Institutions for Behaviour Specification[(*)]

J.L.Fiadeiro and **J.F.Costa**

ILTEC & Dept. Informatics
Faculty of Sciences, University of Lisbon,
Campo Grande, 1700 Lisboa, PORTUGAL
{llf,fgc}@di.fc.ul.pt

Abstract.□Capitalising on the profusion of modal logics that have been proposed for reactive system specification since [Pnueli 77], on current work that explores categorical formalisations of models of concurrency such as [Sassone et al 93], and on our own past work relating specification logics and such process models [Fiadeiro and Costa 93, Fiadeiro et al 93, Sernadas et al 94], we develop a notion of institution of behaviour in which structural properties of logics and models that are relevant for specifying system behaviour can be formalised and discussed. In this framework, we characterise and relate the existence of adjoint situations between theories and models with the existence of terminal models and the difference between underspecification and nondeterminism.

1 Introduction

This paper is a preliminary account of a systematic study of the relationships between models of reactive system behaviour and their specifications taken as theories in a given logic. The overall aim of this work is to provide, for reactive system development, the same kind of formal support as has been provided for relational systems through the various algebraic techniques that have been proposed for Abstract Data Type (ADT) specification, namely in what concerns structure and modularity.

The work on ADT specification led to the proposal of institutions [Goguen and Burstall 92] as a means of defining specification concepts and techniques independently of the underlying logic. Our specific purpose in this paper is to put forward a specialisation of the notion of institution as a means of capturing the kind of structural properties that arise in modelling systems that are reactive. The motivation for this specialisation has its origins in three different sources.

Firstly, there is a large body of work on reactive system specification, be it *behaviour*

[(*)] This work was partially supported by the Esprit BRAs 6071 (ISCORE) and 8035 (MODELAGE), the HCM Scientific Network MEDICIS, and JNICT under contract PBIC/C/TIT/1227/92.

oriented, i.e. based on a process language and mathematical models like transition systems, or *logic oriented*, i.e. based on the use of formulas of some logic to specify properties of systems. The use of modal logics, namely temporal logics, has been proposed for this purpose since Pnueli's seminal paper [Pnueli 77]. The relationship between behaviours and formulas of the logic can be established by a *satisfaction relation*. This relation can be defined directly over the syntax of a process language (e.g. [Barringer et al 85] for CSP) or via a correspondence between the process algebra and the semantic structures of the logic (e.g. [Hennessy and Milner 85]).

Secondly, recent work on the behaviour side of concurrency [Sassone et al 93] has shown that category theory can be used to provide an abstract characterisation of typical constructions in process models (e.g. limits in suitable categories characterise parallel composition of processes) and a way of relating different models of concurrency (through adjunctions). Such categorial characterisations bring the field much closer to Goguen's categorial approach to General Systems Theory [Goguen and Ginali 78], and make the semantic domains much easier to "institutionalise".

Thirdly, the work developed in the IS-CORE project (BRA 3023/6071) has investigated categorial techniques on both sides of the satisfaction relation, having built several institutions for object specification based on modal logics [e.g. Fiadeiro and Maibaum 91, 92, Sernadas et al 94] and proposed several categorial models of process and object behaviour [e.g. Ehrich et al 91, Costa et al 92]. In particular, we have recently concerned ourselves with the relationship between the two sides [Fiadeiro et al 93], and have established an adjunction between the theories of a linear temporal logic and a trace-based model of behaviour [Fiadeiro and Costa 93] suggesting that the categorial approach provides a very expressive framework for relating behaviour models and logics.

Questions which arise immediately are "what can be generalised (and how) to other logics and behaviour models?" and "how does the categorial approach compare with the more classical approaches?". Our purpose in this paper is to investigate how the institutional framework can be used to start answering these questions by clarifying what the notion of "categorial approach to behaviour specification" might be.

More precisely, we put forward a notion of institution of behaviour (β-institution) that characterises a subclass of institutions that have the structural properties that we have found useful for modelling behaviour and which are related to well known properties of Kripke semantics of modal logics. We then begin exploring the structural properties of β-institutions, discussing in particular the relationship between "programs/systems" and "specifications", and characterising and relating adjoint situations between theories and models with underspecification and nondeterminism. In the concluding remarks, we discuss some of the future lines of research that we intend to undertake.

2 Institutions of behaviour

We have already mentioned that, following the work of Pnueli and others, we take modal logics as a starting point for the definition of β-institutions. The idea, however, is not to specialise the grammar and model functors of institutions in order to encode Kripke structures and their corresponding modal operators, but to capture the structural properties of Kripke semantics that we have found important for modelling behaviour.

We shall, therefore, dedicate very little attention to the syntactic aspects and adopt, as for institutions, a category SIGN of signatures and a grammar functor Sen: SIGN→SET for the syntactic structures of a β-institution. As an example, we may take any propositional modal logic: SIGN consists of SET (every signature consists of the set of atomic formulae of the logic) and Sen provides, for every choice of atomic formulae, the set of formulae that they generate using a fixed collection of logical connectives.

Let us then concentrate on the model theory of β-institutions. Let us recall some definitions from Kripke semantics of modal logics taken from [Goldblatt 87]:

Definition 2.1: A *frame* is a pair <S,R> where S is a set and R a binary relation on S (R⊆S×S). ∎

The set S is the set of possible worlds (or points, or states) and R is the accessibility relation between worlds.

For instance, frames for linear temporal logic can be defined as follows [Wolper 89]:

Definition 2.2: A frame for linear temporal logic is a pair <S,R> such that R is a total function. ∎

We have already stated that we are interested in the structural property of frames and, hence, on the mappings between frames that preserve such structures. In the mathematical logic literature, frames come equipped with a notion of morphism known as *p-morphism* [Goldblatt 87] or *zigzag morphism* [van Benthem 84]:

Definition□2.3: Let $<S_1,R_1>$ and $<S_2,R_2>$ be frames. A *p-morphism* $f:<S_1,R_1>\rightarrow<S_2,R_2>$ is a function $f:S_1\rightarrow S_2$ satisfying:

- sR_1t implies $f(s)R_2f(t)$;
- $f(s)R_2u$ implies the existence of t such that sR_1t and f(t)=u. ∎

The second condition is a kind of "conservativeness" requirement on the translation between possible worlds. It says that no new alternative worlds are provided in $<S_2,R_2>$ for the worlds which come from S_1. In terms of system behaviour, this means that nondeterminism cannot be increased. We shall see later on why this condition is important and what its impact is on the relationship between specifications and behaviours.

Categorially speaking, we have the following characterisation:

Proposition 2.3: Frames and p-morphisms constitute a category KRI. This category is concrete over SET. ∎

The frames which typically arise in models of system behaviour usually have a different presentation. For instance, consider the model of process behaviour that we adopted in [Fiadeiro and Costa 93]:

Definition 2.4: A process behaviour (or process, for short) is a pair <E,Λ> where E is a pointed set (i.e. a set with a distinguished element $\bot_E$) and Λ is a subset Λ of E^{ω} (i.e., each $\lambda\in\Lambda$ is a function $\lambda:\omega\rightarrow E$, an infinite sequence of elements of E). ▮

Given a process $P=<E,\Lambda>$ we refer to E as P_{α} — the *alphabet* of P — and we refer to Λ as P_{Λ} — the *language* of P. The elements of E are called *events*. The designated event $\bot_E$ corresponds to idle steps of the process (steps that are performed by the environment).

Such process models usually induce frames. For instance, there is a canonical way of generating from a process a frame for linear temporal logic:

Definition/Proposition 2.5: Given a process $P=<E,\Lambda>$, the corresponding frame is $Kri(P)=<S_P,R_P>$ defined as follows:

- $S_P=\{<\lambda,i> \mid \lambda\in\Lambda,\ i\in\omega\}$
- $<\lambda,i>R_P<\mu,j>$ iff $\lambda=\mu$ and $j=i+1$

The relation R_P thus defined is a total function and, hence, we obtain a frame for linear temporal logic as defined in 2.2. ▮

Models of process behaviour also come equipped with morphisms (see [Sassone et al 93] for a classification of several such categories corresponding to different models). Morphisms for the trace-based model defined in 2.4 are defined as follows [Fiadeiro and Costa 93] (also explored in [Dionísio 91]):

Definition□2.6:□Let h be a morphism from a pointed set A to a pointed set B (i.e. h is a total function $A\rightarrow B$ such that $f(\bot_A)=\bot_B$). The *extension* of h to A^{ω} is the function $h^{\omega}:A^{\omega}\rightarrow B^{\omega}$ defined by $h^{\omega}(\lambda)=\lambda;h$. A *process morphism* $h:<P_{\alpha},P_{\Lambda}>\rightarrow<Q_{\alpha},Q_{\Lambda}>$ is a morphism $h:P_{\alpha}\rightarrow Q_{\alpha}$ of pointed sets such that $h^{\omega}(P_{\Lambda})\subseteq Q_{\Lambda}$. ▮

A morphism $h:<P_{\alpha},P_{\Lambda}>\rightarrow<Q_{\alpha},Q_{\Lambda}>$ can be seen as a way of making Q a *component-of* P, or a way in which P can *simulate* Q.

Proposition□2.7: Processes and process morphisms constitute a category PROC. There is a forgetful functor $U_{\bot}:PROC\rightarrow SET_{\bot}$ that sends each process P to its alphabet P_{α} (where $SET_{\bot}$ is the category of pointed sets). This functor is faithful, making PROC concrete over $SET_{\bot}$. ▮

The fact that processes behave as frames for linear temporal logic corresponds to the following result:

Proposition 2.8: Let $h:<P_{\alpha},P_{\Lambda}>\rightarrow<Q_{\alpha},Q_{\Lambda}>$ be a process morphism. Let Kri(h) be the mapping $S_P\rightarrow S_Q$ defined by $Kri(h)(\lambda,i)=(h^{\omega}(\lambda),i)$. The mapping Kri that acts on processes as defined in 2.4. and on morphisms as above is a functor PROC→KRI.
proof: We have to check that every mapping Kri(h) satisfies the conditions for a p-morphism. The first condition is trivially verified. In order to check conservatiness, let $(h^{\omega}(\lambda),i)R_Q(\mu.j)$. By definition of R_Q, $\mu=h^{\omega}(\lambda)$ and $j=i+1$, i.e. $(\mu,j)=Kri(h)(\lambda,i+1)$. But $(\lambda,i)R_P(\lambda,i+1)$ by definition. ▮

Following the motivation given in the introduction and at the beginning of the section, our idea is to adopt for the "semantic" component of β-institutions categories BEHA of process behaviour such as PROC. If, indeed, the logical language is a modal one, then it should be possible to establish a functor BEHA→KRI such as for PROC above in order to provide the semantic structures that are necessary to interpret the language. We shall, however, abstain from making commitments on the existence and the nature of any relationship between BEHA and KRI. The reasons for doing so are twofold. On the one hand, many logics work with more than one accessibility relation, requiring more than one functor mapping BEHA to KRI to exist. On the other hand, we would not like to rule out non-modal formalisms if they satisfy the structural properties that we are going to lay down for β-institutions. However, in the definition of these structural properties, we shall draw on the characteristics of modal logics.

Consider now the relationship between syntax and semantics. In modal logics, language is interpreted over a *model* for the atomic formulae [Goldblatt 87]:

Definition 2.9: Given a signature (set of atomic formulae) Σ, a *Σ-model* is a triple $<S,R,V>$ where $<S,R>$ is a frame and $V: \Sigma \rightarrow 2^S$ is a (total) function. ▮

Hence V is a function assigning to every atomic formula $p \in \Sigma$ a subset $V(p)$ of S that stands for the worlds in which p is true (in the case of linear temporal logic interpreted over PROC, propositions correspond to actions so that V(p) returns the events during which action p occurs).

The way V extends to formulae is logic-dependent. It gives rise to a S-indexed family of truth relations ${}^{a}{}_{s}$ between models and formulae. This relation gives rise to another truth relation ^{a} between models and formulae defined by $M{}^{a}A$ iff $M{}^{a}{}_{s}A$ for every $s \in S$. It is the "truth in a model" relation that we adopt for β-institutions in order to abstract from the choice of the space of possible worlds. This relation will henceforth be called "satisfaction relation" to keep faithful to the traditional terminology of institutions.

It still remains to abstract the notion of model, namely the function V, from standard Kripke semantics to β-institutions. In order to give further motivation of the solution that we adopted, consider the extension of the notion of p-morphism to models [Goldblatt 87]:

Definition 2.10: Let $<S_1,R_1,V_1>$ and $<S_2,R_2,V_2>$ be Σ-models. A p-morphism $f:<S_1,R_1,V_1> \rightarrow <S_2,R_2,V_2>$ is a frame p-morphism $f:<S_1,R_1> \rightarrow <S_2,R_2>$ such that $s \in V_1(p)$ iff $f(s) \in V_2(p)$ for every $p \in \Sigma$. ▮

If we write the condition in the following equivalent way:

$$V_1=V_2;f^{-1}$$

we see that the contravariant power-set functor is being applied to functions between possible worlds as a means of connecting the domains of the two valuation functions. The same functor appears in the definition of valuation functions (2.9) operating on possible worlds. This suggests that the relationship between syntax and semantics in a β-institution be based on a functor Sign: BEHA → SIGN^{op}.

Models over a signature Σ can now be defined as in 2.9: a Σ-model is a pair $<\sigma,B>$ where B is a behaviour and $\sigma:\Sigma \rightarrow \mathrm{Sign}(B)$. According to 2.10, a morphism f: $<\sigma_1,B_1> \rightarrow <\sigma_2,B_2>$ is now a morphism $f:B_1 \rightarrow B_2$ such that $\sigma_1=\sigma_2;\mathrm{Sign}(f)$. That is, the category $\mathrm{Mod}(\Sigma)$ of models for a signature Σ is the comma category $(\mathrm{Sign}\downarrow\Sigma)$.

Given a signature morphism $\sigma:\Sigma \rightarrow \Sigma'$, we can easily defined the associated reduct functor as mapping every Σ'-model $<\sigma,B>$ to the Σ-model $<\mu;\sigma,B>$. That is, we define

Mod:SIGN→CAT^{op} to be the functor (Sign$\downarrow$_).

Given these definitions, the satisfaction condition of institutions translates to:

for every σ:Σ→Σ', μ:Σ'→Sign(B) and A∈Sen(Σ), $<\mu,B>^{a}{}_{\Sigma'}\sigma(A)$ iff $<\sigma;\mu,B>^{a}{}_{\Sigma}A$

We can now make precise the definition that we have been progressively building:

Definition 2.11: An institution of behaviour (β-institution) consists of:

<u>syntax</u>:

- a category SIGN (of signatures)
- a functor Sen: SIGN → SET (grammar)

<u>semantics</u>:

- a category BEHA (of behaviours)
- a functor Sign: BEHA→SIGN^{op}

<u>model-theory</u>:

- a |SIGN|-indexed family of satisfaction relations $<\sigma,B>^{a}{}_{\Sigma}A$ where A∈Sen(Σ) and σ:Σ→Sign(B), satisfying the following property: for every σ:Σ→Σ', μ:Σ'→Sign(B) and A∈Sen(Σ), $<\mu,B>^{a}{}_{\Sigma'}\sigma(A)$ iff $<\sigma;\mu,B>^{a}{}_{\Sigma}A$ ▮

As examples of β-institutions we have, as expected, modal logics for which BEHA is a subcategory of KRI whose objects are the frames which satisfy the intended requirements on the accessibility relation (e.g. linearity as in 2.2). In the case of propositional logics, SIGN is SET and Sign is the power-set functor applied to the forgetful functor that projects frames into their underlying sets of possible worlds as discussed above.

As an example of a β-institution built over a process model, we have linear temporal logic over PROC as defined in [Fiadeiro and Costa 93]. Another example is given in [Sernadas et al 94] for a category of labelled transition systems. There are other obvious associations of logics with some of the process models defined in [Sassone et al 93] to form β-institutions, e.g. branching time logic for synchronisation trees, but no systematic assignment as been attempted yet.

As intended, β-institutions are a specialisation of the notion of institution:

Definition and Proposition 2.12: Every β-institution defines a corresponding institution as follows:

- its category of signatures is SIGN as for the β-institution;
- its grammar functor is Sen as for the β-institution;
- its model functor is (Sign$\downarrow$_): SIGN→CAT^{op};
- its family of satisfaction relations is ^{a} as for the β-institution ▮

3 Specifications and the p-property

All the components of β-institutions were motivated with standard notions of the model theory of modal logics except for the satisfaction condition. This property has no counterpart in modal logics because changes of language, as captured by signature morphisms, are not usually studied in mathematical and philosophical logic.

There is, however, an important structural property satisfied by Kripke structures and which is not captured in definition 2.11:

Proposition 3.1: Let $f{:}<S_1,R_1,V_1>\rightarrow<S_2,R_2,V_2>$ be a morphism of Σ-models and A a formula over Σ. For any $s\in S_1$, $<S_1,R_1,V_1>\ ^{a}_{s}$ A□□iff□□$<S_2,R_2,V_2>\ ^{a}_{f(s)}$ A. ▮

This property, taken from [Goldblatt 87], cannot be directly formulated for β-institutions. It admits, however, the following corollary (which in [van Benthem 84] is attributed to Segerberg):

Corollary 3.2: Let $f{:}<S_1,R_1,V_1>\rightarrow<S_2,R_2,V_2>$ be a morphism of Σ-models. For any formula A over Σ, $<S_1,R_1,V_1>\ ^{a}$ A□□if□ $<S_2,R_2,V_2>\ ^{a}$ A. ▮

This property is now generalisable to β-institutions:

Definition 3.3: We say that a β-institution has the *p-property* iff it satisfies the following condition: for every signature Σ, Σ-models $<\sigma_1,B_1>$ and $<\sigma_2,B_2>$, and morphism f: $<\sigma_1,B_1>\rightarrow<\sigma_2,B_2>$, if $<\sigma_2,B_2>\models A$ then $<\sigma_1,B_1>\models A$ for every $A\in Sen(\Sigma)$. ▮

It is this property that requires conservativeness with respect to nondeterminism of p-morphisms as defined in 2.3. This is intuitive if we think in terms of process behaviour and, using the terminology put forward in [Kuiper 89], distinguish between *required* nondeterminism and *allowed* nondeterminism. Required nondeterminism is nondeterminism which any implementation must possess because the specification requires so. It translates into branching of the accessibility relation. Allowed nondeterminism is the freedom that is left to the implementor due to the vagueness of the specification, which results in the existence of more than one model to the specification.

The p-property implies that if a process P_1 simulates a process P_2, then any property satisfied by P_2 is also satisfied by P_1. Hence, because required nondeterminism is a property, it cannot decrease from P_2 to P_1: if behaviour morphisms allowed for further branching in P_2, then P_2 would show more properties (the existence of more alternative behaviours) than P_1.

This means that if we are working over a process model BEHA which is rich enough to model required nondeterminism (e.g. labelled transition systems as defined in [Sassone et al 93]) and we want to define a β-institution over that process model in such a way that it has the p-property (we shall see that there are good reasons for doing so), then we have to select a subcategory of BEHA with the same objects but only those morphisms that do not allow for nondeterminism to be increased.

If, however, the logical language is not rich enough for requiring nondeterminism (as, for instance, in [Sernadas et al 94] where linear temporal logic is interpreted over labelled transition systems), then there is no need for moving to a subcategory. (The linear tense operators are not interpreted directly over a transition system but over the sets of runs generated from that transition system.) The p-property is, indeed, a relationship between syntax and semantics, not a property of models alone.

The interest of the p-property lies in the acquired structural properties of the category of theories (specifications) of the β-institution. We discuss now some of these properties.

Definition 3.4: The category THEO of the theories of a β-institution is defined as for its corresponding institution. The objects of THEO are pairs $<\Sigma,\Gamma>$ where $\Gamma\subseteq Sen(\Sigma)$ is closed under consequence, i.e. $A\in\Gamma$ whenever, for every $M\in Mod(\Sigma)$, $M^{a}{}_{\Sigma}\Gamma$ implies $M^{a}{}_{\Sigma}A$. A theory morphism $\sigma:<\Sigma_1,\Gamma_1>\rightarrow<\Sigma_2,\Gamma_2>$ is a signature morphism $\sigma:\Sigma_1\rightarrow\Sigma_2$ such that $Sen(\sigma)(\Gamma_1)\subseteq Sen(\sigma)(\Gamma_2)$. ∎

A particularly interesting class of β-institutions consists of those for which the functor Sign:BEHA→SIGN^{op} can be lifted to a functor Spec:BEHA→THEO^{op} such that Mod itself can be lifted to THEO as Mod(T)=(Spec↓T). The intuition for this interest is that we should be able to associate with every behaviour B a canonical specification Spec(B) so that every morphism T→Spec(B) identifies T as a specification of B and, dually, B as a realisation of T. In this case, the models of T correspond to all possible refinements of T into programs. Identifying BEHA with (the behaviours of) programs (or processes), this corresponds to the idea that programs can themselves be regarded as specifications, and that the models of a specification can be identified with the programs which, in some sense, "complete" the specification.

Notice that the relationships between processes and specifications which can be found in the literature (e.g. [Hennessy and Milner 85, Pnueli 85, Graf and Sifakis 89]) are also based on the existence of a way of associating a set of formulas Spec(B) with every behaviour B. Although, in these approaches, the main interest is in discussing the relationship between the equivalence relation induced by Spec and the one inherent from BEHA, Pnueli already suggests in [Pnueli 85] that the induced orderings be compared as well. The functoriality of Spec is a kind of correctness requirement on these orderings, namely that if B_1 simulates B_2 then Spec(B_2) "implies" Spec(B_1). We shall further discuss other requirements on Spec below.

Although it is always possible to define a mapping that associates a theory Spec(B) with every behaviour B – the obvious candidate assigns to Spec(B) the signature Sign(B) and theorems $\{A\in Sen(Sign(B)): <id,B>\vdash A\}$ – it is not always possible to extend it to a functor that lifts Sign, i.e. such that Spec(h:B→B')=Sign(h). In order to motivate why this is so, assume that we have a functor Spec:BEHA→THEO^{op} and consider a morphism h:B'→B. Because Spec is a functor, we have Spec(h):Spec(B)→Spec(B'). Hence, if B is a model of a specification T, i.e. if we have a morphism σ:T→Spec(B), then B' is also a model of T through the morphism σ;Spec(h). In particular this implies that, for every signature Σ and $A\in Sen(\Sigma)$, if $<\sigma,B>^{a}{}_{\Sigma}A$ then $<\sigma;Sign(h),B'>^{a}{}_{\Sigma}A$. That is, the β-institution is required to have the p-property.

Proposition 3.5: If the satisfaction relation has the p-property, then the mapping Spec defined by

- for every B:BEHA, $Spec(B)=<Sign(B),\{A\in Sen(Sign(B)): <id,B>^{a}A\}>$
- for every h:B→B', Spec(h:B→B')=Sign(h)

is a functor BEHA→THEO^{op} .

proof: it is sufficient to prove that Spec(h:B→B') is, indeed, a theory morphism Spec(B')→Spec(B). Let A'∈Spec(B'). Then, $<id,B'>^{a}A'$. The p-property implies that

<Sign(h),B>ªA'. The satisfaction condition then implies <id,B>ªSen(Sign(h))(A'). ∎

Definition 3.6: Given a theory T=<Σ,Γ>, Mod(T) is defined as for the corresponding institution, i.e. as the full subcategory of Mod(Σ) which consists of the Σ-models which satisfy T. ∎

Proposition 3.7: If the satisfaction relation has the p-property, then Mod(T)=(Spec↓T).

proof:

1. Let <σ,B>∈Mod(T), i.e. <σ,B>⊨A for every A∈T. Then, by the satisfaction condition, <id,B>ªσ(A) for every A∈T, i.e. σ(T)∈Spec(B), which means that σ is a theory morphism T→Spec(B).
2. Let T→Spec(B) and A∈T. Then, because σ is a theory morphism, σ(A)∈Spec(B), i.e. <id,B>ªσ(A). The satisfaction condition then implies that <σ,B>ªA. Hence, <σ,B>∈Mod(T). ∎

Corollary 3.8: If the satisfaction relation has the p-property, Mod=(Spec↓_) ∎

It is well known that (Spec↓_) is a pullback-preserving functor and, hence, the p-property leads to an exact institution [Meseguer 89]. This is a particularly important property on the modularity allowed by the formalism because it tells us that the models of a composite specification (taken as a colimit) are all the possible compositions (taken as limits) of the models of the components. That is, in order to develop a system that satisfies a composite specification, we may put together any models of the component specifications.

Corollary 3.9: If a β-institution has the p-property then it is exact. ∎

As already mentioned, the p-property has an interesting interpretation from the point of view of behaviour specification. Intuitively, this property corresponds to an "open semantics" of behaviour: any behaviour that simulates a model is itself a model; or: if a component of a system is a model of a specification, then the system itself is a model for that specification. We say that this corresponds to an open semantics of behaviour because it implies that if one is satisfied with a process (because it meets a certain specification), then one must be satisfied with any other process which simulates it; it is not possible to specify a process in isolation.

By the way it is defined, Spec satisfies one of the compatibility requirements for *expressivity* of the logic for the domain BEHA as defined by Pnueli [Pnueli 85]. Adapting to our context, namely to the use of theories instead of formulae as the target of Spec, this is the property

<σ,B>ªA iff σ(A)∈Spec(B)

The other requirement, that B_1 and B_2 be "equivalent" (isomorphic) whenever B_1 is a model of Spec(B_2) is, however, a little awkward in our setting because morphisms provide a much finer spectrum of relationships between behaviours. A formulation which we think is in the spirit of Pnueli's requirement is:

if <σ:Sign(B_2)→Sign(B_1),B_1>∈Mod(Spec(B_2))

then there is $h:B_1 \to B_2$ such that Sign(h)=σ.

This says that B_1 is related to B_2 whenever B_1 is a model of Spec(B_2). Because $<\sigma$:Sign(B_2)→Sign(B_1),B_1>∈Mod(Spec(B_2)) iff σ:Spec(B_2)→Spec(B_1), this requirement corresponds to fullness of Spec. It is also in the spirit of the notion of expressiveness proposed by Pnueli that Spec be faithful, as it corresponds to saying that distinctions between behaviours are not ignored by their specifications.

Definition 3.10: A β-institution is said to be *expressive*□iff it has the p-property and Spec is full and faithful. ▮

The classification on the relationship between the logic and the process models put forward in [Hennessy and Milner 85] concerns just the equivalence relations of the process model and the one induced on them by the logic:

Definition 3.11: A β-institution is said to be *adequate* iff it has the p-property and Spec reflects isomorphisms, i.e. if whenever h is a behaviour morphism such that Spec(h) is a theory isomorphism, then h is itself an isomorphism. ▮

Notice that functors always preserve isomorphisms. Hence, adequacy means that two behaviours are isomorphic iff their specifications are isomorphic.

Because faithful and full functors reflect isomorphisms, the relationship between expressiveness and adequacy is as in [Pnueli 85]:

Proposition 3.12: Every expressive β-institution is adequate. ▮

The problems of characterising the equivalences imposed on models by temporal logics and of developing logics which are expressive for given process models are well documented in the literature (e.g. [Goltz et al 92], [Graf and Sifakis 89]).

4 Programs, adjointness and fibrations

We have seen that, under the p-property, the models of a specification T can be associated with all the possible refinements of T into specifications which correspond to "programs". Intuitively, a specification T should correspond to a program when it has a "canonical" model. Because models of a specification are closed under morphisms (simulation) this corresponds to the existence of a behaviour [T] such that every other model is a simulation of [T]. In categorial terms:

Definition 4.1: We say that a specification T is a program iff Mod(T) has a terminal object (which we denote by [T]). ▮

A typical β-institution, i.e. a typical formalism for behaviour specification, will be such that not every specification is a program. Indeed, the idea of working with specifications is to be able to start from "loose" requirements which do not identify a single program, and refine them, by adding detail, until a "program" is obtained, i.e. a specification to which a program can be assigned.

A β-institution which corresponds to a formalism for "programming", i.e. such that every specification has a canonical realisation as a program, is then an institution with a terminal semantics:

Definition 4.2: A β-institution is said to have a terminal semantics iff, for every T:THEO, Mod(T) has a terminal object. ∎

If a β-institution has a terminal semantics, then a functor Beha: THEO→$BEHA^{op}$ can be defined which returns the behaviour that corresponds to every program (theory):

Proposition 4.3: Consider a β-institution with terminal semantics. The mapping Beha defined by

- for every P:THEO, Beha(P) is the behaviour of "the" terminal object of Mod(P);
- for every σ:P_1→P_2, Beha(σ) is the unique morphism Beha(P_2)→Beha(P_1) that results from Beha(P_1) being terminal in Mod(P_1) and Beha(P_2), through the reduct functor associated with σ, being a model of P_1;

is a functor THEO→$PROG^{op}$.

proof: We shall just detail the construction of Beha(σ). Let <μ_i,Beha(P_i)> be terminal in Mod(P_i). By the reduct functor associated with σ, <σ;μ_2,Beha(P_2)> is a model of P_1. Hence, because <μ_1,Beha(P_1)> is terminal in Mod(P_1), there is a unique h:Beha(P_2)→Beha(P_1) such that σ;μ_2;Sign(h)=σ;μ_1. We take Beha(σ) to be h. ∎

Thanks to the p-property, β-institutions with terminal semantics have very strong structural properties:

Proposition 4.4: A β-institution with the p-property has a terminal semantics iff the functor Spec: BEHA→$THEO^{op}$ has a right adjoint.

proof: This is a consequence of a general result of category theory – a functor F:C→D has a right adjoint iff for every object B of D there is a terminal object in the category (F↓B). See [Crole 93] for a proof. ∎

Corollary 4.5: A β-institution with the p-property has a terminal semantics iff Beha is a right adjoint of Spec. ∎

These results on adjointness and exactness (3.9), which generalise [Fiadeiro and Costa 93], are one of the motivating forces behind the construction of β-institutions, as they have always been perceived as important structural properties for behaviour specification. Notice that this adjoint situation does not follow from the usual Galois correspondence of logics because we are dealing with models and not sets of models.

The existence of this adjointness implies that the model functor Mod: $THEO^{op}$→CAT can be factorised as Beha;(BEHA↓_) which characterises the β-institution as follows:

Corollary 4.6: A β-institution with the p-property has a terminal semantics iff it is categorical, in the sense of [Meseguer 89], on BEHA. ∎

That is, categoricity characterises "programmability".

As an example of a β-institution which has terminal semantics we have linear temporal logic with trace-based semantics as defined in [Fiadeiro and Costa 93]. The construction and proof of the existence of the adjunction can also be found in that paper. The intuitive interpretation of this result is the well known fact that linear temporal logic does not distinguish between underspecification and nondeterminism! That is, the existence of several models of a linear temporal theory can always be attributed to nondeterminism. We shall return to this distinction later on.

Expressive β-institutions, as defined in 3.10, satisfy a stronger property:

Proposition 4.7: If an expressive β-institution has a terminal semantics, every behaviour B is isomorphic to Beha(Spec(B)), i.e. Spec(B) is a program whose behaviour is B.
proof: This is a consequence of the fact that, in an adjunction, the left adjoint is faithful and full iff the unit is a natural isomorphism. ▮

The proof shows that the converse is also true: if a β-institution with the p-property has a terminal semantics, then it is expressive if B≈Beha(Spec(B)) for every B:BEHA.

Proposition 4.7 is somewhat surprising as it might be expected that the specification of a behaviour be a program! It is certainly true that B is a model of Spec(B), but it may not be the maximal one if the logic and the process model are not fully compatible. For instance, the process model may assign more structure to behaviours than what is needed for interpreting the logic (e.g. transition systems interpreting linear temporal logic as in [Sernadas et al 94]). In this case, a subcategory of BEHA would be needed where, by removing some of the morphisms, the notion of equivalence is coarser.

We shall now attempt at further characterising the specifications (theories) that are programs, i.e. that have terminal models. We are going to provide that characterisation in the context of additional structural properties of the β-institution. These structural properties are of two kinds.

First we are going to assume that the functor Sign: BEHA→SIGN^{op} can be factorised as BEHA□^{Alph},——→□ALPH□^{F},——→□SIGN^{op}. The category ALPH captures the notion of alphabet. We shall further require that Alph be faithful, making BEHA concrete over ALPH. All models presented in [Sassone et al 93] are indeed concrete categories. In the case of the trace-based model that we defined in 2.4 and 2.6, alphabets are pointed sets (of events).

Definition 4.8: We say that a β-institution is factorised over a category ALPH iff the functor Sign: BEHA→ SIGN^{op} can be factorised as BEHA□^{Alph},——→□ALPH□^{F},——→□SIGN^{op} where Alph is faithful. ▮

We did not define β-institutions using this factorisation from the beginning because results so far did not depend on it. Factorised β-institutions also have the advantage of exhibiting a pleasing symmetry between syntax and semantics, both being given through faithful functors (concrete categories) THEO→SIGN and BEHA→ALPH. The relationship between syntax and semantics is then left to the functor F:ALPH→SIGN^{op}.

The first structural property that we shall require is that F:ALPH→SIGN^{op} admits a right adjoint. The motivation for this condition is that the "distance" between specifications and models should lie in the axioms of the specification and the behaviours of the model, not in signatures and alphabets. Hence, the difference between programs and specifications should not be reflected in the signatures but in the axiomatisations: a specification is refined into a program essentially by addition of detail (axioms). (The fact that we work with an adjunction and not an equivalence means that there is still some transformation to be performed on the signatures but that this tranformation is well determined.) In the case of linear temporal logic over trace-based models of behaviour as defined in [Fiadeiro and Costa 93], the adjunction between $SET_{\perp}$ and SET^{op} is established through the contravariant power-set functor. That is, events are taken as sets of action propositions.

The right adjoint to the functor F, which we shall denote by G, fixes the alphabet of the terminal model of a specification T, if it exists. The second structural property will allow us to determine the behaviour of the terminal model among the behaviours over the chosen alphabet. That is, it will allow us to choose the terminal behaviour over the subcategory of models whose alphabet is the one determined by the functor G.

Definition 4.9: For every theory T, we define $Mod^*(T)$ to be the (full) subcategory of Mod(T) whose objects are of the form $<\varepsilon_\Sigma:\Sigma \rightarrow Sign(B), B>$ where Σ is the signature of T, $\varepsilon_\Sigma:\Sigma \rightarrow F(G(\Sigma))$ is the co-unit of the adjunction, and $G(\Sigma)$ is the alphabet of B. ∎

The category $Mod^*(T)$ is, therefore, defined over the fibre $Alph^{-1}(G(\Sigma))$, i.e. over the behaviours whose alphabet is $G(\Sigma)$. Because the refinement morphism is now fixed to ε_Σ, we have the following characterisation for $Mod^*(T)$:

Proposition 4.10: $Mod^*(T)$ is isomorphic to a subcategory of $Alph^{-1}(G(\Sigma))$: the full subcategory whose objects are the $G(\Sigma)$-behaviours B such that $<\varepsilon_\Sigma,B>\models A$ for every $A\in T$. ∎

Because the typical behaviour categories with which we deal are fibre-small, we may assume that $Mod^*(T)$ is itself small. Moreover, because fibres are partial orders, $Mod^*(T)$ is also a partial order.

The second structural property requires Alph:BEHA→ALPH to be a split cofibration [Barr and Wells 90]. Intuitively, this property implies that, given an alphabet morphism $h:\alpha \rightarrow \alpha'$, and a behaviour B such that $Alph(B)=\alpha$, we can find its "image" h(B), i.e. a behaviour whose alphabet is α' and is "closest" to B. In this case, closest means that: (1) h is itself a morphism between B and h(B); (2) for every other behaviour B' over α' for which h is a morphism from B to B', there is a unique morphism h(B)→B' which is projected to $id_{\alpha'}$.

Proposition 4.11: A functor $\_^\bullet$: $Mod(\Sigma) \rightarrow Mod^*(\Sigma)$ is defined as follows:

- $<\sigma:\Sigma \rightarrow Sign(B),B>^\bullet$ is $<\varepsilon_\Sigma:\Sigma \rightarrow F(G(\Sigma)), \sigma^+(B)>$ where ε_Σ is the co-unit of the adjunction and σ^+ is the unique morphism $Alph(B) \rightarrow G(\Sigma)$ such that $(\varepsilon_\Sigma;F(\sigma^+)=\sigma)$;
- given models $<\sigma,B>$ and $<\sigma',B'>$ and $h:B \rightarrow B'$ such that $\sigma=\sigma';F(h)$, $h^\bullet$ is the unique morphism $\sigma^+(B) \rightarrow \sigma'^+(B')$ such that $(h;\sigma'^+)=\sigma^+;h^\bullet$.

This functor is a left-adjoint of the inclusion functor of $Mod^*(\Sigma)$ into $Mod(\Sigma)$. ∎

This functor translates every model to another model over the canonical alphabet $G(\Sigma)$.

Corollary 4.12: $Mod^*(\Sigma)$ is a reflective sub-category of $Mod(\Sigma)$. ∎

In order to lift the functor $(\_^\bullet)$ from signatures to theories, i.e. to obtain, for every theory

T, Mod*(T) as a (full) reflective subcategory of Mod(T), we have to require that the satisfaction relation be compatible with the cofibration:

Definition 4.13: A factorised β-institution is said to be fibred iff

- Alph:BEHA$\rightarrow$ALPH is a split cofibration;
- for every signature Σ, $A\in Sen(\Sigma)$, $\sigma:\Sigma\rightarrow F(\alpha)$ and morphism $h:Alph(B)\rightarrow\alpha$, $\langle\sigma;F(h),B\rangle\models A$ implies $\langle\sigma,h(B)\rangle\models A$. ▮

Notice that the satisfaction condition implies that we have an equivalence.

As an example of a fibred β-institution we have linear temporal logic over PROC as defined in [Fiadeiro and Costa 93]. In this case, Alph is even topological. Another example is given in [Sernadas et al 94] for labelled transition systems.

Proposition 4.14: In a fibred factorised β-institution, if $\langle\sigma:\Sigma\rightarrow Sign(B),B\rangle$ is a model of $T=\langle\Sigma,\Gamma\rangle$, then $\langle\sigma:\Sigma\rightarrow Sign(B),B\rangle^\bullet$ is also a model of T.
proof: Let $A\in\Gamma$. We want to prove that $\langle\varepsilon_\Sigma,\sigma^+(B)\rangle\models A$. Because the satisfaction relation is compatible with the cofibration, it is sufficient to prove that $\langle\varepsilon_\Sigma;F(\sigma^+),B\rangle\models A$. But, by definition, $\varepsilon_\Sigma;F(\sigma^+)=\sigma$, and $\langle\sigma,B\rangle\models A$ because $\langle\sigma:\Sigma\rightarrow Sign(B),B\rangle$ is a model of T. ▮

Corollary 4.15: In a fibred factorised β-institution, Mod*(T) can be characterised as a (full) reflective subcategory of Mod(T). ▮

Suppose now that Mod*(T) has a terminal object T*. Let $\langle\sigma,B\rangle$ be a model of T (which implies that $\langle\varepsilon_\Sigma,\sigma^+(B)\rangle$ is also a model of T). Because T* is terminal in Mod*(T) there is a unique morphism $h:\sigma^+(B)\rightarrow T^*$ such that $F(h)=id_\Sigma$ where Σ is the signature of T. The composition $\sigma^+;h$ gives us a morphism $B\rightarrow T^*$. Moreover, $\varepsilon_\Sigma;F(\sigma^+;h)=\varepsilon_\Sigma;(F(\sigma^+);F(h))=\varepsilon_\Sigma;F(\sigma^+)=\sigma$. On the other hand, suppose we have another morphism $g:B\rightarrow T^*$ such that $\varepsilon_\Sigma;F(g)=\sigma$. Then, necessarily, g=h because of the adjunction. Hence, $\langle\varepsilon_\Sigma,T^*\rangle$ is a terminal object of Mod(T).

Proposition 4.16: In a fibred factorised β-institution, if Mod*(T) has a terminal object than so does Mod(T) and they coincide. ▮

We have thus reduced the problem of finding a terminal object for Mod(T) to determining a terminal object for Mod*(T).

Being small and a partial order, a sufficient condition for Mod*(T) to admit a terminal model is that Mod*(T) admits (small) colimits. In fact, this is equivalent to admitting small coproducts. Indeed, because Mod*(T) is a partial order, the uniqueness condition for the morphisms into the colimit of the whole category is automatically satisfied and, hence, its colimit is a terminal object.

Proposition 4.17: In a fibred factorised β-institution, a theory T has a terminal model if Mod*(T) is closed under small coproducts. ▮

This condition, i.e. that Mod*(T) admits arbitrary (but small) coproducts, has an intuitive interpretation. Indeed, like products correspond to parallel composition, coproducts correspond, in most process models, to *nondeterministic choice*. Hence, the condition requires that Mod*(T) be closed under choice, i.e. that any arbitrary choice between models of T be still a model of T. Intuitively, this means that we obtain a program when we have eliminated all the underspecification, the remaining models being able to be reduced to one via a choice operator. It is important to notice that this choice operator is internalising part of what we called *allowed nondeterminism* in section 3, i.e. that nondeterminism which, as opposed to *required nondeterminism*, originates in the vagueness of the specification and not in specific requirements of the specification for nondeterministic behaviour.

It is interesting to note that, for linear temporal logic, programs coincide with theories because the logic does not distinguish between nondeterminism and underspecification. The same, however, is not true, for instance, for branching time logic or the μ-calculus, which allow for required nondeterminism but, depending on how their model theory is defined, may or may not internalise allowed nondeterminism. We are currently working on the categorial characterisations of the two kinds of nondeterminism and their relationships with theories.

5 Concluding remarks

In this paper, we showed how an abstract and general account of the relationships between process models and logics for the specification of reactive system behaviour can be given in the context of a specialisation of the notion of institution. The notion of β-institution was proposed taking into account the structural properties of traditional Kripke semantics of modal logics, the recent work on categorial models of concurrency [Sassone et al 93], and our own past work on the modularisation of temporal specifications [Fiadeiro and Maibaum 92] and on relating logics and models of reactive behaviour [Fiadeiro et al 93, Fiadeiro and Costa 93]. We were thus able to establish abstract characterisation of some of the concepts and techniques that have been proposed for reactive system development, and lay the foundations for future work.

As an example of the expressive power of this framework, we showed how concepts that have been put forward for relating behavioural and logical approaches based on notions of observational equivalence [Hennessy and Milner 85, Pnueli 85, Graf and Sifakis 87] can be formalised and refined in the context of β-institutions. As part of future research, we intend to explore this new added expressive power in the generalisation of these studies, namely in a classification of logics of concurrency in the spirit of [Goltz et al 92].

As another application, we explored the structural properties of the category of theories of β-institutions to characterise exactness and the existence of adjoint situations between theories and models (not sets of models as in the standard Galois correspondence). Exactness is a property of β-institutions that satisfy an important structural property (the p-property of Kripke structures). The existence of such adjoint situations was shown to be equivalent to the availability of terminal models for specifications (meaning that every specification denotes a canonical program). We also showed how the existence of terminal models of specifications is related to the elimination of underspecification in favour of nondeterminism. Further work is necessary in order to characterise the difference between required and allowed nondeterminism (as in [Kuiper 89]) in the context of the adopted categorial approach. We also intend to explore other aspects of β-institutions, namely mappings between β-institutions as a means of establishing a proof-theoretic counterpart to the classification mechanisms developed in [Sassone et al 93] for concurrency models.

It is only fair to say that other applications of algebraic techniques to concurrency do exist, notably the work developed in Genova [e.g. Reggio 91] around entity institutions. The application of the institutional framework that these authors have developed is different from ours in that their starting point is an algebraic specification of the Kripke structures of the logic, thus developing an ontology (dynamic entities) which in our approach is missing: whereas in entity institutions "processes" are denoted by terms, in β-institutions they are denoted by theories. This is also the difference between exogeneous and endogeneous applications of logic to system modelling. It seems worth developing a more formal relationship between the two approaches, as they seem to complement each other, entity institutions providing a way of dealing explicitly with the external structure (configuration) of complex systems, and β-institutions providing a more abstract framework in which to discuss the structural properties of specifications and their underlying formalisms.

Acknowledgements

We are grateful to Tom Maibaum for many fruitful discussions.

References

[Barr and Wells 90]
M.Barr and C.Well, *Category Theory for Computing Science*, Prentice-Hall 1990

[Barringer et al 85]
H.Barringer, R.Kuiper and A.Pnueli, "A Compositional Temporal Approach to a CSP-like Language", in E.Neuhold and G.Chroust (eds) *Formal Models in Programming*, North-Holland 1985, 207-227

[Costa *et al* 92]
J.F.Costa, A.Sernadas, C.Sernadas and H.-D.Ehrich, "Object Interaction", in *Proc. MFCS'92*, LNCS 629, Springer-Verlag 1992, 200-208.

[Crole 93]
R.Crole, *Categories for Types,* Cambridge University Press 1993

[Dionísio 91]
F.M.Dionísio, *Um Modelo e Submodelos Categoriais de Processos Concorrentes*, MSc. Thesis, Dept.Mathematics, Fac.Engineering, Technical University of Lisbon, 1991.

[Ehrich et al 91]
H.-D.Ehrich, J.Goguen and A.Sernadas, "A Categorial Theory of Objects as Observed Processes", in J.deBakker, W.deRoever and G.Rozenberg (eds) *Foundations of Object-Oriented Languages*, LNCS 489, Springer Verlag 1991, 203-228.

[Fiadeiro and Costa 93]
J.Fiadeiro and J.F.Costa, *Mirror, mirror in my hand: a topological adjunction between temporal theories and processes*, Research Report, DI-FCUL, March 1993.

[Fiadeiro and Maibaum 91]
J.Fiadeiro and T.Maibaum, "Describing, Structuring, and Implementing Objects", in J.deBakker, W.deRoever and G.Rozenberg (eds) *Foundations of Object-Oriented Languages*, LNCS 489, Springer-Verlag 1991, 274-310.

[Fiadeiro and Maibaum 92]

J.Fiadeiro and T.Maibaum, "Temporal Theories as Modularisation Units for Concurrent System Specification", *Formal Aspects of Computing* 4(3), 1992, 239-272.

[Fiadeiro et al 93]
J.Fiadeiro, J.F.Costa, A.Sernadas and T.Maibaum, "Process Semantics of Temporal Logic Specification", in M.Bidoit and C.Choppy (eds) *Recent Trends in Data Type Specification*, LNCS 655, Springer-Verlag 1993, 236-253.

[Goguen and Burstall 92]
J.Goguen and R.Burstall, "Institutions: Abstract Model Theory for Specification and Programming", *Journal of the ACM* 39(1), 1992, 95-146

[Goguen and Ginali 78]
J.Goguen and S.Ginali, "A Categorical Approach to General Systems Theory", in G.Klir (ed) *Applied General Systems Research,* Plenum 1978, 257-270.

[Goldblatt 87]□
R.Goldblatt, *Logics of Time and Computation*, CSLI 1987.

[Goltz et al 92]
U.Goltz, R.Kuiper, W.Penczek, "Propositional Temporal Logics and Equivalences", in W.Cleaveland (ed) *CONCUR'92*, LNCS 630, Springer-Verlag 1992, 222-236.

[Graf and Sifakis 89]
S.Graf and J.Sifakis, "An Expressive Logic for a process Algebra with Silent Actions", in B.Banieqbal, H.Barringer and A.Pnueli (eds) *Temporal Logic in Specification*, LNCS 398, Springer-Verlag 1989, 44-61.

[Hennessy and Milner 85]
M.Hennessy and R.Milner, "Algebraic Laws for Nondeterminism and Concurrency", *Journal of the ACM* 32(1), 1985, 137-161

[Kuiper 89]
R.Kuiper, "Enforcing Nondeterminism via Linear Temporal Logic Specifications using Hiding", in B.Banieqbal, H.Barringer and A.Pnueli (eds) *Temporal Logic in Specification*, LNCS 398, Springer-Verlag 1989, 295-303.

[Meseguer 89]
J.Meseguer, "General Logics", in H.-D.Ebbinghaus et al (eds) *Logic Colloquium 87,* North-Holland 1989.

[Pnueli 77]
A.Pnueli, "The Temporal Logic of Programs", in *Proc 18th Annual Symposium on Foundations of Computer Science,* IEEE 1977, 45-57.

[Pnueli 85]
A.Pnueli, "Linear and Branching Structures in the Semantics and Logics of Reactive Systems", in *ICALP'85*, LNCS 194, Springer-Verlag 1985, 15-32.

[Reggio 91]
G.Reggio, "Entities: an Institution for Dynamic Systems", in H.Ehrig, K.Jankte, F.Orejas and H.Reichel (eds) *Recent Trends in Data Type Specification*, LNCS 534, Springer-Verlag 1991, 244-265

[Sassone et al 93]
V.Sassone, M.Nielsen and G.Winskel , "A Classification of Models for Concurrency", in E.Best (ed) *CONCUR'93*, LNCS 715, Springer-Verlag 1993, 82-96.

[Sernadas et al 94]
A.Sernadas, J.F.Costa and C.Sernadas, "An Institution of Object Behaviour", in H.Ehrig and F.Orejas (eds) *Recent Trends in Data Type Specification*, LNCS 785, Springer-Verlag 1994.

[van Benthem 84]
J.van Benthem, "Correspondence Theory", in D.Gabbay and F.Guenthner (eds) *Handbook of Philosphical Logic* vol II, Reidel 1984, 167-247.

[Wolper 89]
P.Wolper, "On the Relation of Programs and Computations to Models of Temporal Logic", in B.Banieqbal, H.Barringer and A.Pnueli (eds) *Temporal Logic*

in Specification, LNCS 398, Springer-Verlag 1989, 75-123.