $\mathsf{q}'_O = \mathsf{protector}(\mathsf{q}'_P)$ and and also that $\mathsf{protector}(\mathsf{q}_P) \sim \mathsf{protector}(\mathsf{q}'_P)$.

As a corollary it follows from lemma 324 that $\mathsf{strategize}(\sigma^{\ddagger})$ is an innocent SCI strategy and it is straightforward to show that it is complete. It also follows from lemma 351 that for all completable s in σ we have $\mathbf{sim}'(\mathsf{s}) \in \mathsf{strategize}(\sigma^{\ddagger})$. ■

Lemma 356 Given an even length sequence $\mathsf{s} \in I_{\mathsf{A}}$ it follows that:

H1 $\mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{O} \in \mathbf{new}_{\pi} \otimes \mathbf{do}_{\pi}$.

H2 For any opponent question $\mathsf{q}_O \in \mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{A}$ the most recent write move justified by $\mathtt{ice}(\mathsf{q}_O)$ is of the form $\mathsf{write}(\overline{\mathbf{thread}(\mathsf{t})})$ where t is the greatest prefix of $\mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{A}$ such that q_O is the most recent unanswered question in the player view.

H3 For any player question $\mathsf{q}_P \in \mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{A}$ that enables player questions from equivalence classes $c_1, \ldots, c_l$ it follows that the most recent write move justified by $\mathtt{oce}(\mathsf{q}_P, c_i)$ is of the form $\mathsf{write}(\overline{\mathbf{thread}(\mathsf{t} \cdot \mathsf{k})})$ where $\mathsf{t} \cdot \mathsf{k}$ is the greatest prefix of $\mathbf{sim}'(\mathsf{s}) \upharpoonright \mathsf{A}$ such that either k is an active answer to a question in c_i that is justified by q_P or else $\mathsf{q}_P = \mathsf{k}$.

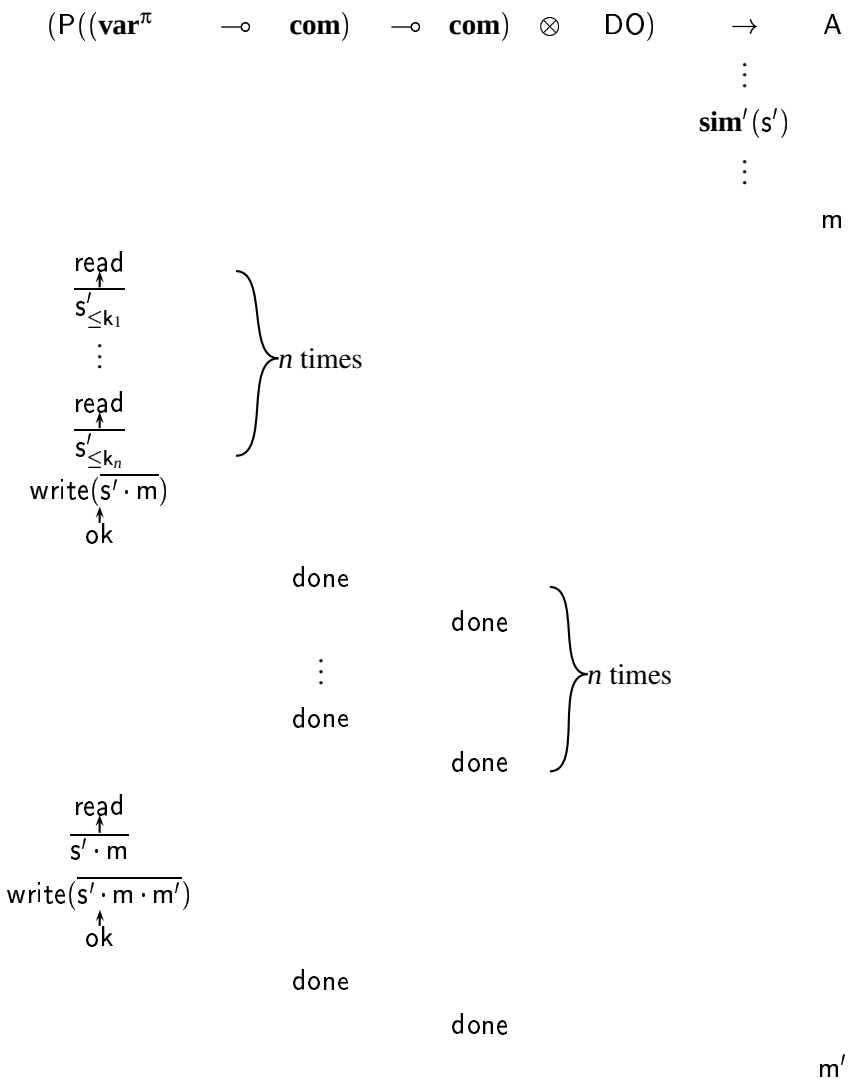
Proof Our proof is by induction on the length of s.

Base Case If $\mathsf{s} = \varepsilon$ then the lemma is trivial.

Inductive Step Let $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{m}'$. We will consider only the case when m and m' are answers as there are numerous cases and they can all be proved in a similar way. We let j justify m and enable opponent questions from equivalence classes $c_1, \ldots, c_l$ and we let the justifier of m' be q_O. For all

$1 \leq i \leq n$ let k_i be the most recent answer to an active move from c_i that is justified by j, or else j itself if no such answer exists.

We see that $\mathbf{sim}'(\mathsf{s})$ has the following form.



We first apply inductive hypotheses **H1**, **H2** and **H3** to s'. By inductive hypotheses **H2** and **H3** we see that the read moves in our diagram validate **H1**. Furthermore, the write moves in our diagram validate **H2** and **H3**. ■

Lemma 357 Given a strategy $\sigma : \mathsf{A}$ and a sequence $\mathsf{s} \in \mathsf{strategize}(\sigma^{\ddagger})$ such that s is empty or finishes with a move in A and such that $\mathsf{s} \upharpoonright \mathsf{O} \in \mathbf{new}_{\pi} \otimes \mathbf{do}_{\pi}$ then it follows that there must be some completable sequence $\mathsf{t} \in \sigma$ such that $\mathbf{sim}'(\mathsf{t}) = \mathsf{s}$.

Proof Our proof is by induction on the length of s.

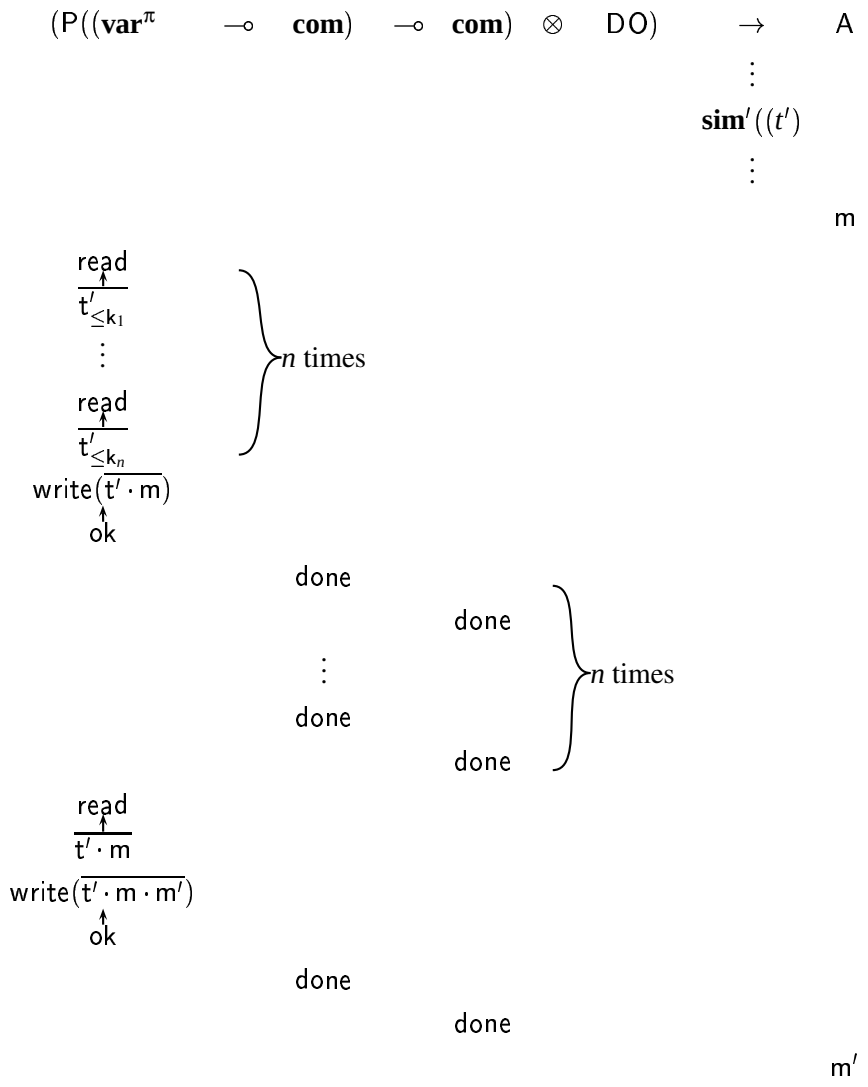
Base Case If $\mathsf{s} = \varepsilon$ then the lemma is trivial.

Inductive Step Let $\mathsf{s} = \mathsf{s}' \cdot \mathsf{m} \cdot \mathsf{s}'' \cdot \mathsf{m}'$ where m and m' are from A and all moves in s'' are from arena O. By inductive hypothesis we know that $\mathsf{s}' = \mathbf{sim}'(\mathsf{t}')$ for some completable $\mathsf{t}' \in \sigma$.

We will consider only the case when m and m' are active answers, as the other cases can all be proved by similar means. We let j justify m and enable opponent questions from equivalence

classes $c_1, \ldots, c_l$ and we let the justifier of m' be q_o. For all $1 \leq i \leq n$ let k_i be the most recent answer an to active move from c_i that is justified by j, or else j itself if no such answer exists.

We are going to prove that $\mathsf{t}' \cdot \mathsf{m} \cdot \mathsf{m}' \in \sigma$ and that s has the following form:



We must work backwards here. By inspection of the definition of $\sigma^{\ddagger}$ we see that player will only play the active answer m' if it is immediately preceded by a done move that closes the predecessor of the ice of its justifier. If $\mathsf{s} \upharpoonright \mathsf{O} \in \mathbf{new}_\pi \otimes \mathbf{do}_\pi$ it must be the case that the move before the opponent done, is a player done that closes the ice. Player will only play such a move if it is preceded by an ok. This in turn must follow a write move that writes an encoding of the SCI-view of some single-threaded sequence in σ, and this encoding must be a one move extension of the answer to the previous read, and this read move must follow a cycle of done moves as shown. The answer to the read move must also match the last move written — such a write · ok pair must immediately precede the done moves. Before this write move there must be a cycle of read moves followed by suitable encodings — we use lemma 298 here to show that the value written is a function of these encodings and the move m which is also in the view. These encodings must be exactly those as shown by the inductive hypothesis and lemma 356, hence s must be of the form shown and it must also follow that $\mathsf{t}' \cdot \mathsf{m} \cdot \mathsf{m}' \in \sigma$. ■

Lemma 358 Given an **SCIR** type A and a compact strategy $\sigma : [\![A]\!]_S$ the innocently compact strategy $\mathsf{strategize}(\sigma^{\ddagger}) : \mathsf{O} \to [\![\mathsf{A}]\!]_{\mathsf{S}}$ is such that there exists a completable sequence $\mathsf{s} \in \sigma$ if and only if there exists a completable sequence $\mathsf{t} \in \mathsf{strategize}(\sigma^{\ddagger})$ such that $\mathsf{t} \upharpoonright \mathsf{O} \in \mathbf{new}_{\pi} \otimes \mathbf{do}_{\pi}$ and $\mathsf{t} \upharpoonright [\![A]\!]_S = \mathsf{s}$.

Proof The proof follows from lemmas 356 and 357. ■

Theorem 359 (Definability for $\mathbf{SCI}_b$) Given **SCIR** type A and a compact strategy $\sigma : [\![A]\!]_S$ we can define an **SCIR** term

$$\vdash_S M : A$$

such that

$$[\![M]\!]_{S_{\mathsf{comp}}} = \sigma_{\mathsf{comp}}.$$

Proof We know from lemma 358 that we can construct a complete innocently compact strategy

$$\mathsf{strategize}(\sigma^{\ddagger}) : \mathsf{O} \to [\![\mathsf{A}]\!]_{\mathsf{S}}$$

such that there exists a complete sequence $\mathsf{s} \in \mathsf{strategize}(\sigma^{\ddagger})$, with $\mathsf{s} \upharpoonright \mathsf{O} \in \mathbf{new}_{\pi} \otimes \mathbf{do}_{\pi}$, if and only if $\mathsf{s} \upharpoonright [\![A]\!]_S \in \sigma$. We can apply theorem 332 to yield a term

$$- \mid f : \mathsf{P}((\mathbf{var} \to \mathbf{com}) \to \mathbf{com}), g : \mathsf{P}(\mathbf{var} \to \mathbf{com}) \to \mathbf{N} \vdash_S M : A$$

such that $[\![M : A]\!]_S = \mathsf{strategize}(\sigma^{\ddagger})$. We can now construct a term by binding the identifiers in the context to yield a term

$$- \mid - \vdash_S (\lambda f.\lambda g.M)(\mathsf{promote}\ (\lambda h.\mathsf{new}\ y := 0\ \mathsf{in}\ hy))(\lambda h.\mathsf{do}\ y := 0\ \mathsf{then}\ gy)$$

It is now straightforward to show that the completable sequences in the semantics of this term coincide exactly with those of σ. ■

7.10 Full Abstraction

Full abstraction for **PCF**, $\mathbf{SCI}_b$ and **SCIR** does not hold in **C** but the definability result enables the construction of a fully abstract model as in chapter 4.

We will consider our interpretation of **SCIR** in **C** as the treatment of the other languages is similar.

Clearly given any open **SCIR** terms

$$x_1 : A_1, \ldots, x : A_{n-1} \vdash_S M : A_n \text{ and } y_1 : A_1, \ldots, y : A_{n-1} \vdash_S M' : A_n$$

it follows that

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash_S M : A_n \sqsubseteq_{\sim} y_1 : A_1, \ldots, y : A_{n-1} \vdash_S M' : A_n$$

if and only if

$$\lambda x_1 \ldots \lambda x_{n-1}.M : A_n \sqsubseteq_{\sim} \lambda y_1 \ldots \lambda y_{n-1}.M' : A_n.$$

Furthermore, for any context $C[-]$: **com** we can create by abstraction an **SCIR** term $C' : A \multimap$ **com** such that for any closed **SCIR** term $M : A$ we have

$$C[M]\Downarrow_S \Leftrightarrow C'(M)\Downarrow_S.$$

There are only two legal strategies for the arena $[\![\textbf{com}]\!]_S$: $[\![\mathsf{skip}]\!]_S$ and $[\![\Omega]\!]_S$. Any strategy $\rho : \mathsf{A} \to$ **com** can be thought of as a *test* for the semantics of closed terms of type $[\![A]\!]_S$.

Definition 360 Given two strategies $\sigma : \mathsf{A}$ and $\tau : \mathsf{B}$ we say $\sigma \lesssim \tau$ if and only if for all $\rho : \mathsf{A} \to$ **com** it follows that:

$$\sigma;\rho = \top \Rightarrow \tau;\rho = \top.$$

In other words $\sigma \lesssim \tau$ if and only if τ passes all the tests that σ does.

It is straightforward to show that if there exists some test that can distinguish a pair of strategies then there must exist a complete, compact test that also distinguishes them. In other words, given two strategies $\sigma : \mathsf{A}$ and $\tau : \mathsf{B}$ then $\sigma \lesssim \tau$ if and only if for all *complete, compact* $\rho : \mathsf{A} \to$ **com** it follows that:

$$\sigma;\rho = \top \Rightarrow \tau;\rho = \top.$$

However we have already shown that all such tests, for the arenas in which we are interested, are the semantics of some **SCIR** term. Hence the following holds:

$$M \sqsubseteq N \Leftrightarrow [\![M]\!]_S \lesssim [\![N]\!]_S.$$

The definition of $\lesssim$ exactly captures the observational preorder in denotational terms and hence we have full abstraction. Note, however, that the full abstraction result does not directly yield a fully abstract quotient of **C**, as it did in chapter 4. We cannot lift our definition of $\lesssim$ to arbitrary morphisms in **C** as we do not have exponentials of every object. We do of course, have all exponentials in the subcategory **P** and so we can collapse the model of **PCF**.

Chapter 8

Conclusions

The aim of this thesis was to develop an accurate game semantics for languages that meet the SCI design criteria first proposed by Reynolds in [47]. We took as our target languages **PCF**, $\mathbf{SCI}_b$, and **SCIR** and defined a category of arenas, **C**, in which we can give a sound and adequate interpretation of each of the three languages. The main novelties in the construction of this category were the notions of SCI-view —a partial ordering on the moves in a play, and SCI-innocence —the restriction that player must always make a response as a function of the SCI-view. We also proved a definability result which we have shown to be sufficient to yield full abstraction for each of the three languages. In this sense the thesis has been a success and has yielded the first fully abstract model of **SCIR** and the first definability results for a model of $\mathbf{SCI}_b$. The research that comprises this thesis is further evidence of the suitability of game semantics for the purpose of accurately characterizing programming languages.

8.1 Avenues for Further Exploration

Perhaps the most obvious next step is to put our model to work. The equivalence at the beginning of chapter 7 seems unnaturally complicated — but it seems we have to work with the collapsed model if we wish to prove any more natural equivalences. There may be, however, other more general properties of the language that may be proved using our model, such as the impossibility of certain definability results for a bireflective model as proved in chapter 7.

8.1.1 Reducing Proofs to Visibility and Innocence in G

We hope that the novel proofs of chapters 3 and 6 will be of interest. It may be possible to to define a more general notion of game and strategy than those in the category **G** and to retain notions of visibility and innocence.

8.1.2 Improving the Model

There are some respects in which our model might be improved. In the original presentation of **SCIR** in [40] the language contains a type that is not present in the language we model here; a non-interfering product type. This type seems intuitively to encapsulate the very essence of SCI. A term $\langle M, N \rangle$ of this type consists of a pair of subterms containing no "actively used" free identifiers in common. Of course, our semantics allows for non-interfering tensor construct but we conjecture that the operational semantics associated with the non-interfering pair necessitates *lifted* tensor types in the denotational semantics. As our arenas do not allow answers to enable questions we do not have lifting as found in [36, 8, 1] and it is not clear how our category should be extended to incorporate lifting. The addition of a lifting monad to the category may shed light on the nature of interference control in a language using call-by-value. Furthermore, it would be desirable to examine interference control in a setting which permits higher order store — such a setting has recently been explored in unpublished work by Jim Laird.

Our model places a restriction, the nesting condition, on opponent that is not present in the familiar game semantic model of $\mathbf{IA}_a$ in [6]. It might be desirable to have no such constraints on opponent — thus allowing the construction of a model of $\mathbf{IA}_a$ that contains subcategories in which to interpret interference controlled languages however this would mean that the un-collapsed models would no longer equate the observational equivalent terms described at the beginning of chapter 7.

8.1.3 Modelling Further Languages in C

We have yet to discover the extent of the robustness of our model. In showing that **SCIR** is a conservative extension of both **PCF** and $\mathbf{SCI}_b$ we have weakened our claim that **C** is a good candidate for a general model of languages that respect the SCI design principles; any model of **SCIR** with the definability property will also contain models for **PCF** and $\mathbf{SCI}_b$ for which the same definability property holds. To strengthen our claim it would be interesting to investigate the possibility of modelling the language SCI2 proposed in [49].

8.1.4 Further Study of the Model of $\mathbf{SCI}_b$

The subcategory of **C** consisting of arenas and strategies comprised of only active moves in which we interpret the language $\mathbf{SCI}_b$ is particularly interesting for two reasons:

- Because we only use *basic* arenas in the semantics of $\mathbf{SCI}_b$ terms, it is straightforward to show that justification information in the strategies that interpret $\mathbf{SCI}_b$ terms can be inferred — any move must be justified by its most recent enabler. This suggests the possibility of a regular language semantics for $\mathbf{SCI}_b$, with finite data types, in the style of [20, 21, 22, 19, 3]. We should recall here that the interpretation of $\mathbf{SCI}_b$ in **C** is *not* fully abstract — so the benefits of formulating a regular language semantics are less obvious. In fact the decidability of observational equivalence in $\mathbf{SCI}_b$ remains an open question.
- Because we know there already exist effectively presentable fully abstract models of $\mathbf{SCI}_b$ [46, 37] there may be some value in attempting to effectively present the collapse of the subcategory of **C** in which we interpret $\mathbf{SCI}_b$.

8.1.5 Game Semantics for Other "Revisited" Type Systems

The partitioning of the context of a judgement into zones is not confined to **SCIR** — for example it has been used in the logics of [12, 53]. However, the elegant permeability rules that are used to move identifier from zone to zone were peculiar to **SCIR** and have since been used in related type systems that control statefulness [54] and complexity [11]. Some aspects of the model in this thesis, in particular the active visibility due to Guy McCusker in personal communication, and defined in chapter 7, seem particularly relevant to other type systems of this kind and are worthy of further attention.

8.1.6 Views as Partial Orders

The major novel aspect of this work was the relaxation of the concept of view from sequences to partial orders. These ideas have a tradition in the theory of concurrency, notably the pomsets of [45] and their introduction to the game semantic setting may yet bear more fruit. This partial order views seem natural to encapsulate some notion of *independent subcomputation*, whether it be due to passivity, SCI, or a language construct used to control independent threads.

As we have seen, elements in our semantics are sequences. A partial ordering is imposed on the sequences due to labelling and justification information. It might be possible to define a model where the elements themselves are the partial orders however it is unclear how to proceed in this direction — and the inherently sequential nature of the languages modelled in **C** might be lost. We should note that this inherent independence of ordering is also present in the games on graphs in [51] and in the *saturation* of strategies in [21, 22].

The issue of interference control appears more difficult, in languages incorporating pointers, particularly object-oriented languages, and there is a good deal of active research in this area [27, 14, 15]. Future work inspired by our "views as partial orders" ideas for games semantics may result in accurate models of these languages or perhaps inspire the creation of new languages based on compelling semantical properties.

Bibliography

[1] S. Abramsky, K. Honda, and G. McCusker. A fully abstract game semantics for general references. In *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science*, page 334. IEEE Computer Society, 1998.

[2] Samson Abramsky. Axioms for definability and full completeness. In Gordon Plotkin, Colin Sterling, and Mads Tofte, editors, *Essays in Honour of Robin Milner*. MIT Press, 2000.

[3] Samson Abramsky. Beyond full abstraction: model-checking for algol-like languages. Marktoberdorf International Summer School 2001. (lecture slides), 2001.

[4] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF (extended abstract). In Masami Hagiya and John C. Mitchell, editors, *Theoretical Aspects of Computer Software. International Symposium TACS'94*, number 789 in Lecture Notes in Computer Science, pages 1–15, Sendai, Japan, April 1994. Springer-Verlag.

[5] Samson Abramsky and Guy McCusker. Games for recursive types. October 1994.

[6] Samson Abramsky and Guy McCusker. Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions (extended abstract). In *Proceedings of 1996 Workshop on Linear Logic*, volume 3 of *Electronic notes in Theoretical Computer Science*. Elsevier, 1996.

[7] Samson Abramsky and Guy McCusker. Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions. In *ALGOL-like languages (v.2)*, pages 297–329 of volume 2. Birkhauser Boston Inc., 1997.

[8] Samson Abramsky and Guy McCusker. Call-by-value games. In Mogens Nielsen and Wolfgang Thomas, editors, *Computer Science Logic: 11th International Workshop Proceedings*, Lecture Notes in Computer Science, pages 1–17. Springer-Verlag, 1998.

[9] Samson Abramsky and Guy McCusker. Game semantics. In H. Schwichtenberg and U. Berger, editors, *Logic and Computation: Proceedings of the 1997 Marktoberdorf Summer School*. Springer-Verlag, 1998.

[10] Samson Abramsky and Guy McCusker. Full abstraction for Idealized Algol with passive expressions. *Theoretical Computer Science*, 227:3–42, 1999.

[11] Klaus Aehlig, Ulrich Berger, Martin Hofmann, and Helmut Schwichtenberg. An arithmetic for non-size-increasing polynomial-time computation. *Theor. Comput. Sci.*, 318(1-2):3–27, 2004.

[12] Andrew Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, LFCS, University of Edinburgh, 1996.

[13] Gérard Berry, P.-L. Curien, and Jean-Jacques Lévy. Full abstraction for sequential languages: the state of the art. In M. Nivat and John Reynolds, editors, *Algebraic Semantics*, pages 89–132. Cambridge University Press, 1985.

[14] David Clark. *Object Ownership and Containment*. PhD thesis, University of New South Wales, 2002.

[15] David Clark, John Potter, and James Noble. Ownership types for flexible alias protection. In *Proceedings, ACM Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA '98)*, pages 48–64, October 1998.

[16] Roy Crole. *Categories for Types*. Cambridge University Press, 1994.

[17] V. Danos and R. Harmer. The anatomy of innocence. In *Proceedings, Tenth Annual Conference of the European Association for Computer Science Logic*. Springer Verlag, 2001.

[18] P. J. Freyd, P. W. O'Hearn, A. J. Power, M. Takeyama, and R. D. Tennent. Bireflectivity. In *Mathematical Foundations of Programming Semantics, Eleventh Annual Conference, Tulane University, New Orleans, LA, March 29 - April 1, 1995*. Elsevier, 1995.

[19] Dan R. Ghica. Game-based software model checking: Case studies and methodological considerations. Technical Report PRG-RR-03-11, Oxford University Computing Laboratory, May 2003.

[20] Dan R. Ghica and Guy McCusker. Reasoning about Idealized Algol using regular languages. In *Proceedings, Twenty-Seventh International Colloquium on Automata, Languages and Programming*, pages 103–115, 2000.

[21] Dan R. Ghica and Andrzej Murawksi. Angelic semantics of fine-grained concurrency. Accepted for publication in Proceedings, FoSSaCS 2004, 2003.

[22] Dan R. Ghica, A. S. Murawski, and C.-H. L. Ong. Syntactic control of concurrency. To appear in Proceedings, ICALP 2004, Springer LNCS.

[23] Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[24] Russell Harmer. *Games and full abstraction for nondeterministic languages*. PhD thesis, University of London, 1999.

[25] Russell Harmer and Guy McCusker. A fully abstract game semantics for finite nondeterminism. In *Proceedings, Fourteenth Annual IEEE Symposium on Logic in Computer Science*, pages 422–430, 1999.

[26] Charles Anthony Richard Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[27] John Hogg. Islands: Aliasing protection in object-oriented languages. In *Proceedings of the OOPSLA '91 Conference on Object-oriented Programming Systems, Languages and Applications*, pages 271–285, November 1991.

[28] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation*, 162(2):285–408, 2000.

[29] J. Laird. Game semantics of linearly-used continuations. In *Proceedings, FoSSaCS 2003*, volume 2620 of *LNCS*, pages 313–327. Springer-Verlag, 2003.

[30] Jim Laird. Full abstraction for functional languages with control. In *Logic in Computer Science*, pages 58–67, 1997.

[31] Jim Laird. Games, control and full abstraction. Submitted to Theoretical Computer Science, February 2000.

[32] Olivier Laurent. Polarized games. In *Logic in Computer Science*, pages 265–274, Los Alamitos, CA, USA, July 22–25 2002. IEEE Computer Society.

[33] R. Loader. Finitary PCF is not decidable. Unpublished manuscript, 1996.

[34] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, Berlin, 1971.

[35] S. MacLane. *Categories for the working mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, Berlin, 1971.

[36] Guy McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. PhD thesis, Department of Computing, Imperial College, University of London, 1996.

[37] Guy McCusker. A fully abstract relational model of syntactic control of interference. In *Proceedings, Computer Science Logic (CSL) 2002*, volume 2471 of *Lecture Notes in Computer Science*, pages 247–261. Springer-Verlag, 2002.

[38] Guy McCusker. Categorical models for syntactic control of interference revisited, revisited. Draft, 2004.

[39] H. Nickau. Hereditarily sequential functionals. In *Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg*, Lecture notes in Computer Science. Springer, 1994.

[40] P. W. O'Hearn, A. J. Power, M. Takeyama, and R. D. Tennent. Syntactic control of interference revisited. *Theoretical Computer Science*, 228(1–2):211–252, 1999. A preliminary version appeared in the proceedings of MFPS XI.

[41] P. W. O'Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, June 1999.

[42] Peter O'Hearn and Robert D. Tennent. *ALGOL-like languages (v.2)*. Birkhauser Boston Inc., 1997.

[43] P.W. O'Hearn. On bunched typing. *em Journal of Func. Prog.*, 13:747–796, 2003.

[44] Gordon Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[45] Vaughan R. Pratt. Modelling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986.

[46] Uday S. Reddy. Global state considered unnecessary: Object-based semantics for interference-free imperative programs. *Lisp and Symbolic Computation*, 9(1), 1996.

[47] J. C. Reynolds. Syntactic control of interference. In *Conf. Record 5th ACM Symposium on Principles of Programming Languages*, pages 39–46, 1978.

[48] J. C. Reynolds. Idealized Algol and its specification logic. In D. Néel, editor, *Tools and Notions for Program Construction*, pages 121–161. Cambridge University Press, 1982.

[49] J. C. Reynolds. Syntactic control of interference, part 2. In *16th International Colloquiumon Automata, Languages, and Programming, Lecture Notes in Computer Science*, pages 675–700. Springer-Verlag, Berlin (1991), 1991.

[50] John C. Reynolds. The essence of Algol. In *Proceedings of the 1981 International Symposium on Algorithmic Languages*, pages 345–372. North-Holland, 1981.

[51] A. C. Schalk and J. ME. Hyland. Games on graphs and sequentially realizable functionals. extended abstract. In *Proceedings Logic in Computer Science 2002*, pages 257–264. IEEE Press, June 2002. ISBN 0769514839.

[52] Dana S. Scott. Outline of a mathematical theory of computation. Technical Monograph PRG-2, Oxford University Computing Laboratory, November 1970.

[53] P. L. Wadler. A taste of linear logic. In *Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science, Gdánsk*, New York, NY, 1993. Springer-Verlag.

[54] H. Yang and U. Reddy. Imperative lambda calculus revisited, 1997.

[55] Honseok Yang and Howard Huang. Type reconstruction for syntactic control of interference. In *IEEE Computer Society International Conference on Computer Languages 1998, Loyola University, Chicago*, pages 164–173, Los Alamitos, California, 1998.