An analysis of total correctness refinement models for partial relation semantics II

MOSHE DEUTSCH, MARTIN C. HENSON, *Department of Computer Science, University of Essex, UK*

Abstract

This is the second in a series of papers devoted to the thorough investigation of (total correctness) refinement based on an underlying partial relational model. This paper investigates *operation refinement* and *data-refinement* based on a *weakest precondition* interpretation for specifications whose semantics is given by *partial* relations. We consider three refinement theories based on a weakest precondition interpretation for partial relation semantics: an operation refinement theory, and theories characterising data-refinement with forward and backward simulations. We show that each of these is equivalent to a (corresponding) model-theoretic refinement theory that is based on the standard approach involving relational completion operators. In addition, we demonstrate that each of the three is also equivalent to a (corresponding) proof-theoretic notion of refinement.

1 Introduction

This paper investigates total correctness[1] *operation refinement* and *data-refinement* based on a *weakest precondition* interpretation for specifications whose semantics is given by *partial* relations. The most well-known example of such a semantics is that for Z, in which schemas denote sets of bindings, establishing a partial relation between before and after states.

We consider three refinement theories based on a weakest precondition interpretation for partial relation semantics: an operation refinement theory, a theory characterising data-refinement with forward simulation and a theory characterising data-refinement with backward simulation. We show that each of these is equivalent to a (corresponding) model-theoretic refinement theory based on the standard approach involving relational completion operators (discussed in, for example, [41, 11] and investigated in detail in [12, 13]). In addition, we demonstrate that each one of them is also equivalent to a (corresponding) proof-theoretic notion characterising refinement in a very natural manner. This serves to illuminate not only the weakest precondition approach for refinement but also the standard model-theoretic account based on relations. In particular we wish to answer a basic question concerning weakest precondition data-refinement based on partial relation semantics: *how do simulations combine with postconditions?*

We begin by developing a weakest precondition interpretation for operations (section 2). As in [27, 28], this is based directly on an underlying partial relation seman-

[1] Now that we have established that we are dealing with total correctness, we will drop this qualification: unless we specify otherwise, we are interested only in total correctness refinement.

tics. We then introduce an operation refinement theory based on this interpretation and prove that it is equivalent to two other characterisations of refinement (section 3). The first is purely proof-theoretic and captures refinement directly in terms of the language, the relationship between the data types involved, and the concept of precondition. It is a more abstract, less constructive notion than the second characterisastion which is a model-theoretic notion based on the *relational completion* of the underlying partial semantics (see appendix B). This is sometimes called the *lifted-totalisation* and involves an additional distinguished element called *bottom*, written $\bot$, which is not needed in the proof-theoretic approach.
We then introduce the notion of *data simulation* (underlying the forward and backward simulation data-refinement techniques) and the *lifted* simulations used in model-theoretic data-refinement (section 4). These prepare the way for an investigation of data-refinement in sections 5 and 6 which generalises the investigation of section 3 to forward and backward simulation data-refinement (respectively). We define, in each case, a refinement theory founded on the weakest precondition semantics and demonstrate that each is equivalent to an appropriate model-theoretic and proof-theoretic data-refinement characterisation.
Our investigation takes place in $\mathcal{Z}_C$, the logic for Z reported in [25] and a simple *conservative extension* $\mathcal{Z}_C^\bot$ [12] which incorporates $\bot$ into the types of $\mathcal{Z}_C$ (we summarise this, and additional notational conventions, for convenience in appendix A). This allows us to work with Z schemas as easily as with abstract relations and therefore to make an explicit link with the most well-known specification language which is based on partial relation semantics. Having said this, it should be noted that *nothing we show here is specifically confined to Z*: it is merely a convenient linguistic vehicle for state-based specification.
We employ a novel technique of rendering *all* the theories of refinement that we consider as sets of introduction and elimination rules. This leads to a uniform and simple method for proving the various equivalence results in the sequel. It contrasts with the more semantic techniques employed in [8]. We shall have more to say about this in section 3.4.

2 Weakest precondition semantics

2.1 Context

The notion of weakest precondition was introduced by Dijkstra [14], further evolved in [15] and used as an underlying semantics for a variety of programming languages (*e.g.* [19] and [22]). It was also adopted as the elementary semantics for specification statements [31] which underly refinement calculus as a unified language for developing programs from specifications (*e.g.* [32]). Work on a weakest precondition semantics for Z was undertaken in [5, 6] and was developed for the purpose of supporting on ZRC: linking Z with refinement calculus (*e.g.* [7]). This framework is founded on predicate transformers and compared with, rather than based on, the relational semantics underlying Z. Josephs [27] developed a weakest precondition semantics for operation schemas based directly on the partial relation semantics for Z. Operation refinement and data-refinement (the forward simulation case) in this framework are discussed in [27] and [28] (respectively).

In this section, we develop a weakest precondition interpretation for operation schemas in $\mathcal{Z}_C$. Following [27, 28], our interpretation is based on the relational semantics of Z, rather than predicate transformers as used in [5, 6]. We generalise Joseph's work by introducing both operation refinement and (forward and backward simulation) data-refinement theories based on this interpretation and also by investigating their relationship with other refinement theories, including some introduced in our earlier work [12, 13]. This is our answer to Dijkstra and Scholten [15, p.126], who favour predicate transformers over the relational approach: "*either no one trying to apply the relational calculus in this area mastered it well enough, or ... the relational calculus needs a few notational revisions before it can be considered a workable tool*". One of the aims of the series of papers, of which this is the second, that we are preparing is to thoroughly investigate the relational approach, both in its relationship to the predicate transformer/weakest precondition approach and in terms of its expressive power.

2.2 Technical preliminaries

We begin by introducing a notion of *postcondition* to complement the standard idea of preconditions (see appendix A, section A.2).

DEFINITION 2.1

$$Post\ U\ z_0 =_{df} \{z_1' \mid z_0 \star z_1' \in U\}$$

Note that this introduces a set, rather than a predicate.

PROPOSITION 2.2
The following introduction and elimination rules are derivable for postcondition:

$$\frac{t_0 \star t_1' \in U}{t_1' \in Post\ U\ t_0}\ (Post^+) \qquad \frac{t_1' \in Post\ U\ t_0}{t_0 \star t_1' \in U}\ (Post^-)$$

■

With this in place we can introduce the weakest precondition interpretation of an operation schema. Again, the specified postcondition (C in the definition below) is expressed as a set rather than as a predicate.

DEFINITION 2.3

$$wp\ U\ C =_{df} \{z \mid Pre\ U\ z \wedge Post\ U\ z \subseteq C\}$$

The reason why we choose to work with sets, rather than predicates, is simply that it casts the technical material in a similar style to $W_\bullet$-refinement (see section 3.3) which also constructs a set from the underlying partial relation.

PROPOSITION 2.4
The following introduction and elimination rules for the weakest precondition of U are derivable:

$$\frac{Pre\ U\ t \quad z' \in Post\ U\ t \vdash z' \in C}{t \in wp\ U\ C}\ (wp^+)$$

where z is a fresh variable.

$$\frac{t \in wp\ U\ C}{Pre\ U\ t}\ (wp_{\circ}^{-}) \qquad \frac{t_0 \in wp\ U\ C \quad t_1' \in Post\ U\ t_0}{t_1' \in C}\ (wp_{1}^{-})$$

∎

Compare definition 2.3 with definition 3.1 of [27].

LEMMA 2.5
The following additional rule is derivable:

$$\frac{Pre\ U\ t}{t \in wp\ U\ (Post\ U\ t)}$$

PROOF. Trivial:

$$\frac{\dfrac{Pre\ U\ t \quad \overline{t_0' \in Post\ U\ t}\ (1)}{t \in wp\ U\ (Post\ U\ t)}\ (1)}{}$$

Since the reverse direction always holds, we have established that $t \in wp\ U\ (Post\ U\ t)$ and $Pre\ U\ t$ are equivalent. ∎

3 Operation refinement

Operation refinement concerns the derivation of a more concrete operation from a given abstract one, without changing the specification of the underlying state. It is sometimes called *algorithmic refinement* [33, 17] or *algorithm design* [43]. We have an immediate question: *what does it mean for one partial relation to refine another?* We begin by introducing three distinct notions of operation refinement[2], based on three different answers to the question above, one of which is based on the weakest precondition interpretation (section 2), the other is a proof-theoretic characterisation that captures refinement directly in the language and in terms of the natural properties of preconditions and postconditions, and the third one is based on the standard approach involving a lifted-totalisation of the underlying partial relations (see *e.g.* [41] and [11]). We then go on to compare them and to show that they are all equivalent. We will use the meta-variables U_0 and U_1 to range over operation schemas (partial relations), where U_0 will always be concrete and U_1 abstract. Since we only deal with operation refinement in this section, the abstract and concrete operations will have the same type, that is $\mathbb{P}(T^{in} \curlyvee T^{out'})$. With this settled, we can now omit the type superscripts in most places in this section.

3.1 WP-refinement

WP-refinement is a characterisation of refinement based on the weakest precondition interpretation introduced in the preceding section. The (second-order) definition is standard (*e.g.* [2] and [34]).
WP-refinement is written $U_0 \sqsupseteq_{wp} U_1$ (U_0 WP-refines U_1) and is given by the following $\mathcal{Z}_C$ definition:

[2] These are also investigated in [12, 13] along with other characterisations of operation refinement.

DEFINITION 3.1

$$U_0 \sqsupseteq_{wp} U_1 =_{df} \forall C^{\mathbb{P}\, T^{out'}} \bullet wp\ U_1\ C \subseteq wp\ U_0\ C$$

PROPOSITION 3.2
The following introduction and elimination rules for WP-refinement are derivable:

$$\frac{z \in wp\ U_1\ C \vdash z \in wp\ U_0\ C}{U_0 \sqsupseteq_{wp} U_1}\ (\sqsupseteq^+_{wp})$$

where z and C are fresh variables.

$$\frac{U_0 \sqsupseteq_{wp} U_1 \quad t \in wp\ U_1\ C}{t \in wp\ U_0\ C}\ (\sqsupseteq^-_{wp})$$

∎

3.2 S-refinement

In this section, we introduce a purely proof-theoretic characterisation of refinement, which is closely connected to (sufficient conditions for) refinement as introduced by Spivey (hence "S"-refinement) in [39] and as discussed in, for example, [27, 29], [16, p.169], [43, p.200] and [35, p.222-223].
This notion is based on two basic observations regarding the properties one expects in a refinement: First, that a refinement may involve the reduction of nondeterminism; second that it may also involve the expansion of the domain of definition[3]. Put another way, we have a refinement providing that *postconditions do not weaken* (we do not permit an increase in nondeterminism in a refinement) and that *preconditions do not strengthen* (we do not permit requirements in the domain of definition to disappear in a refinement).
This notion can be captured by forcing the refinement relation to hold *exactly* when these conditions apply. *S-refinement* is written $U_0 \sqsupseteq_s U_1$ and is given by the definition that leads directly to the following rules:

PROPOSITION 3.3
Let z, z_0, z_1 be fresh variables.

$$\frac{Pre\ U_1\ z \vdash Pre\ U_0\ z \quad Pre\ U_1\ z_0, z_0 \star z_1' \in U_0 \vdash z_0 \star z_1' \in U_1}{U_0 \sqsupseteq_s U_1}\ (\sqsupseteq^+_s)$$

$$\frac{U_0 \sqsupseteq_s U_1 \quad Pre\ U_1\ t}{Pre\ U_0\ t}\ (\sqsupseteq^-_{s_0}) \qquad \frac{U_0 \sqsupseteq_s U_1 \quad Pre\ U_1\ t_0 \quad t_0 \star t_1' \in U_0}{t_0 \star t_1' \in U_1}\ (\sqsupseteq^-_{s_1})$$

∎

This theory does not depend on, and makes no reference to, the $\perp$ value we will require in the next section. It can be formalised in the core theory $\mathcal{Z}_C$.

[3] Some refer to this property as reduction of *undefinedness* (*e.g.* [30]).

3.3 $W_\bullet$-refinement

$W_\bullet$-refinement constitutes the standard model-theoretic characterisation of refinement based on a relational completion operator (see appendix B and [12, 13]). This notion is adapted from [41] (chapter 16, *et seq.*), hence "W" for Woodcock[4]. It is written $U_0 \sqsupseteq_{w_\bullet} U_1$ and given by the following $\mathcal{Z}_C^\perp$ definition:

DEFINITION 3.4

$$U_0 \sqsupseteq_{w_\bullet} U_1 =_{df} \overset{\bullet}{U_0} \subseteq \overset{\bullet}{U_1}$$

We obtain the obvious introduction and elimination rules:

PROPOSITION 3.5
Let z be a fresh variable.

$$\frac{z \in \overset{\bullet}{U_0} \vdash z \in \overset{\bullet}{U_1}}{U_0 \sqsupseteq_{w_\bullet} U_1} \ (\sqsupseteq^+_{W_\bullet}) \qquad \frac{U_0 \sqsupseteq_{w_\bullet} U_1 \quad t \in \overset{\bullet}{U_0}}{t \in \overset{\bullet}{U_1}} \ (\sqsupseteq^-_{W_\bullet})$$

■

3.4 Proving equivalence of refinement theories

Methodologically, we shall be showing that all judgements of refinement in one theory are contained among the refinements sanctioned by another. Such results can always be established proof-theoretically because we have expressed even our model-theoretic approach as a theory (a set of introduction and elimination rules). Specifically, we will show that the refinement relation of a theory T_0 satisfies the elimination rule (or rules) for refinement of another theory T_1. Since the elimination rules and introduction rules of a theory enjoy the usual symmetry properties, this is sufficient to show that all T_0-refinements are also T_1-refinements. Equivalence can then be shown by interchanging the roles of T_0 and T_1 in the above.

3.5 WP-refinement and S-refinement are equivalent

In this section we demonstrate that WP-refinement and S-refinement are equivalent. We begin by showing that WP-refinement satisfies the two S-refinement elimination rules.

PROPOSITION 3.6
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{wp} U_1 \quad Pre\ U_1\ t}{Pre\ U_0\ t}$$

[4]The "bullet" subscript of "W" signifies the relational completion operator involved; in this case the *chaotic* lifted-totalisation. We use this notation in order to distinguish among various refinement characterisations we investigated in [12, 13]; these are based on other notions of relational completion such as: the *strict* (chaotic) lifted totalisation [5], the *abortive* lifted-totalisation [4] [11, p.76] and the *non-lifted* totalisation [12].

PROOF.

$$\dfrac{U_0 \sqsupseteq_{wp} U_1 \quad \dfrac{Pre\ U_1\ t}{t \in wp\ U_1\ (Post\ U_1\ t)}\ (L.\ 2.5)}{\dfrac{t \in wp\ U_0\ (Post\ U_1\ t)}{Pre\ U_0\ t}}$$

∎

Now the second elimination rule.

PROPOSITION 3.7
The following rule is derivable:

$$\dfrac{U_0 \sqsupseteq_{wp} U_1 \quad Pre\ U_1\ t_0 \quad t_0 \star t_1' \in U_0}{t_0 \star t_1' \in U_1}$$

PROOF.

$$\dfrac{\dfrac{U_0 \sqsupseteq_{wp} U_1 \quad \dfrac{Pre\ U_1\ t_0}{t_0 \in wp\ U_1\ (Post\ U_1\ t_0)}\ (L.\ 2.5)}{t_0 \in wp\ U_0\ (Post\ U_1\ t_0)} \qquad \dfrac{t_0 \star t_1' \in U_0}{t_1' \in Post\ U_0\ t_0}}{\dfrac{t_1' \in Post\ U_1\ t_0}{t_0 \star t_1' \in U_1}}$$

∎

THEOREM 3.8

$$\dfrac{U_0 \sqsupseteq_{wp} U_1}{U_0 \sqsupseteq_{s} U_1}$$

PROOF. This follows immediately, by (S^+), from propositions 3.6 and 3.7[5]. ∎

We now show that every S-refinement is a WP-refinement.

PROPOSITION 3.9

$$\dfrac{U_0 \sqsupseteq_{s} U_1 \quad t \in wp\ U_1\ C}{t \in wp\ U_0\ C}$$

PROOF.

$$\dfrac{\dfrac{U_0 \sqsupseteq_{s} U_1 \quad \dfrac{t \in wp\ U_1\ C}{Pre\ U_1\ t}}{Pre\ U_0\ t} \qquad \dfrac{t \in wp\ U_1\ C \qquad \dfrac{\dfrac{U_0 \sqsupseteq_{s} U_1 \quad \dfrac{t \in wp\ U_1\ C}{Pre\ U_1\ t} \quad \dfrac{\overline{t_0' \in Post\ U_0\ t}\ (1)}{t \star t_0' \in U_0}}{t \star t_0' \in U_1}}{t_0' \in Post\ U_1\ t}}{t_0' \in C}\ (1)}{t \in wp\ U_0\ C}$$

∎

[5]The proofs of such theorems are always automatic by the structural symmetry between introduction and elimination rules. We shall not give them in future.

THEOREM 3.10

$$\frac{U_0 \sqsupseteq_s U_1}{U_0 \sqsupseteq_{wp} U_1}$$

∎

Theorems 3.8 and 3.10 establish that WP-refinement and S-refinement are equivalent. It is interesting to note that WP-refinement is a *second-order* definition, involving quantification over sets (the postconditions), whereas S-refinement is a first-order theory: evidently there is nothing *essentially* second-order about weakest precondition refinement.

3.6 WP-refinement and W•-refinement are equivalent

We now show that WP-refinement and W•-refinement are equivalent. In previous work [12] we showed that S-refinement and W•-refinement are equivalent; this section illustrates directly the connection between WP-refinement and W•-refinement, highlighting the precise role of $\perp$ in ensuring the equivalence.
First, we prove that WP-refinement satisfies the W•-refinement elimination rule. We will write $T^\star$ for the set $T^{in}_\perp \star T^{out'}_\perp$.

PROPOSITION 3.11
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{wp} U_1 \quad t_0 \star t_1' \in \overset{\bullet}{U_0}}{t_0 \star t_1' \in \overset{\bullet}{U_1}}$$

PROOF.

$$\dfrac{\dfrac{t_0 \star t_1' \in \overset{\bullet}{U_0}}{t_0 \star t_1' \in T^\star} \qquad \dfrac{\dfrac{U_0 \sqsupseteq_{wp} U_1 \quad \dfrac{\overline{Pre\ U_1\ t_0}\ (1)}{t_0 \in wp\ U_1\ (Post\ U_1\ t_0)}\ (L.\ 2.5)}{t_0 \in wp\ U_0\ (Post\ U_1\ t_0)} \qquad \begin{array}{c}\delta\\ \vdots \\ t_1' \in Post\ U_0\ t_0\end{array}}{t_1' \in Post\ U_1\ t_0}}{t_0 \star t_1' \in U_1}\ (1)$$

$$t_0 \star t_1' \in \overset{\bullet}{U_1}$$

Where δ stands for the following branch:

$$\dfrac{t_0 \star t_1' \in \overset{\bullet}{U_0} \qquad \dfrac{U_0 \sqsupseteq_{wp} U_1 \quad \overline{Pre\ U_1\ t_0}\ (1)}{Pre\ U_0\ t_0}\ (P.\ 3.6)}{t_0 \star t_1' \in U_0}$$

$$t_1' \in Post\ U_0\ t_0$$

∎

From this we immediately get the following theorem:

THEOREM 3.12

$$\frac{U_0 \sqsupseteq_{wp} U_1}{U_0 \sqsupseteq_{w_\bullet} U_1}$$

■

We now show that W$_\bullet$-refinement satisfies the WP-refinement elimination rule. For this, we will need the following lemma:

LEMMA 3.13

$$\frac{U_0 \sqsupseteq_{w_\bullet} U_1 \quad \mathit{Pre}\ U_1\ t}{\mathit{Pre}\ U_0\ t}$$

PROOF.

$$\dfrac{\dfrac{U_0 \sqsupseteq_{w_\bullet} U_1 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\mathit{Pre}\ U_1\ t}{t \in T^{in}}}{t \in T^{in}_{\perp}}\ (L.\ B.3(iii))}{t\star \perp' \in \overset{\bullet}{U_0}}}{t\star \perp' \in \overset{\bullet}{U_1}} \qquad \mathit{Pre}\ U_1\ t}{t\star \perp' \in U_1}}{\dfrac{\mathit{false}}{\mathit{Pre}\ U_0\ t}\ (1)}\ (L.\ A.5)$$

■

PROPOSITION 3.14
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{w_\bullet} U_1 \quad t \in \mathit{wp}\ U_1\ C}{t \in \mathit{wp}\ U_0\ C}$$

PROOF.

$$\dfrac{U_0 \sqsupseteq_{w_\bullet} U_1 \qquad \dfrac{t \in \mathit{wp}\ U_1\ C}{\mathit{Pre}\ U_1\ t}}{\mathit{Pre}\ U_0\ t}\ (L.\ 3.13) \qquad \dfrac{t \in \mathit{wp}\ U_1\ C \quad t_0' \in \mathit{Post}\ U_1\ t \quad \overset{\delta}{\vdots}}{t_0' \in C}\ (1)$$

$$t \in \mathit{wp}\ U_0\ C$$

Where δ stands for the following branch:

$$\dfrac{\dfrac{U_0 \sqsupseteq_{w_\bullet} U_1 \qquad \dfrac{\dfrac{\dfrac{}{t_0' \in \mathit{Post}\ U_0\ t}\ (1)}{t\star t_0' \in U_0}}{t\star t_0' \in \overset{\bullet}{U_0}}\ (L.\ B.3(i))}{t\star t_0' \in \overset{\bullet}{U_1}} \qquad \dfrac{t \in \mathit{wp}\ U_1\ C}{\mathit{Pre}\ U_1\ t}}{t\star t_0' \in U_1}$$

$$t_0' \in \mathit{Post}\ U_1\ t$$

■

The use of lemma 3.13 in this proof is reminiscent of proposition 4.11 in [12] and proposition 5 in [13]. These were used for showing that W$_\bullet$-refinement satisfies the S-refinement elimination rule for preconditions as part of the proof demonstrating that W$_\bullet$-refinement is *sound* with respect to S-refinement. This result captures the fact that W$_\bullet$-refinement guarantees that preconditions do not strengthen. Note that the $\perp$ value, explicitly used in the proof of lemma 3.13, has a crucial role in establishing this property.
The following theorem is then straightforward:

THEOREM 3.15

$$\frac{U_0 \sqsupseteq_{w_\bullet} U_1}{U_0 \sqsupseteq_{wp} U_1}$$

■

Theorems 3.12 and 3.15 together establish that the theories of WP-refinement and W$_\bullet$-refinement are equivalent.
In conclusion, despite their superficial dissimilarity, all three theories of refinement are equivalent. We will, in sections 5 and 6, pursue a similar investigation but generalising to data-refinement with forward and backward simulations.

4 Data Simulations

The methods of data-refinement in state-based systems are well established. The conditions under which a transformation is a correct refinement step can be summarised by two simulation based refinement techniques: *forward simulation* and *backward simulation* [10]. In this section we revise these and introduce some essential material underlying our investigation.

4.1 Background

A data simulation [41, 43] is a relation between an abstract data space and a concrete counterpart. Data simulations[6] are sometimes known as *retrieve relations* [10, 11], *abstraction relations* [21, 20] or *linking/abstraction invariants* [37, 3]. These two techniques that enable us to verify data-refinement are shown by the two semi-commuting diagrams in Fig. 1. Forward and backward simulations are also respectively known as downward and upward simulations [10, 11, 21, 20] due to their directions in the commuting diagrams in Fig. 1; de Rover and Engelhardt [8] name them L and L^{-1} simulation (respectively) after the shapes created by the initial paths in the commuting diagrams. We prefer to name them forward and backward for reasons given in [42]. One significant pragmatic distinction between them is that backward simulation refinement permits the postponement of nondeterministic choices, whereas forward simulation prohibits that [11]. For this very reason, the B Abstract Machine Notation [1, 37] only permits forward simulation refinement [38].

[6] The notion of simulation is overloaded in the literature. Various authors use it to denote a certain refinement technique, whereas others use it to denote the *retrieve relation* used in a certain refinement technique. In this paper we use the word "simulation" to specifically denote a retrieve relation.

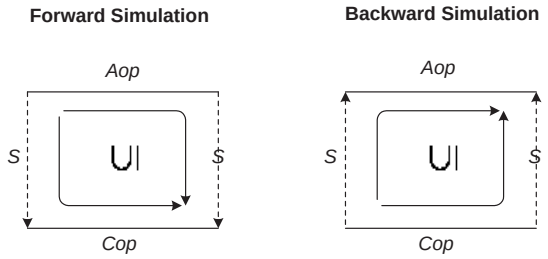

FIG. 1: Forward simulation and backward simulation refinement techniques. *Aop* and *Cop* represent the abstract and concrete operations (respectively), whereas S represents the simulation. Note that a forward simulation is oriented (by composition) from the abstract to the concrete data space and, in a backward simulation, in the opposite direction.

Both forward and backward simulation refinement techniques are known to be sound but neither of them is singly complete. However, they are *jointly complete* [20, 42]. This underlies Derrick's *powersimulation* refinement [9], which constitutes a single complete refinement technique for relational based systems. There is also a single complete refinement technique using predicate transformers [18], where *backward simulation refinement* (named *cosimulation* in that context) is shown to be sound and complete.

4.2 Preliminaries

As in section 3, we will use the meta-variables U_0 and U_1 to respectively range over concrete and abstract operation schemas. We adopt an approach to typing similar to that taken in [8]: our concrete type is $\mathbb{P}(T_0 \Upsilon T_0')$ and the abstract type is $\mathbb{P}(T_1 \Upsilon T_1')$. Therefore, forward simulation (abstract to concrete) is of type $\mathbb{P}(T_1 \Upsilon T_0')$ and backward simulation (concrete to abstract) is of type $\mathbb{P}(T_0 \Upsilon T_1')$.[7] In this way a simulation is modelled as a set of bindings like any other operation schema.
We will need to incorporate the $\perp$ element when a simulation is combined with lifted-totalised operations (see appendix B and [12, 13]). Naturally, the chaotic totalisation [41] is unacceptable here, as this would enforce a link between abstract and concrete states that are not supposed to be linked. The conventional approach [41, 11] is to (non-strictly) lift[8] $\perp$ in the input type of the simulation, thus retaining its partiality. This leads to the following definition:

[7] Having established these types for operations and simulations, we can omit the type superscripts in most places in the sequel.

[8] Lifting signifies mapping $\perp$ of the input set of the relation onto all the states of its output set.

DEFINITION 4.1 (Non-strictly lifted forward simulation)

$$S^{\mathbb{P}(\overset{\circ}{T_1} \curlyvee T'_0)} =_{df} \{z_1 \star z'_0 \in T_{1_\perp} \star T'_{0_\perp} \mid z_1 \neq \perp \Rightarrow z_1 \star z'_0 \in S\}$$

Then the following introduction and elimination rules are derivable:

PROPOSITION 4.2

$$\frac{t_1 \star t'_0 \in T_{1_\perp} \star T'_{0_\perp} \quad t_1 \neq \perp \vdash t_1 \star t'_0 \in S}{t_1 \star t'_0 \in \overset{\circ}{S}} \ (\circ^+) \qquad \frac{t_1 \star t'_0 \in \overset{\circ}{S} \quad t_1 \neq \perp}{t_1 \star t'_0 \in S} \ (\circ^-_\circ)$$

$$\frac{t_1 \star t'_0 \in \overset{\circ}{S}}{t_1 \star t'_0 \in T_{1_\perp} \star T'_{0_\perp}} \ (\circ^-_1)$$

∎

LEMMA 4.3
The following additional rules are derivable for non-strictly lifted forward simulations:

$$\frac{}{S \subseteq \overset{\circ}{S}} \ (i) \qquad \frac{}{\perp \in \overset{\circ}{S}} \ (ii) \qquad \frac{t' \in T'_{0_\perp}}{\perp \star t' \in \overset{\circ}{S}} \ (iii) \qquad \frac{t_1 \star \perp' \in \overset{\circ}{S}}{t_1 = \perp} \ (iv)$$

∎

Likewise, we provide a similar definition for backward simulation:

DEFINITION 4.4 (Non-strictly lifted backward simulation)

$$S^{\mathbb{P}(\overset{\circ}{T_0} \curlyvee T'_1)} =_{df} \{z_0 \star z'_1 \in T_{0_\perp} \star T'_{1_\perp} \mid z_0 \neq \perp \Rightarrow z_0 \star z'_1 \in S\}$$

We obtain similar introduction and elimination rules to the ones in proposition 4.2, modulo appropriate amendments of the types: we will not state them explicitly. Additionally, we have the following standard properties for non-strictly lifted backward simulations:

LEMMA 4.5

$$\frac{}{S \subseteq \overset{\circ}{S}} \ (i) \qquad \frac{}{\perp \in \overset{\circ}{S}} \ (ii) \qquad \frac{t' \in T'_{1_\perp}}{\perp \star t' \in \overset{\circ}{S}} \ (iii) \qquad \frac{t_0 \star \perp' \in \overset{\circ}{S}}{t_0 = \perp} \ (iv)$$

∎

Lemmas 4.3(i – iv) and 4.5(i – iv) demonstrate that definitions 4.1 and 4.4 are consistent with the intentions described in [41] and [11]: the underlying partial relation is contained in the lifting; the $\perp$ element is present in the relation and is mapped onto every after state, and no other initial state is so mapped.

5 Data-refinement with forward simulation

In this section we explore the relationships between three data-refinement theories founded on *forward simulation*. Each of these theories constitutes a generalisation of an operation refinement theory rendered in section 3: *WPF-refinement* based on the weakest precondition semantics; *SF-refinement*, a proof-theoretic notion characterising refinement in terms of the behaviour of preconditions and the fundamental properties expected in a forward simulation refinement; and *WF$_\bullet$-refinement*, a model-theoretic characterisation based on the lifted-totalisation of the operations and (non-strict) lifting of the simulations. We demonstrate that these are all equivalent.

5.1 WPF-refinement

In this section we develop *WPF-refinement*, a forward simulation data-refinement theory based on the weakest precondition semantics presented in section 2. Unlike [28], WPF-refinement will be established solely on the weakest precondition semantics. By taking this approach, we generalise the approach taken in section 3.1 and establish an account which we believe is clear and succinct. It is rather in the spirit of Definition 3 in [33]; however, since we use an underlying relational semantics (rather than predicate transformers) we will require some additional relational nomenclature. In particular we need to express the way in which the simulation properly combines with the postcondition to capture data-refinement.
We define the *image* operator for simulations with respect to a (postcondition) set C. This defines the set of states which belong to the *range* of the simulation after restricting its *domain* to the set C. Therefore, the image operator on forward simulations is a set of type $\mathbb{P}\, T_0$.

DEFINITION 5.1

$$[C^{\mathbb{P}\, T_1}]S^{\mathbb{P}(T_1 \upsilon T_0')} =_{df} \{z_0 \in T_0 \mid \exists\, z_1 \in C \bullet z_1 \star z_0' \in S\}$$

Note the slight subtleties arising from the priming involved.

PROPOSITION 5.2
The following introduction and elimination rules are derivable:

$$\frac{t_1 \in C \quad t_1 \star t_0' \in S}{t_0 \in [C]S}\ ([C]S^+) \qquad \frac{t \in [C]S \quad y \in C, y \star t' \in S \vdash P}{P}\ ([C]S^-)$$

The usual sideconditions apply to the eigenvariable y. ∎

We can now define WPF-refinement written $U_0 \overset{s}{\sqsupseteq}_{wpf} U_1$ (U_0 WPF-refines U_1 with respect to the simulation S)[9].

DEFINITION 5.3

$$U_0 \overset{s}{\sqsupseteq}_{wpf} U_1 =_{df} \forall\, C^{\mathbb{P}\, T_1'} \bullet [wp\ U_1\ C]S \subseteq wp\ U_0\ [C]S'$$

[9] We will omit the superscript S from now on, in this and other notions of refinement that depend upon a simulation.

PROPOSITION 5.4
The following introduction and elimination rules are derivable for WPF-refinement:

$$\frac{z \in [wp\ U_1\ C]S \vdash z \in wp\ U_0\ [C]S'}{U_0 \sqsupseteq_{wpf} U_1}\ (\sqsupseteq^+_{wpf})$$

Where z and C are fresh variables.

$$\frac{U_0 \sqsupseteq_{wpf} U_1 \quad t \in [wp\ U_1\ C]S}{t \in wp\ U_0\ [C]S'}\ (\sqsupseteq^-_{wpf})$$

∎

This theory does not depend on, and makes no reference to, the $\perp$ value; it is formalised in the theory $\mathcal{Z}_C$.

5.2 SF-refinement

In this section, we introduce a proof-theoretic characterisation of forward simulation refinement, which is closely connected to sufficient refinement conditions introduced by, for example, Josephs [28], King [29], Woodcock [41, p.260] (indicated as "F-corr"), Derrick and Boiten [11, p.90], Jacky [26, p.251] and Potter *et al.* [35, p.236]. These conditions correspond to the premises of our introduction rule for *SF-refinement*.
As a generalisation of S-refinement (section 3.2), this notion is based on the two properties expected in a refinement: that *postconditions do not weaken and preconditions do not strengthen in the presence of a forward simulation*. Thus, SF-refinement can be expressed by admitting refinement to hold *precisely* when these conditions apply. It is written $U_0 \overset{s}{\sqsupseteq}_{sf} U_1$ and is given by the $\mathcal{Z}_C$ definition that leads directly to the following rules:

PROPOSITION 5.5
Let x_0, x_1, z_0, z_1, z_2 be fresh variables.

$$\frac{\begin{array}{lcl} z_1 \star z_0' \in S, Pre\ U_1\ z_1 & \vdash & Pre\ U_0\ z_0 \\ Pre\ U_1\ x_1, x_0 \star z_2' \in U_0, x_1 \star x_0' \in S & \vdash & x_1 \star t' \in U_1 \\ Pre\ U_1\ x_1, x_0 \star z_2' \in U_0, x_1 \star x_0' \in S & \vdash & t \star z_2' \in S \end{array}}{U_0 \sqsupseteq_{sf} U_1}\ (\sqsupseteq^+_{sf})$$

$$\frac{U_0 \sqsupseteq_{sf} U_1 \quad Pre\ U_1\ t_1 \quad t_1 \star t_0' \in S}{Pre\ U_0\ t_0}\ (\sqsupseteq^-_{sf_\circ})$$

$$\frac{U_0 \sqsupseteq_{sf} U_1 \quad Pre\ U_1\ t_1 \quad t_0 \star t_2' \in U_0 \quad t_1 \star t_0' \in S \quad t_1 \star y' \in U_1, y \star t_2' \in S \vdash P}{P}\ (\sqsupseteq^-_{sf_1})$$

The usual sideconditions apply to the eigenvariable y. ∎

5.3 WF$_\bullet$-refinement

WF$_\bullet$-refinement constitutes a generalisation of W$_\bullet$-refinement (section 3.3) to forward simulation data-refinement. This characterisation of refinement is based on the (chaotic) relational completion operator (set out in appendix B) applied to the underlying operations and on (non-strict) lifting of the simulation (as developed in section 4.2). It is also discussed in [41, p.246] and [10].
WF$_\bullet$-refinement captures, schematically, the forward simulation commuting diagram in Fig. 1 and is based on *schema* or, more generally, *relational composition* (see appendix A, proposition A.1). It is written $U_0 \overset{s}{\sqsupseteq}_{wf_\bullet} U_1$ and is defined in $\mathcal{Z}_C^\perp$ as follows:

DEFINITION 5.6

$$U_0 \overset{s}{\sqsupseteq}_{wf_\bullet} U_1 =_{df} \overset{\circ}{S} \fatsemi \overset{\bullet}{U_0} \subseteq \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}$$

Obvious introduction and elimination rules for WF$_\bullet$-refinement follow from this definition.

5.4 WPF-refinement and SF-refinement are equivalent

We now show that WPF-refinement and SF-refinement are equivalent, beginning by proving that WPF-refinement satisfies the two SF-refinement elimination rules. First the rule for preconditions.

PROPOSITION 5.7
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{wpf} U_1 \quad Pre\ U_1\ t_1 \quad t_1 \star t_0' \in S}{Pre\ U_0\ t_0}$$

PROOF.

$$\dfrac{\dfrac{U_0 \sqsupseteq_{wpf} U_1 \qquad \dfrac{\dfrac{Pre\ U_1\ t_1}{t_1 \in wp\ U_1\ (Post\ U_1\ t_1)}\ (L.\ 2.5) \qquad t_1 \star t_0' \in S}{t_0 \in [wp\ U_1\ (Post\ U_1\ t_1)]S}}{t_0 \in wp\ U_0\ [Post\ U_1\ t_1]S'}}{Pre\ U_0\ t_0}$$

∎

Turning now to the second elimination rule in SF-refinement.

PROPOSITION 5.8
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{wpf} U_1 \quad Pre\ U_1\ t_1 \quad t_0 \star t_2' \in U_0 \quad t_1 \star t_0' \in S \quad t_1 \star y' \in U_1, y \star t_2' \in S \vdash P}{P}$$

PROOF.

$$\dfrac{U_0 \sqsupseteq_{wpf} U_1 \quad \dfrac{\dfrac{Pre\ U_1\ t_1}{t_1 \in wp\ U_1\ (Post\ U_1\ t_1)}\ (L.\ 2.5) \quad t_1 \star t_0' \in S}{t_0 \in [wp\ U_1\ (Post\ U_1\ t_1)]S} \quad \dfrac{t_0 \star t_2' \in U_0}{t_2' \in Post\ U_0\ t_0} \quad \begin{array}{c}\delta\\ \vdots \\ P\end{array}}{\dfrac{t_0 \in wp\ U_0\,[Post\ U_1\ t_1]S'}{\dfrac{t_2' \in [Post\ U_1\ t_1]S'}{P}}}\ (1)$$

Where δ stands for the following branch:

$$\dfrac{\dfrac{\dfrac{}{y' \in Post\ U_1\ t_1}\ (1)}{t_1 \star y' \in U_1} \quad \dfrac{\dfrac{}{y' \star t_2 \in S'}\ (1)}{y \star t_2' \in S}}{t_1 \star y' \in U_1 \wedge y \star t_2' \in S}$$
$$\vdots$$
$$P$$

THEOREM 5.9

$$\dfrac{U_0 \sqsupseteq_{wpf} U_1}{U_0 \sqsupseteq_{sf} U_1}$$

We now show that SF-refinement satisfies the WPF-elimination rule.

PROPOSITION 5.10
The following rule is derivable:

$$\dfrac{U_0 \sqsupseteq_{sf} U_1 \quad t \in [wp\ U_1\ C]S}{t \in wp\ U_0\,[C]S'}$$

PROOF.

$$\dfrac{t \in [wp\ U_1\ C]S \quad \dfrac{\dfrac{U_0 \sqsupseteq_{sf} U_1 \quad \dfrac{\dfrac{}{y \in wp\ U_1\ C}\ (1)}{Pre\ U_1\ y} \quad \dfrac{}{y \star t' \in S}\ (1)}{Pre\ U_0\ t} \quad \begin{array}{c}\delta_0\\ \vdots \\ t_0' \in [C]S'\end{array}}{t \in wp\ U_0\,[C]S'}\ (2)}{t \in wp\ U_0\,[C]S'}\ (1)$$

Where δ_0 is:

$$\dfrac{U_0 \sqsupseteq_{sf} U_1 \quad \dfrac{\dfrac{}{y \in wp\ U_1\ C}\ (1)}{Pre\ U_1\ y} \quad \dfrac{\dfrac{}{t_0' \in Post\ U_0\ t}\ (2)}{t \star t_0' \in U_0} \quad \dfrac{}{y \star t' \in S}\ (1) \quad \begin{array}{c}\delta_1\\ \vdots \\ t_0' \in [C]S'\end{array}}{t_0' \in [C]S'}\ (3)$$

and δ_1 is:

$$\dfrac{\dfrac{}{y \in wp\ U_1\ C}\ (1) \quad \dfrac{\dfrac{}{y \star w' \in U_1}\ (3)}{w' \in Post\ U_1\ y} \qquad \dfrac{}{w \star t_0' \in S}\ (3)}{\dfrac{w' \in C \qquad\qquad w' \star t_0 \in S'}{t_0' \in [C]S'}}$$

∎

THEOREM 5.11

$$\dfrac{U_0 \sqsupseteq_{sf} U_1}{U_0 \sqsupseteq_{wpf} U_1}$$

∎

Theorems 5.9 and 5.11 together establish that the theories of SF-refinement and WPF-refinement are equivalent.

5.5 WPF-refinement and WF$_\bullet$-refinement are equivalent

In this section, we demonstrate that WPF-refinement and WF$_\bullet$-refinement are equivalent. We begin by showing that every WPF-refinement is a WF$_\bullet$-refinement. For this we prove that WPF-refinement satisfies the elimination rule for WF$_\bullet$-refinement.

PROPOSITION 5.12
The following rule is derivable:

$$\dfrac{U_0 \sqsupseteq_{wpf} U_1 \quad t_1 \star t_0' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_0}}{t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}}$$

Notice that the following derivation depends on the use of the *law of excluded middle* (see, for example, [40]).

PROOF.

$$\dfrac{t_1 \star t_0' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_0} \qquad \dfrac{\dfrac{}{Pre\ U_1\ t_1 \vee \neg\ Pre\ U_1\ t_1}\ (LEM) \qquad \overset{\delta_0}{\vdots}\ t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S} \qquad \overset{\delta_1}{\vdots}\ t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}}{t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}}\ (2)}{t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}}\ (1)$$

Where δ_0 stands for the following branch:

$$\dfrac{\overset{\beta_0}{\vdots}\ t_0' \in [Post\ U_1\ t_1]S' \qquad \dfrac{\dfrac{\dfrac{}{w' \in Post\ U_1\ t_1}\ (3)}{t_1 \star w' \in U_1}\ (L.\ B.3(i))}{t_1 \star w' \in \overset{\bullet}{U_1}} \qquad \dfrac{\dfrac{}{w' \star t_0 \in S'}\ (3)}{w \star t_0' \in \overset{\circ}{S}}\ (L.\ 4.3(i))}{t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}}\ (3)$$

Where β_0 is:

$$\dfrac{\dfrac{\beta_1}{y \in wp\ U_0\,[Post\ U_1\ t_1]S'} \qquad \dfrac{\dfrac{\dfrac{}{y \star t_0' \in \overset{\bullet}{U_0}}\,(1) \qquad \dfrac{\beta_1}{y \in wp\ U_0\,[Post\ U_1\ t_1]S'}}{Pre\ U_0\ y}}{y \star t_0' \in U_0}}{t_0' \in [Post\ U_1\ t_1]S'}$$

and β_1 is:

$$\dfrac{U_0 \sqsupseteq_{wpf} U_1 \qquad \dfrac{\dfrac{\dfrac{}{Pre\ U_1\ t_1}\,(2)}{t_1 \in wp\ U_1\,(Post\ U_1\ t_1)}\,(L.\ 2.5) \qquad \dfrac{\dfrac{}{t_1 \star y' \in \overset{\circ}{S}}\,(1) \qquad \dfrac{\dfrac{}{Pre\ U_1\ t_1}\,(2)}{t_1 \neq \bot}\,(L.\ A.5)}{t_1 \star y' \in S}}{y \in [wp\ U_1\,(Post\ U_1\ t_1)]S}}{y \in wp\ U_0\,[Post\ U_1\ t_1]S'}$$

and δ_1 stands for the following branch:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\ Pre\ U_1\ t_1}\,(2) \qquad \dfrac{\dfrac{\dfrac{}{t_1 \star y' \in \overset{\circ}{S}}\,(1)}{t_1 \star y' \in T_{1_\bot} \star T'_{0_\bot}}}{t_1 \in T_{1_\bot}}\,(L.\ B.3(iii))}{t_1 \star \bot' \in \overset{\bullet}{U_1}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{y \star t_0' \in \overset{\bullet}{U_0}}\,(1)}{y \star t_0' \in T_{0_\bot} \star T'_{0_\bot}}}{t_0' \in T'_{0_\bot}}\,(L.\ 4.3(iii))}{\bot \star t_0' \in \overset{\circ}{S}}}{t_1 \star t_0' \in \overset{\bullet}{U_1} \fatsemi \overset{\circ}{S}}$$

∎

The following theorem is then immediate:

THEOREM 5.13

$$\dfrac{U_0 \sqsupseteq_{wpf} U_1}{U_0 \sqsupseteq_{wf_\bullet} U_1}$$

∎

We now show that WF$_\bullet$-refinement satisfies the elimination rule for WPF-refinement. For this, we will need the following lemma:

LEMMA 5.14

$$\dfrac{U_0 \sqsupseteq_{wf_\bullet} U_1 \quad Pre\ U_1\ t_1 \quad t_1 \star t_0' \in S}{Pre\ U_0\ t_0}$$

PROOF.

$$\dfrac{\dfrac{\delta}{t_1 \star \perp' \in \overset{\bullet}{U_1} \mathbin{⨾} \overset{\circ}{S}} \quad \dfrac{\dfrac{\dfrac{}{t_1 \star y' \in \overset{\bullet}{U_1}}\,(2) \quad \dfrac{\dfrac{}{y \star \perp' \in \overset{\circ}{S}}\,(2)}{y = \perp}\,(L.\ 4.3(iv))}{t_1 \star \perp' \in \overset{\bullet}{U_1}} \quad Pre\ U_1\ t_1}{t_1 \star \perp' \in U_1} \quad \dfrac{}{false}\,(L.\ A.5)}{false}\,(2)}{Pre\ U_0\ t_0}\,(1)$$

where δ is:

$$\dfrac{U_0 \sqsupseteq_{wf_\bullet} U_1 \quad \dfrac{\dfrac{t_1 \star t_0' \in S}{t_1 \star t_0' \in \overset{\circ}{S}}\,(L.\ 4.3(i)) \quad \dfrac{\dfrac{}{\neg Pre\ U_0\ t_0}\,(1) \quad \dfrac{\dfrac{t_1 \star t_0' \in S}{t_0 \in T_0}}{t_0 \in T_{0_\perp}}}{t_0 \star \perp' \in \overset{\bullet}{U_0}}\,(L.\ B.3(iii))}{t_1 \star \perp' \in \overset{\circ}{S} \mathbin{⨾} \overset{\bullet}{U_0}}}{t_1 \star \perp' \in \overset{\bullet}{U_1} \mathbin{⨾} \overset{\circ}{S}}$$

∎

PROPOSITION 5.15
The following rule is derivable:

$$\dfrac{U_0 \sqsupseteq_{wf_\bullet} U_1 \quad t \in [wp\ U_1\ C]S}{t \in wp\ U_0\ [C]S'}$$

PROOF.

$$\dfrac{t \in [wp\ U_1\ C]S \quad \dfrac{\dfrac{U_0 \sqsupseteq_{wf_\bullet} U_1 \quad \dfrac{\dfrac{}{y \in wp\ U_1\ C}\,(1)}{Pre\ U_1\ y} \quad \dfrac{}{y \star t' \in S}\,(1)}{Pre\ U_0\ t}\,(L.\ 5.14) \quad \dfrac{\delta_0}{t_0' \in [C]S'}}{t \in wp\ U_0\ [C]S'}\,(2)}{t \in wp\ U_0\ [C]S'}\,(1)$$

Where δ_0 is:

$$\dfrac{U_0 \sqsupseteq_{wf_\bullet} U_1 \quad \dfrac{\dfrac{\dfrac{}{y \star t' \in S}\,(1)}{y \star t' \in \overset{\circ}{S}}\,(L.\ 4.3(i)) \quad \dfrac{\dfrac{}{t_0' \in Post\ U_0\ t}\,(2)}{t \star t_0' \in U_0}}{t \star t_0' \in \overset{\bullet}{U_0}}\,(L.\ B.3(i))}{y \star t_0' \in \overset{\circ}{S} \mathbin{⨾} \overset{\bullet}{U_0}} \quad \dfrac{\delta_1}{t_0' \in [C]S'}}{t_0' \in [C]S'}\,(3)$$

and δ_1 is:

$$\dfrac{\dfrac{\dfrac{}{y \in wp\ U_1\ C}\ (1) \quad \dfrac{\overset{\alpha}{\vdots}\ y \star w' \in U_1}{w' \in Post\ U_1\ y}}{w' \in C} \qquad \dfrac{\dfrac{}{w \star t_0' \in \overset{\circ}{S}}\ (3) \quad \dfrac{\overset{\alpha}{\vdots}\ y \star w' \in U_1}{w \neq \bot}\ (L.\ A.5)}{w \star t_0' \in S}}{w' \star t_0 \in S'}}{t_0' \in [C]S'}$$

Where α stands for the following branch:

$$\dfrac{\dfrac{}{y \star w' \in \overset{\bullet}{U}_1}\ (3) \quad \dfrac{\dfrac{}{y \in wp\ U_1\ C}\ (1)}{Pre\ U_1\ y}}{y \star w' \in U_1}$$

∎

The fact that the proof above depends on the use of lemma 5.14 is consistent with the investigation carried out in section 3.6. This lemma once again captures the basic property expected in a refinement, but in the context of WF$_\bullet$-refinement: preconditions do not strengthen in the presence of forward simulation. It is part of the proof demonstrating that WF$_\bullet$-refinement is *sound* with respect to SF-refinement[10], where, again, the $\bot$ value is vital for establishing this result.
The following theorem is then derivable:

THEOREM 5.16

$$\dfrac{U_0 \sqsupseteq_{wf_\bullet} U_1}{U_0 \sqsupseteq_{wpf} U_1}$$

∎

Together, theorems 5.13 and 5.16 show that WPF-refinement and WF$_\bullet$-refinement are equivalent. Naturally, we have as a final corollary for this section that SF- refinement and WF$_\bullet$-refinement are also equivalent.
Once again, we have demonstrated that what appear to be quite different models of specification and refinement are, in fact, intimately related.

6 Data-refinement with backward simulation

This section provides an analogous investigation to that provided in the previous section. Here we explore three *backward simulation* data-refinement theories, again, generalising the corresponding operation refinement theories introduced in section 3: *WPB-refinement*, a weakest precondition backwards refinement; *SB-refinement*, the proof-theoretic characterisation of data-refinement with backward simulation; and *WB$_\bullet$-refinement*, a model-theoretic notion characterising refinement underlying lifted-totalised operations and lifted backward simulation. Using the same approach as in sections 3 and 5, we will show that these three notions of refinement are all equivalent.

[10] This result constitutes part of a general investigation of forward simulation data-refinement: this will appear as part III of the series of papers of which the current paper is the second.

6.1 WPB-refinement

We will now develop *WPB-refinement*, the backward simulation counterpart theory to WPF-refinement (section 5.1) and based on the weakest precondition semantics introduced in section 2.
The key issue in the previous section was discovering the correct mechanism for relating the simulation and the postcondition in the context of a weakest precondition framework. In this case we require the *left residual* of a backward simulation S *under* the set C. This is rather in the spirit of the calculus for binary relations [36], in which *residuation* of binary relations correlates with *numeric division*; thus, Pratt [36] uses the notation a/b to denote the left residual of b under a, where both a and b are binary relations. In contrast, having C as a set (of abstract states), rather than a relation, induces our notion of left residuation to produce a set (of concrete states) as opposed to a relation in [36]. We define the left residual of S under C to be the set (of type $\mathbb{P}\, T_0$) of all concrete states, drawn from the domain of S, which *only* represent abstract states that are members of C.

DEFINITION 6.1

$$S^{\mathbb{P}(T_0 \curlyvee T_1')}[C^{\mathbb{P}\, T_1}] =_{df} \{z_0 \in T_0 \mid \forall\, z_1 \bullet z_0 \star z_1' \in S \Rightarrow z_1 \in C\}$$

Note carefully the subtleties arising from the priming in the definition above.

PROPOSITION 6.2
Let z be a fresh variable, then the following introduction and elimination rules for the left residual of S under C are derivable:

$$\frac{t_0 \star z' \in S \vdash z \in C}{t_0 \in S[C]}\ (S[C]^+) \qquad \frac{t_0 \in S[C] \quad t_0 \star t_1' \in S}{t_1 \in C}\ (S[C]^-)$$

∎

WPB-refinement is written $U_0 \overset{s}{\sqsupseteq}_{wpb} U_1$ and is defined in $\mathcal{Z}_C$ as follows:

DEFINITION 6.3

$$U_0 \overset{s}{\sqsupseteq}_{wpb} U_1 =_{df} \forall\, C^{\mathbb{P}\, T_1'} \bullet S[wp\ U_1\ C] \subseteq wp\ U_0\ S'[C]$$

PROPOSITION 6.4
The following introduction and elimination rules are derivable for WPB-refinement:

$$\frac{z \in S[wp\ U_1\ C] \vdash z \in wp\ U_0\ S'[C]}{U_0 \sqsupseteq_{wpb} U_1}\ (\sqsupseteq^+_{wpb})$$

where z and C are fresh variables.

$$\frac{U_0 \sqsupseteq_{wpb} U_1 \quad t \in S[wp\ U_1\ C]}{t \in wp\ U_0\ S'[C]}\ (\sqsupseteq^-_{wpb})$$

∎

6.2 SB-refinement

SB-refinement constitutes a proof-theoretic characterisation of data-refinement, generalising S-refinement (section 3.2) with backward simulation. It embodies the two standard attributes expected in a refinement in the presence of (backward) simulation. This notion of refinement is reminiscent of sufficient refinement conditions introduced by Woodcock [41, p.270] (indicated as "B-corr") and by Derrick and Boiten [11, p.93]. These conditions correspond to the premises of our introduction rule for SB-refinement.

SB-refinement is written $U_0 \overset{s}{\sqsupseteq}_{sb} U_1$ and is given by the definition that leads directly to the following introduction and elimination rules:

PROPOSITION 6.5
Let x, x_0, x_1, z, z_0 be fresh variables.

$$\frac{\begin{array}{ll} x \star z' \in S \Rightarrow Pre\ U_1\ z & \vdash\ Pre\ U_0\ x \\ z_0 \star z' \in S \Rightarrow Pre\ U_1\ z, x_0 \star x_1' \in S, z_0 \star x_0' \in U_0 & \vdash\ z_0 \star t' \in S \\ z_0 \star z' \in S \Rightarrow Pre\ U_1\ z, x_0 \star x_1' \in S, z_0 \star x_0' \in U_0 & \vdash\ t \star x_1' \in U_1 \end{array}}{U_0 \sqsupseteq_{sb} U_1}\ (\sqsupseteq^{+}_{sb})$$

$$\frac{U_0 \sqsupseteq_{sb} U_1 \quad t \star z' \in S \vdash Pre\ U_1\ z}{Pre\ U_0\ t}\ (\sqsupseteq^{-}_{sb_0})$$

$$\frac{\begin{array}{ll} & t_0 \star z' \in S \vdash Pre\ U_1\ z \\ U_0 \sqsupseteq_{sb} U_1 & t_1 \star t_2' \in S \quad t_0 \star t_1' \in U_0 \quad t_0 \star y' \in S, y \star t_2' \in U_1 \vdash P \end{array}}{P}\ (\sqsupseteq^{-}_{sb_1})$$

The usual sideconditions apply to the eigenvariable y. ∎

This theory does not depend on, and makes no reference to, the $\perp$ value; it is formalised in the core theory $\mathcal{Z}_{\mathcal{C}}$.

Notice the *universal force* of the variable z in the rightmost premise of $(\sqsupseteq^{-}_{sb_0})$. In fact, we can see the similarity between this premise and the notion of *left residuation* delineated in the preceding section. Rephrasing that elimination rule according to this notion shows that t is in the precondition of U_0 *only if* $U_0 \sqsupseteq_{sb} U_1$ and t belongs to the *left residual* of S under the *domain* of U_1. In other words, a concrete state is in the precondition of the concrete operation *only if every* abstract state that it represents (by the simulation) is manifested by the abstract operation (providing SB-refinement holds).

6.3 $WB_\bullet$-refinement

We now introduce *$WB_\bullet$-refinement*, a model-theoretic characterisation of backward simulation refinement based on a lifted-totalised operation (appendix B) and a lifted simulation (section 4.2). This notion of refinement is also discussed in [41, p.247] and [10]. $WB_\bullet$-refinement captures the backward simulation commuting diagram in Fig. 1. using relational composition. It is written $U_0 \overset{s}{\sqsupseteq}_{wb_\bullet} U_1$ and is given by the following definition in $\mathcal{Z}^{\perp}_{\mathcal{C}}$:

DEFINITION 6.6

$$U_0 \overset{s}{\sqsupseteq}_{wb\bullet} U_1 =_{df} \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S} \subseteq \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}$$

We obtain obvious introduction and elimination rules for WB$_\bullet$-refinement, which in this case we will not state explicitly.

6.4 WPB-refinement and SB-refinement are equivalent

We now show that WPB-refinement and SB-refinement are equivalent using our usual technique. We begin by demonstrating that every WPB-refinement is also an SB-refinement by proving that WPB-refinement satisfies both SB-elimination rules. We begin with the rule for preconditions.

PROPOSITION 6.7
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{wpb} U_1 \quad t \star z' \in S \vdash Pre\ U_1\ z}{Pre\ U_0\ t}$$

PROOF.

$$\dfrac{\dfrac{U_0 \sqsupseteq_{wpb} U_1 \quad \dfrac{\dfrac{\dfrac{}{t \star t_0' \in S}\,(1)}{\vdots}}{\dfrac{Pre\ U_1\ t_0}{t_0 \in wp\ U_1\ (Post\ U_1\ z)}\,(L.\ 2.5)}}{t \in S[wp\ U_1\ (Post\ U_1\ z)]}\,(1)}{t \in wp\ U_0\ S'[Post\ U_1\ z]}$$
$$\overline{Pre\ U_0\ t}$$

∎

Now we consider the second elimination rule in SB-refinement.

PROPOSITION 6.8
The following rule is derivable:

$$\frac{\begin{array}{ll} & t_0 \star z' \in S \vdash Pre\ U_1\ z \\ U_0 \sqsupseteq_{wpb} U_1 & t_1 \star t_2' \in S \quad t_0 \star t_1' \in U_0 \quad t_0 \star y' \in S, y \star t_2' \in U_1 \vdash P \end{array}}{P}$$

The following derivation depends on the use of the *law of excluded middle*.

PROOF.

$$\dfrac{\begin{array}{ll} \delta_0 & \dfrac{\dfrac{}{t_0 \star y' \in S}\,(1) \quad \dfrac{}{y \star t_2' \in U_1}\,(1)}{t_0 \star y' \in S \wedge y \star t_2' \in U_1} \\ \vdots & \vdots \\ t_0 \star t_2' \in S \fatsemi U_1 & P \end{array}}{P}\,(1)$$

Where δ_0 stands for the following branch:

$$\dfrac{\dfrac{}{t_0 \star t' \in S \vee t_0 \star t' \notin S}\ (LEM) \quad \dfrac{\dfrac{}{t_0 \star t' \in S}\ (2) \quad \overset{\delta_1}{\vdots}\ t \star t_2' \in U_1}{t_0 \star t_2' \in S \fatsemi U_1} \quad \overset{\delta_2}{\vdots}\ t_0 \star t_2' \in S \fatsemi U_1}{t_0 \star t_2' \in S \fatsemi U_1}\ (2)$$

Where δ_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{t_0 \star t' \in S}\ (3)}{\vdots}}{Pre\ U_1\ t}}{t \in wp\ U_1\ (Post\ U_1\ t)}\ (L.\ 2.5)}{t_0 \in S[wp\ U_1\ (Post\ U_1\ t)]}\ (3) \quad U_0 \sqsupseteq_{wpb} U_1}{t_0 \in wp\ U_0\ S'[Post\ U_1\ t]} \quad \dfrac{t_0 \star t_1' \in U_0}{t_1' \in Post\ U_0\ t_0}}{t_1' \in S'[Post\ U_1\ t]} \quad \dfrac{t_1 \star t_2' \in S}{t_1' \star t_2 \in S'}}{t_2' \in Post\ U_1\ t}}{t \star t_2' \in U_1}$$

and δ_2 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{t_0 \star t' \notin S}\ (2)}{t_0 \in S[wp\ U_1\ (Post\ S \fatsemi U_1\ t_0)]}\ (*) \quad U_0 \sqsupseteq_{wpb} U_1}{t_0 \in wp\ U_0\ S'[Post\ S \fatsemi U_1\ t_0]} \quad \dfrac{t_0 \star t_1' \in U_0}{t_1' \in Post\ U_0\ t_0}}{t_1' \in S'[Post\ S \fatsemi U_1\ t_0]} \quad \dfrac{t_1 \star t_2' \in S}{t_1' \star t_2 \in S'}}{t_2' \in Post\ S \fatsemi U_1\ t_0}}{t_0 \star t_2' \in S \fatsemi U_1}$$

∎

Note that the proof step labelled $(*)$ above is justified by the fact that definition 6.1 can be expressed using disjunction (in the obvious way) in place of implication, in which case we obtain two introduction rules that correspond to both cases of *disjunction introduction*; thus, $(*)$ denotes the application of $(S[C]^{+}_{\mathrm{o}})$.
The following theorem is then immediate:

THEOREM 6.9

$$\dfrac{U_0 \sqsupseteq_{wpb} U_1}{U_0 \sqsupseteq_{sb} U_1}$$

∎

We now show that SB-refinement satisfies the WPB-elimination rule.

PROPOSITION 6.10
The following rule is derivable:

$$\dfrac{U_0 \sqsupseteq_{sb} U_1 \quad t \in S[wp\ U_1\ C]}{t \in wp\ U_0\ S'[C]}$$

PROOF.

$$\dfrac{\dfrac{U_0 \sqsupseteq_{sb} U_1 \quad \dfrac{\dfrac{t \in S[wp\ U_1\ C] \quad \overline{t \star t_2' \in S}\ (2)}{t_2 \in wp\ U_1\ C}}{Pre\ U_1\ t_2}}{Pre\ U_0\ t}\ (2) \qquad \dfrac{\delta}{\dfrac{t_1' \in C}{t_0' \in S'[C]}\ (3)}}{t \in wp\ U_0\ S'[C]}\ (1)$$

Where δ stands for the following branch:

$$\dfrac{U_0 \sqsupseteq_{sb} U_1 \quad \overset{\beta_0}{\vdots}\ t \star t_3' \in S \Rightarrow Pre\ U_1\ t_3 \quad \dfrac{\overline{t_0' \in Post\ U_0\ t}\ (1)}{t \star t_0' \in U_0} \quad \dfrac{\overline{t_0' \star t_1 \in S'}\ (3)}{t_0 \star t_1' \in S} \quad \overset{\beta_1}{\vdots}\ t_1' \in C}{t_1' \in C}\ (4)$$

Where β_0 is:

$$\dfrac{\dfrac{t \in S[wp\ U_1\ C] \quad \overline{t \star t_3' \in S}\ (5)}{t_3 \in wp\ U_1\ C}}{\dfrac{Pre\ U_1\ t_3}{t \star t_3' \in S \Rightarrow Pre\ U_1\ t_3}\ (5)}$$

and β_1 is:

$$\dfrac{t \in S[wp\ U_1\ C] \quad \overline{t \star y' \in S}\ (4) \qquad \dfrac{\overline{y \star t_1' \in U_1}\ (4)}{t_1' \in Post\ U_1\ y}}{t_1' \in C}$$

∎

THEOREM 6.11

$$\dfrac{U_0 \sqsupseteq_{sb} U_1}{U_0 \sqsupseteq_{wpb} U_1}$$

∎

Theorems 6.9 and 6.11 together establish that the theories of SB-refinement and WPB-refinement are equivalent.

6.5 WPB-refinement and WB$_\bullet$-refinement are equivalent

We now demonstrate that WPB-refinement and WB$_\bullet$-refinement are equivalent. To begin with, we prove that WPB-refinement satisfies the elimination rule for WB$_\bullet$-refinement.

PROPOSITION 6.12
The following rule is derivable:

$$\dfrac{U_0 \sqsupseteq_{wpb} U_1 \quad t_0 \star t_1' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S}}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}$$

The following derivation also depends on the use of the *law of excluded middle*.

PROOF. Let ϕ be: $\forall z \bullet t_0 \star z' \in S \Rightarrow Pre\ U_1\ z \vee \exists z \bullet t_0 \star z' \in S \wedge \neg\ Pre\ U_1\ z$

$$\dfrac{t_0 \star t_1' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S} \qquad \dfrac{\dfrac{}{\phi}\ (LEM) \qquad \begin{matrix}\delta_0 \\ \vdots \\ t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}\end{matrix} \qquad \begin{matrix}\delta_1 \\ \vdots \\ t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}\end{matrix}}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}\ (2)}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}\ (1)$$

Where δ_0 stands for the following branch (provable due to proposition 6.8):

$$\dfrac{U_0 \sqsupseteq_{wpb} U_1 \qquad \dfrac{}{\forall z \bullet t_0 \star z' \in S \Rightarrow Pre\ U_1\ z}\ (2) \qquad \begin{matrix}\beta_0 \\ \vdots \\ t_0 \star y' \in U_0\end{matrix} \qquad \begin{matrix}\beta_1 \\ \vdots \\ y \star t_1' \in S\end{matrix} \qquad \begin{matrix}\beta_2 \\ \vdots \\ t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}\end{matrix}}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}\ (3)$$

Where β_0 is:

$$\dfrac{\dfrac{}{t_0 \star y' \in \overset{\bullet}{U_0}}\ (1) \qquad \dfrac{U_0 \sqsupseteq_{wpb} U_1 \qquad \dfrac{}{\forall z \bullet t_0 \star z' \in S \Rightarrow Pre\ U_1\ z}\ (2)}{Pre\ U_0\ t_0}\ (P.\ 6.7)}{t_0 \star y' \in U_0}$$

and β_1, β_2 are respectively:

$$\dfrac{\dfrac{}{y \star t_1' \in \overset{\circ}{S}}\ (1) \qquad \dfrac{\begin{matrix}\beta_0 \\ \vdots \\ t_0 \star y' \in U_0\end{matrix}}{y \neq \bot}\ (L.\ A.5)}{y \star t_1' \in S} \qquad \dfrac{\dfrac{\dfrac{}{t_0 \star w_0' \in S}\ (3)}{t_0 \star w_0' \in \overset{\circ}{S}}\ (L.\ 4.5(i)) \qquad \dfrac{\dfrac{}{w_0 \star t_1' \in U_1}\ (3)}{w_0 \star t_1' \in \overset{\bullet}{U_1}}\ (L.\ B.3(i))}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}$$

δ_1 stands for the following branch:

$$\dfrac{\dfrac{}{\exists z \bullet t_0 \star z' \in S \wedge \neg\ Pre\ U_1\ z}\ (2) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{t_0 \star w_1' \in S \wedge \neg\ Pre\ U_1\ w_1}\ (4)}{t_0 \star w_1' \in S}\ (L.\ 4.5(i))}{t_0 \star w_1' \in \overset{\circ}{S}} \qquad \begin{matrix}\alpha \\ \vdots \\ w_1 \star t_1' \in \overset{\bullet}{U_1}\end{matrix}}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}\ (4)}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}$$

Where α is:

$$\dfrac{\dfrac{\dfrac{}{t_0 \star w_1' \in S \wedge \neg\ Pre\ U_1\ w_1}\ (4)}{\neg\ Pre\ U_1\ w_1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{t_0 \star w_1' \in S \wedge \neg\ Pre\ U_1\ w_1}\ (4)}{t_0 \star w_1' \in S}}{w_1 \in T_1}}{w_1 \in T_{1_\bot}} \qquad \dfrac{\dfrac{\dfrac{}{y \star t_1' \in \overset{\circ}{S}}\ (1)}{y \star t_1' \in T_{0_\bot} \star T'_{1_\bot}}}{t_1' \in T'_{1_\bot}}}{w_1 \star t_1' \in \overset{\bullet}{U_1}}\ (L.\ B.3(iv))$$

∎

The following theorem is then straightforward:

THEOREM 6.13

$$\frac{U_0 \sqsupseteq_{wpb} U_1}{U_0 \sqsupseteq_{wb_\bullet} U_1}$$

■

We now show that $WB_\bullet$-refinement satisfies the WPB-elimination rule. In line with propositions 3.14 and 5.15, we will require an auxiliary lemma for this proof demonstrating that $WB_\bullet$-refinement guarantees that preconditions do not strengthen in the presence of backward simulation. It is again a result coming from a wider investigation showing that $WB_\bullet$-refinement and SB-refinement are equivalent.[11]

LEMMA 6.14

$$\frac{U_0 \sqsupseteq_{wb_\bullet} U_1 \quad t \star z' \in S \vdash \mathit{Pre}\ U_1\ z}{\mathit{Pre}\ U_0\ t}$$

PROOF.

$$\dfrac{U_0 \sqsupseteq_{wb_\bullet} U_1 \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg\ \mathit{Pre}\ U_0\ t}\ (1) \quad \dfrac{\dfrac{}{t \in T_0}\ (T)}{t \in T_{0_\perp}}}{t \star \perp' \in \overset{\bullet}{U_0}}\ (L.\ B.3(iii)) \qquad \dfrac{}{\perp \star \perp' \in \overset{\circ}{S}}\ (L.\ 4.5(ii))}{t \star \perp' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S}}}{t \star \perp' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}} \qquad \begin{array}{c}\delta \\ \vdots \\ \mathit{false}\end{array}}{\dfrac{\mathit{false}}{\mathit{Pre}\ U_0\ t}\ (1)}\ (2)$$

Where δ stands for the following branch:

$$\dfrac{\dfrac{}{y \star \perp' \in \overset{\bullet}{U_1}}\ (2) \qquad \begin{array}{c}\dfrac{\dfrac{}{t \star y' \in \overset{\circ}{S}}\ (2) \quad \dfrac{\dfrac{}{t \in T_0}\ (T)}{t \neq \perp}}{t \star y' \in S} \\ \vdots \\ \mathit{Pre}\ U_1\ y\end{array}}{\dfrac{y \star \perp' \in U_1}{\mathit{false}}\ (L.\ A.5)}$$

■

Note that the ability to distinguish between $\mathcal{Z}_\mathcal{C}$ and $\mathcal{Z}_\mathcal{C}^\perp$ types is a crucial factor in the above proof: the proof steps labelled (T) denote the use of proposition 2.3 in [25]. This is an admissible axiom for $\mathcal{Z}_\mathcal{C}$, in which every term of type T is a member of the corresponding *carrier set*[12]. It is not admissible for $\mathcal{Z}_\mathcal{C}^\perp$ as terms may involve $\perp$

[11] A thorough investigation of backwards simulation data-refinement will be reported in part III of this series of papers.

[12] In $\mathcal{Z}_\mathcal{C}^\perp$ we call these *natural carrier sets* which explicitly exclude bindings that contain at least one observation bound to $\perp$ (see appendix A for further detail).

values. Hence, this proof step is valid because SB-refinement is defined in $\mathcal{Z}_C$ (section 6.2).

PROPOSITION 6.15
The following rule is derivable:

$$\frac{U_0 \sqsupseteq_{wb\bullet} U_1 \quad t \in S[wp\ U_1\ C]}{t \in wp\ U_0\ S'[C]}$$

PROOF.

$$\dfrac{\dfrac{\delta_0}{Pre\ U_0\ t} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{t_0' \in Post\ U_0\ t}\ (1)}{t \star t_0' \in U_0}\ (L.\ B.3(i))}{t \star t_0' \in \overset{\bullet}{U_0}} \qquad \dfrac{\dfrac{}{t_0' \star t_1 \in S'}\ (3) \quad \dfrac{}{t_0 \star t_1' \in S}\ (L.\ 4.5(i))}{t_0 \star t_1' \in \overset{\circ}{S}}}{t \star t_1' \in \overset{\bullet}{U_0} \mathbin{\overset{\circ}{\circ}} \overset{\circ}{S}} \qquad \dfrac{\dfrac{\delta_1}{t_1' \in C}}{t_1' \in C}\ (4)}{t \star t_1' \in \overset{\circ}{S} \mathbin{\overset{\bullet}{\circ}} \overset{\bullet}{U_1}}}{t_1' \in C}\ (3)}{t_0' \in S'[C]}\ (1)}{t \in wp\ U_0\ S'[C]}$$

Where δ_0 is:

$$\dfrac{\dfrac{U_0 \sqsupseteq_{wb\bullet} U_1 \qquad \dfrac{\dfrac{t \in S[wp\ U_1\ C] \quad \dfrac{}{t \star t_2' \in S}\ (2)}{t_2 \in wp\ U_1\ C}}{Pre\ U_1\ t_2}}{Pre\ U_0\ t}\ (2)\ (L.\ 6.14)}{}$$

and δ_1 is:

$$\dfrac{\dfrac{y \in wp\ U_1\ C \qquad \dfrac{\dfrac{\dfrac{}{y \star t_1' \in \overset{\bullet}{U_1}}\ (4) \quad \dfrac{\vdots\ \alpha}{y \in wp\ U_1\ C}}{Pre\ U_1\ y}}{y \star t_1' \in U_1}}{t_1' \in Post\ U_1\ y}}{t_1' \in C}$$

Where α stands for the following branch:

$$\dfrac{t \in S[wp\ U_1\ C] \qquad \dfrac{\dfrac{\dfrac{}{t \star y' \in \overset{\circ}{S}}\ (4) \quad \dfrac{t \in S[wp\ U_1\ C]}{t \in T_0}}{t \neq \bot}}{t \star y' \in S}}{y \in wp\ U_1\ C}$$

∎

The following theorem is immediate:

THEOREM 6.16

$$\frac{U_0 \sqsupseteq_{wb\bullet} U_1}{U_0 \sqsupseteq_{wpb} U_1}$$

■

Together theorems 6.13 and 6.16 show that WPB-refinement and $WB_\bullet$-refinement are equivalent.
In conclusion, we have established that the theories of WPB-refinement, SB-refinement and $WB_\bullet$-refinement are all equivalent.

7 Conclusions and future work

In this paper, we introduced a weakest precondition interpretation for specifications whose semantics is based on partial relations. This was influenced, at least in part, by early work carried out by [27, 28] involving partial relation semantics and weakest preconditions as developed in [14] and thoroughly discussed in, for example, [19], [22] and [15]. We established three refinement characterisations based on this interpretation: *WP-refinement* characterising operation refinement, *WPF-refinement* characterising data-refinement with forward simulation and *WPB-refinement* characterising data-refinement with backward simulation. By reformulating these as theories, rather than as sufficient conditions for refinement, we established a framework, based on the logic for Z, which allowed a comprehensive analysis of their relationships with other notions of refinement. We demonstrated, in sections 3, 5 and 6, that each of the refinement theories based on the weakest precondition semantics is equivalent to a corresponding standard approach based on lifted-totalisation. We also indicated that each of them is equivalent to a corresponding proof-theoretic notion (one of the "S" family of refinements) characterising refinement in terms of natural properties. The central question we posed earlier has been answered by the investigation: we see now how simulations interact with the postconditions in a weakest precondition framework for data-refinement based on partial relation semantics.
Each member of the "S" family of refinements is an entirely proof-theoretic notion, characterising refinement directly in terms of the language and the behaviour of preconditions and two basic observations regarding the properties one expects in a refinement: preconditions do not strengthen and postconditions do not weaken. In data-refinement, these two properties must hold in the presence of forward simulation (SF-refinement) and backward simulation (SB-refinement). By proving equivalence between members of the "WP" family and respective members of both the "S" family and the "$W_\bullet$" family of refinements, we showed, indirectly, that the standard relational completion approaches are equivalent to the fundamental proof-theoretic characterisations, each of which guarantees the two rudimentary properties above. In fact we have seen that the $\perp$ value, underlying the standard model-theoretic approaches, plays a crucial role in assuring that preconditions do not strengthen during refinement (lemmas 3.13, 5.14 and 6.14). Indeed we will, in the third part of this series of papers, discuss this in detail and examine both the conceptual and the mathematical roles of the $\perp$ value in model-theoretic data-refinement, the extent to which the standard relational completion approach is canonical, and the extent to which it is arbitrary. This is accomplished by generalising the wide spectrum of model-theoretic operation refinement characterisations given in [12] to forward and backward simulation theories. These are based on other notions of relational completion such as the *strict* (chaotic) lifted totalisation [5] and the *non-lifted* totalisation (based on a revised notion of preconditions) [12]. In this work we advocate a different approach

from [41] and [11] by taking SF-refinement and SB-refinement to be the fundamental characterisations of refinement, rather than (what we denote as) $\mathrm{WF}_\bullet$-refinement and $\mathrm{WB}_\bullet$-refinement. In other words, we take SX-refinement (were "X" stands for the *orientation* of the simulation) as *normative*: this is our prescription for data-refinement, and another X-theory is acceptable providing it is at least sound with respect to its normative counterpart. Such an approach has two major advantages: first, we establish a clear normative framework based on unquestionable properties: we shall see that whenever a potential theory fails to be sound with respect to the normative theory, we can pinpoint the grounds for that failure, in terms of the two basic properties concerning preconditions and postconditions and this aids us in isolating the problem and to construct representative counterexamples, illuminating relational completion, in general, and the non-lifted totalisation underlying data-refinement, in particular. Second, having a normative theory for investigating the relationships amongst various candidate theories evidently simplifies the process by identifying similarities in the proofs and thus avoiding unnecessary repetitions; it also enables us to compare details of the proofs and therefore to illuminate phenomena emerging from mathematical consequences of the various methods for *lifting* the underlying relations.
Finally, another interesting purely mathematical observation arising from several proofs in sections 3, 5 and 6 are the several uses of the *law of excluded middle*. We suspect that these results are strictly classical, and there appear to be many other examples of this in refinement theory: abandoning the *constructive approach* for Z which was investigated in, for example, [23] and [24] may then be inevitable.

8 Acknowledgements

Moshe Deutsch is supported by the British Council through an ORS award. Special thanks for particularly important discussions and comments go to Steve Reeves, Ray Turner, Lindsay Groves, Greg Reeve, David Streader, Mark Utting, Sam Steel and Jim Woodcock.

References

[1] J. R. Abrial. *The B-Book*. Cambridge University Press, 1996.
[2] R. J. R. Back and J. von Wright. Refinement calculus part I: Sequential nondeterministic programs. In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, *Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, volume 430 of *Lecture Notes in Computer Science*, pages 42–66. Springer Verlag, 1989.
[3] R. J. R. Back and J. von Wright. *Refinement Calculus: A Systematic Introduction*. Springer, 1998.
[4] C. Bolton, J. Davies, and J. C. P. Woodcock. On the refinement and simulation of data types and processes. In K. Araki, A. Galloway, and K. Taguchi, editors, *Integrated Formal Methods (IFM'99)*. Springer, 1999.
[5] A. Cavalcanti and J. C. P. Woodcock. A weakest precondition semantics for Z. *Technical Monograph PRG-TR-16-97. Oxford University Computing Laboratory*, 1997.
[6] A. Cavalcanti and J. C. P. Woodcock. A weakest precondition semantics for Z. *The Computer Journal*, 41(1):1–15, 1998.
[7] A. Cavalcanti and J. C. P. Woodcock. ZRC – a refinement calculus for Z. *Formal Aspects of Computing*, 10(3):267–289, 1998.
[8] W. P. de Roever and K. Engelhardt. *Data Refinement: Model-Oriented Proof Methods and Their Comparison*. Prentice Hall International, 1998.

[9] J. Derrick. A single complete refinement rule for Z. *Logic and Computation*, 10(5):663–675, October 2000.
[10] J. Derrick and E. Boiten. Calculating upward and downward simulations of state-based specifications. *Information and Software Technology*, 41:917–923, July 1999.
[11] J. Derrick and E. Boiten. *Refinement in Z and Object-Z: Foundations and Advanced Applications.* Formal Approaches to Computing and Information Technology – FACIT. Springer, May 2001.
[12] M. Deutsch, M. C. Henson, and S. Reeves. An analysis of total correctness refinement models for partial relation semantics I. *University of Essex, technical report CSM-362*, 2001. To appear in the Logic Journal of the IGPL.
[13] M. Deutsch, M. C. Henson, and S. Reeves. Results on formal stepwise design in Z. In *9th Asia Pacific Software Engineering Conference (APSEC 2002)*, pages 33–42. IEEE Computer Society Press, December 2002.
[14] E. W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, 1976.
[15] E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semantics.* Springer-Verlag, 1990.
[16] A. Diller. *Z: An Introduction to Formal Methods.* J. Wiley and Sons, 2nd edition, 1994.
[17] P. H. B. Gardiner and C. C. Morgan. Data refinement of predicate transformers. *Theoretical Computer Science*, 87(1):143–162, 1991.
[18] P. H. B. Gardiner and C. C. Morgan. A single complete rule for data refinement. *Formal Aspects of Computing*, 5:367–382, 1993.
[19] D. Gries. *The Science of Programming.* Springer-Verlag, 1981.
[20] J. He and C.A.R Hoare. Prespecification and data refinement. In *Data Refinement in a Categorical Setting*, Technical Monograph PRG-90. Oxford University Computing Laboratory, 1990.
[21] J. He, C.A.R Hoare, and J.W. Sanders. Data refinement refined. In G. Goos and J. Hartmanis, editors, *European Symposium on Programming (ESOP '86)*, volume 213 of *Lecture Notes in Computer Science*, pages 187–196. Springer-Verlag, 1986.
[22] E. C. R. Hehner. *The Logic of Programming.* Prentice Hall International, 1984.
[23] M. C. Henson and S. Reeves. Constructive foundations for Z. In S. Kuru, M. U. Caglayan, and H. L. Akin, editors, *Proc. 12th International Symposium on Computer and Information Sciences: ISCIS XII*, pages 585–591. Bogazici University press, 1997.
[24] M. C. Henson and S. Reeves. New foundations for Z. In J. Grundy, M. Schwenke, and T. Vickers, editors, *Proc. International Refinement Workshop and Formal Methods Pacific '98*, pages 165–179. Springer, 1998.
[25] M. C. Henson and S. Reeves. Investigating Z. *Logic and Computation*, 10(1):43–73, 2000.
[26] J. Jacky. *The Way of Z: Practical Programming with Formal Methods.* Cambridge University Press, 1997.
[27] M. B. Josephs. Formal methods for stepwise refinement in the Z specification language. *Technical Monograph PRG-TR-1-86. Oxford University Computing Laboratory*, 1986.
[28] M. B. Josephs. The data refinement calculator for Z specifications. *Information Processing Letters*, 27:29–33, 1988.
[29] S. King. Z and the Refinement Calculus. In D. Bjørner, C. A. R. Hoare, and H. Langmaack, editors, *VDM '90 VDM and Z – Formal Methods in Software Development*, volume 428 of *Lecture Notes in Computer Science*, pages 164–188. Springer-Verlag, April 1990.
[30] R. Miarka, E. Boiten, and J. Derrick. Guards, preconditions, and refinement in Z. In J. P. Bowen, S. Dunne, A. Galloway, and S. King, editors, *ZB2000: Formal Specification and Development in Z and B*, volume 1878 of *Lecture Notes in Computer Science*, pages 286–303. Springer-Verlag, August 2000.
[31] C. C. Morgan. The specification statement. *ACM Transactions on Programming Languages and Systems*, 10:403–419, 1988.
[32] C. C. Morgan. *Programming from Specifications.* Prentice Hall International, 2nd edition, 1994.
[33] C. C. Morgan and P. H. B. Gardiner. Data refinement by calculation. *Acta Informatica*, 27(6):481–503, 1989.

[34] C. C. Morgan and K. Robinson. Specification statements and refinement. In C. C. Morgan, K. Robinson, and P. H. B. Gardiner, editors, *On the Refinement Calculus*, Technical Monograph PRG-70. Oxford University Computing Laboratory, pages 31–57, 1988.
[35] B. Potter, J. Sinclair, and D. Till. *An Introduction to Formal Specification and Z.* Prentice Hall, 2nd edition, 1996.
[36] V. Pratt. Origins of the calculus of binary relations. In *Proc. 7th Annual IEEE Symposium on Logic in Computer Science (LICS '92)*, pages 248–254. IEEE Computer Society Press. Santa Cruz, CA, June 1992.
[37] S. Schneider. *The B-Method – An Introduction.* Correctness of Computing. Palgrave, 2001.
[38] S. Schneider. Private communication. Royal Holloway College, University of London, January 2002.
[39] J. M. Spivey. *The Z Notation: A Reference Manual.* Prentice Hall, 2nd edition, 1992.
[40] N. W. Tennant. *Natural Logic.* Edinburgh University Press, 2nd edition, 1990.
[41] J. C. P. Woodcock and J. Davies. *Using Z: Specification, Refinement and Proof.* Prentice Hall, 1996.
[42] J. C. P. Woodcock and C. C. Morgan. Refinement of state-based concurrent systems. In D. Bjørner, C. A. R. Hoare, and H. Langmaack, editors, *VDM '90 VDM and Z – Formal Methods in Software Development*, volume 428 of *Lecture Notes in Computer Science*, pages 340–351. Springer-Verlag, April 1990.
[43] J. B. Wordsworth. *Software Development with Z – A Practical Approach to Formal Methods in Software Engineering.* Internalional Computer Science Series. Addison-Wesley, 1992.

A Specification logic - a synopsis

In this appendix, we will revise a little Z logic, settling some notational conventions in the process. This is included for convenience only and the reader may need to consult [25] and [12] at least in order to fully understand our notational and meta-notational conventions. Our mathematical account takes place in a simple conservative extension $\mathcal{Z}_C^{\perp}$ of $\mathcal{Z}_C$, the core Z-logic of [25]. This provides a convenient basis, in particular a satisfactory logical account of the schema calculus, upon which the present work can be formalised.

A.1 Schemas

$\mathcal{Z}_C$ is a typed theory in which the types of higher-order logic are extended with *schema types* whose values are unordered, label-indexed tuples called *bindings*. For example, if the T_i are types and the $\mathbf{z}_i$ are labels (constants) then:

$$[\cdots \mathbf{z}_i : T_i \cdots]$$

is a (schema) type. Values of this type are bindings, of the form:

$$\langle\!| \cdots \mathbf{z}_i \Rrightarrow t_i \cdots |\!\rangle$$

where the term t_i has type T_i.
The symbols $\preceq$, $\curlywedge$ and $\curlyvee$ denote the *schema subtype* relation, and the operations of *schema type intersection* and (compatible) *schema type union*. We let U (with diacriticals when necessary) range over operation schema expressions. These are sets of bindings linking, as usual before (unprimed) observations with after (primed) observations. This captures the informal account to be found in the literature (*e.g.* [16, 35, 41]). We can always, then, write the type of such operation schemas as $\mathbb{P}(T^{in} \curlyvee T^{out'})$ where T^{in} is the type of the input sub-binding and $T^{out'}$ is the type of the output sub-binding. We also permit *binding concatenation*, written $t_0 \star t_1$ when the alphabets of t_0 and t_1 are disjoint. This is, in fact, exclusively used for partitioning bindings in operation schemas into before and after

components, so the terms involved are necessarily disjoint. We lift this operation to sets (of appropriate type):

$$C_0 \star C_1 =_{df} \{z_0 \star z_1 \mid z_0 \in C_0 \wedge z_1 \in C_1\}$$

The same restriction obviously applies here: the types of the sets involved must be disjoint. In this way reasoning in Z becomes no more complex than reasoning with binary relations. We simplify the introduction and elimination rules for schema composition (provided in [25]). A composition of two operation schemas $U_0 \fatsemi U_1$ has the type $\mathbb{P}(T_0 \curlyvee T_1')$ where, as expected, U_0 is of type $\mathbb{P}(T_0 \curlyvee T_2')$ and U_1 is of type $\mathbb{P}(T_2 \curlyvee T_1')$.

PROPOSITION A.1
The following rules are derivable (the usual sideconditions apply to the eigenvariable y):

$$\frac{t_0 \star t_2' \in U_0 \quad t_2 \star t_1' \in U_1}{t_0 \star t_1' \in U_0 \fatsemi U_1}\ (U_{\fatsemi}^{+}) \qquad \frac{t_0 \star t_1' \in U_0 \fatsemi U_1 \quad t_0 \star y' \in U_0, y \star t_1' \in U_1 \vdash P}{P}\ (U_{\fatsemi}^{-})$$

∎

We introduce a notational convention in order to avoid the repeated use of filtering in the context of equality propositions.

DEFINITION A.2
Let $T \preceq T_0$ and $T \preceq T_1$.

$$t_0^{T_0} =_T t_1^{T_1} =_{df} t_0 \upharpoonright T = t_1 \upharpoonright T$$

The only modification we need to make in $\mathcal{Z}_C^{\perp}$ is to include the new distinguished terms which are explicitly needed in the approach taken in [41]. Specifically: the types of $\mathcal{Z}_C$ are extended to include terms $\perp^T$ for every type T. There are, additionally, a number of axioms which ensure that all the new $\perp^T$ values interact properly, e.g.

$$\overline{\perp^{[\mathtt{z}_0:\mathtt{T}_0 \cdots \mathtt{z}_n:\mathtt{T}_n]} = \langle\!| \mathtt{z}_0 \Rrightarrow \perp^{T_0} \cdots \mathtt{z}_n \Rrightarrow \perp^{T_n} |\!\rangle}$$

In other words, $\perp^{[\mathtt{z}_0:\mathtt{T}_0 \cdots \mathtt{z}_n:\mathtt{T}_n]}.\mathtt{z}_i = \perp^{T_i}$ $(0 \leq i \leq n)$. Note that this is the *only* axiom concerning distinguished bindings, hence, binding construction is *non-strict* with respect to the $\perp^T$ values.
Finally, the extension of $\mathcal{Z}_C^{\perp}$ which introduces schemas as sets of bindings and the various operators of the schema calculus is undertaken as usual (see [25]) but the carrier sets of the types must be adjusted to form what we call the *natural carrier sets* which are those sets of elements of types which *explicitly exclude* the $\perp^T$ values:

DEFINITION A.3
Natural carriers for each type are defined by closing: $\mathbb{N} =_{df} \{z^{\mathbb{N}} \mid z \neq \perp^{\mathbb{N}}\}$ under the operations of cartesian product, powerset and schema set.[13]

As a result the schema calculus is *hereditarily* $\perp$*-free*:

DEFINITION A.4 (Semantics for atomic schemas)

$$[T \mid P] =_{df} \{z \in T \mid z.P\}$$

Note that this definition draws bindings from the natural carrier of the type T. As a consequence, writing $t(\perp)$ for any binding satisfying $t.\mathtt{x} = \perp$, for some observation $\mathtt{x}$, we have:

LEMMA A.5

$$\frac{t(\perp) \in U}{\mathit{false}}$$

∎

[13] The notational ambiguity does not introduce a problem, since only a set can appear in a term or proposition, and only a type can appear as a superscript.

We will also need the *extended carriers*. These are defined for all types as follows:

DEFINITION A.6

$$T_\perp =_{df} T \cup \{\perp^T\}$$

A.2 Preconditions

We can formalise the idea of the *precondition* of an operation schema (the domain of the relation between before and after states that the schema denotes) to express the partiality involved.

DEFINITION A.7
Let $T^{in} \preceq V$.

$$Pre\ U\ x^V =_{df} \exists z \in U \bullet x =_{T^{in}} z$$

Clearly, the preconditions of on an operation schema, in general, will not be the whole of T^{in}. It is in this sense that operation schemas denote *partial relations*.

B Relational completion

In this appendix, we review the chaotic relational completion operator (lifted totalisation) discussed in [12] and [13]. We define this operator in line with the intentions described in [41], chapter 16. We will write $T^\star$ for the set $T^{in}_\perp \star T^{out'}_\perp$.

DEFINITION B.1

$$\overset{\bullet}{U} =_{df} \{z_0 \star z_1' \in T^\star \mid Pre\ U\ z_0 \Rightarrow z_0 \star z_1' \in U\}$$

Then the following introduction and elimination rules are derivable:

PROPOSITION B.2

$$\frac{t_0 \star t_1' \in T^\star \quad Pre\ U\ t_0 \vdash t_0 \star t_1' \in U}{t_0 \star t_1' \in \overset{\bullet}{U}}\ (\bullet^+) \qquad \frac{t_0 \star t_1' \in \overset{\bullet}{U} \quad Pre\ U\ t_0}{t_0 \star t_1' \in U}\ (\bullet^-_o)$$

$$\frac{t_0 \star t_1' \in \overset{\bullet}{U}}{t_0 \star t_1' \in T^\star}\ (\bullet^-_1)$$

∎

LEMMA B.3
The following extra rules are derivable for lifted-totalised sets:

$$\frac{}{U \subseteq \overset{\bullet}{U}}\ (i) \qquad \frac{}{\perp \in \overset{\bullet}{U}}\ (ii) \qquad \frac{\neg\ Pre\ U\ t \quad t \in T^{in}_\perp}{t \star \perp' \in \overset{\bullet}{U}}\ (iii)$$

$$\frac{\neg Pre\ U\ t_0 \quad t_0 \in T^{in}_\perp \quad t_1' \in T^{out'}_\perp}{t_0 \star t_1' \in \overset{\bullet}{U}}\ (iv)$$

∎

Lemmas B.3(i – iv) demonstrate that definition B.1 is consistent with the intentions described in [41] (chapter 16) and [11] (chapter 2): the underlying partial relation is contained in the completion; the $\perp$ element is present in the relation, and more generally, each value outside the precondition maps to every value in the range of the relation, including $\perp$.