Four Theories for Backward Simulation Data-Refinement

Moshe Deutsch and Martin C. Henson

Department of Computer Science, University of Essex, UK.
{mdeuts, hensm}@essex.ac.uk

Abstract. This paper investigates data-refinement by backward simulation. We introduce four distinct theories and demonstrate that these are all equivalent. One theory, SB-refinement, serves as a normative characterisation of backwards refinement. The other model-theoretic characterisations illuminate the standard approach involving non-strictly lifted relational completion operators.

1 Introduction

Refinement underlies the *transformational software process model*, in which design decisions are incorporated into an initial abstract specification deriving, in stages, more concrete versions. In *data-refinement*, the objective is to transform a data type into a form closer to an implementation: the underlying data space is refined along with the operation. This process is sometimes called data design [21]. There are two refinement techniques which enable us to verify such transformations: *forward simulation* and *backward simulation* [19, 4]. These are known to be sound and *jointly* complete [8, 20] and can also be formulated as theories of refinement in their own right.

In this paper, we consider four data-refinement theories, confining attention to the theory induced by backward simulation. These constitute generalisations of various operation refinement theories explored in [5] and [6] of which two are related to previous work [19, 4]. No systematic investigation or results concerning the relationships between them, have been presented or published before. We will show that all these theories are equivalent.

We begin by introducing the notion of data simulations that underlies the forward and backward simulation refinement techniques (section 2), including the *lifted* simulations used in refinement based on relational completion operators (discussed in, for example, [19, 4] and investigated in detail in [5, 6]). These involve an additional distinguished element, called *bottom* and written $\perp$. We then define three alternative characterisations of data-refinement (section 3) based on two distinct relational completion models discussed in [5] (see also appendix B). We show that all three are equivalent to a purely proof theoretic characterisation of backward simulation refinement (section 4). This fourth theory, SB-refinement, captures backward simulation data-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, not involving the introduction of either an auxiliary semantics, nor the introduction of an auxiliary element.

Our approach, sheds light on the role of "lifting" in data-refinement based on relational completion models. In addition, we establish SB-refinement, a normative theory for exploring the validity of refinement approaches based on "backward" simulation.
Our investigation takes place in $\mathcal{Z}_C$, the logic for Z reported in [12] and a simple *conservative extension* $\mathcal{Z}_C^{\perp}$ [5] which incorporates $\perp$ into the types of $\mathcal{Z}_C$ (we summarise this, and additional notational conventions in appendix A). This allows us to work with Z schemas as easily as with abstract relations. Nothing we show here is specifically confined to Z; we use this merely as a linguistic vehicle for state-based specification. We employ a novel technique of rendering all the theories of refinement as sets of introduction and elimination rules. This leads to a uniform and simple method for proving the various equivalence results. As such, it contrasts with the more semantic based techniques employed in [2].

2 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* [3]. In this section we revise these and introduce some essential material underlying our investigation.
A data simulation [19, 21] is a relation between an abstract data space and a concrete counterpart. Data simulations[1] underly two refinement techniques which enable us to verify data-refinement, as shown by the two semi-commuting diagrams in Fig. 1. Both

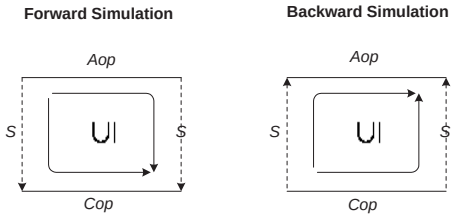

Fig. 1. Forward simulation and backward simulation refinement techniques. *Aop* and *Cop* represent the abstract and concrete operations (respectively), whereas *S* represents the simulation. Note that a forward simulation is oriented (by composition) from the abstract to the concrete data space and,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

forward and backward simulation[2] refinement techniques are known to be sound but neither of them is singly complete. However, they are known to be *jointly complete* [20].

[1] 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.

[2] Forward and backward simulations are also respectively known as *downward* and *upward* simulations [3, 4, 9] due to their directions in the commuting diagrams in Fig. 1.

We will use the meta-variables[3] U_0 and U_1 to range over our specifications. In this paper U_0 will always be the concrete operation and U_1 the abstract operation. We adopt the approach taken in [2]: our concrete relation is always drawn from $\mathbb{P}(T_0 \times T_0')$ and the abstract relation from $\mathbb{P}(T_1 \times T_1')$. A backward simulation (concrete to abstract) belongs to $\mathbb{P}(T_0 \times T_1')$.

We will need to incorporate the $\bot$ element in a simulation used with lifted-totalised operations (see appendix A and [5, 6]). Naturally, Woodcock's chaotic totalisation [19] is unacceptable here, as this might enforce a link between abstract and concrete states that are not supposed to be linked. The conventional approach [19, 4] is to (non-strictly) lift[4] $\bot$ in the input set of the simulation, thus retaining its partiality. This leads to the following definition:

Definition 1 (Non-Strictly Lifted Backward Simulation).

$$S^{\mathbb{P}(T_0 \times T_1')}_{\circ} =_{df} \{z_0 \star z_1' \in T_{0_\bot} \star T_{1_\bot}' \mid z_0 \neq \bot \Rightarrow z_0 \star z_1' \in S\}$$

Then the following introduction and elimination rules are derivable:

Proposition 1.

$$\frac{t_0 \star t_1' \in T_{0_\bot} \star T_{1_\bot}' \quad t_0 \neq \bot \vdash t_0 \star t_1' \in S}{t_0 \star t_1' \in \overset{\circ}{S}} \ (\circ^+) \qquad \frac{t_0 \star t_1' \in \overset{\circ}{S} \quad t_0 \neq \bot}{t_0 \star t_1' \in S} \ (\circ_0^-)$$

$$\frac{t_0 \star t_1' \in \overset{\circ}{S}}{t_0 \star t_1' \in T_{0_\bot} \star T_{1_\bot}'} \ (\circ_1^-)$$

□

Lemma 1. *The following additional rules are derivable for non-strictly lifted simulations:*

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

□

Lemmas 1(i – iv) demonstrate that definition 1 is consistent with the intentions described in [19] and [4]: the underlying partial relation is contained in the lifting; the $\bot$ element is present in the relation and is mapped onto every after state, and no other initial state is so. This raises an immediate question: why does the lifting of the simulation have to be non-strict with respect to $\bot$? This issue was not explored in [19, 4], where the non-strict lifting of the simulation is taken as self-evident. We will gradually provide an answer to this question in the sequel. For that, we will need the definition of a strictly lifted simulation:

[3] We provide some notational conventions in appendix A.

[4] Lifting signifies mapping $\bot$ of the input set of the relation onto all states of the output set. In general, the notion of strictness discussed in this paper is with respect to $\bot$; therefore, strict lifting denotes mapping $\bot$ onto only its output counterpart.

Definition 2 (Strictly Lifted Backward Simulation).

$$S^{\mathbb{P}(\vec{T_0} \vee T_1')} =_{df} \{z_0 \star z_1' \in T_{0_\perp} \star T'_{1_\perp} \mid (z_0 \neq \perp \Rightarrow z_0 \star z_1' \in S) \wedge (z_0 = \perp \Rightarrow z_1' = \perp')\}$$

Obvious introduction and elimination rules follow from this.

Lemma 2. *The following additional rules are derivable for strictly lifted simulations:*

$$\frac{}{S \subseteq \vec{S}}\ (i) \qquad \frac{}{\vec{S} \subseteq \mathring{S}}\ (ii) \qquad \frac{}{\perp \in \vec{S}}\ (iii)$$

$$\frac{t_0 \star t_1' \in \vec{S} \quad t_1' = \perp'}{t_0 = \perp}\ (iv) \qquad \frac{t_0 \star t_1' \in \vec{S} \quad t_1' \neq \perp'}{t_0 \star t_1' \in S}\ (v)$$

□

Lemmas 2(iv – v) embody the strictness captured by definition 2: if the after state is $\perp$ then the initial state must also be $\perp$, and if it is not $\perp$ then the initial state was not either.

3 Data-Refinement with Backward Simulation

In [5] and [6] we investigated operation refinement for specifications whose semantics is given by partial relation semantics (again using Z as an example). We compared three characterisations of *operation refinement*: S-refinement, a proof theoretic characterisation closely connected to refinement as introduced by Spivey [16]; $W_\bullet$-refinement, based on Woodcock's relational completion operator [19]; and $W_\ominus$-refinement based on a strict relational completion operator (see appendix B). We proved that all these refinement theories are equivalent. The investigation also illuminated the crucial role of $\perp$ in total correctness operation refinement.
In this section, we provide four distinct notions of data refinement, generalising the above operation refinement characterisations based on backward simulation. We will then go on to compare them thus providing a complementary investigation to that in [5] and [6].

3.1 SB-Refinement

In this section, we introduce a purely proof theoretic characterisation of backward simulation refinement, which is closely connected to sufficient refinement conditions introduced by Woodcock [19, p.270] (indicated as "B-corr") and by Derrick and Boiten [4, p.93]. These conditions correspond to the premises of our introduction rule for SB-refinement.
This generalisation of S-refinement [5, 6] is based on two properties expected in a refinement: 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). In this case these two properties must hold in the presence of a simulation.

The notion can be captured by forcing the refinement relation to hold *exactly* when these conditions apply. SB-refinement is written $U_0 \stackrel{s}{\sqsupseteq}_{sb} U_1$ (U_0 SB-refines U_1 with respect to the simulation S)[5] and is given by the $\mathcal{Z}_C$ definition that leads directly to the following rules:

Proposition 2. *Let x, x_0, x_1, z, z_0 be fresh variables.*

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

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

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

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 theory $\mathcal{Z}_C$. We take SB-refinement as *normative*: this is our prescription for data-refinement, and another theory is acceptable providing it is at least sound with respect to it.

3.2 Relational Completion Based Refinement

We now introduce three backward simulation refinement theories in the extended framework $\mathcal{Z}_C^{\perp}$. These are based on the two distinct notions of the schema lifted-totalisation set out in appendix B. Each of them captures, schematically, the backward simulation commuting diagram in Fig. 1 and is based on *schema* or, more generally, *relational composition* (see appendix A, proposition 7).

WB$_\bullet$-Refinement. This notion of refinement is also discussed in [19, p.247] and [3]. It is written $U_0 \stackrel{s}{\sqsupseteq}_{wb_\bullet} U_1$ and is defined as follows:

Definition 3. $U_0 \stackrel{s}{\sqsupseteq}_{wb_\bullet} U_1 =_{df} \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S} \subseteq \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}$

The following introduction and elimination rules are immediately derivable for WB$_\bullet$-refinement:

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

Proposition 3. *Let z_0, z_1 be fresh.*

$$\frac{z_0 \star z_1' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S} \vdash z_0 \star z_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}{U_0 \sqsupseteq_{wb_\bullet} U_1} \ (\sqsupseteq^{+}_{wb_\bullet}) \qquad \frac{U_0 \sqsupseteq_{wb_\bullet} U_1 \quad t_0 \star t_1' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S}}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}} \ (\sqsupseteq^{-}_{wb_\bullet})$$

□

WB$_\Phi$-Refinement. The natural generalisation of W$_\ominus$-refinement [5] (at least in the light of the standard literature) is to use strict-lifted totalised operations, yet a non-strict lifted simulation. We name this WB$_\Phi$-refinement, written $U_0 \overset{s}{\sqsupseteq}_{wb_\Phi} U_1$ and defined as follows:

Definition 4. $U_0 \overset{s}{\sqsupseteq}_{wb_\Phi} U_1 =_{df} \overset{\ominus}{U_0} \fatsemi \overset{\circ}{S} \subseteq \overset{\circ}{S} \fatsemi \overset{\ominus}{U_1}$

Obvious introduction and elimination rules follow from this.

WB$_\ominus$-Refinement. Our third characterisation of refinement is motivated by the enquiry raised in section 2. Establishing a refinement theory, in which both the operations and the simulation are strictly lifted, provides a point of reference which will aid us in investigating two important matters: firstly, whether the strict and non-strict relational completion operators are still interchangeable underlying generalisations of data-refinement; secondly, whether the non-strict lifting of the simulation is an essential property. We name this theory WB$_\ominus$-refinement, written $U_0 \overset{s}{\sqsupseteq}_{wb_\ominus} U_1$; it is defined as follows:

Definition 5. $U_0 \overset{s}{\sqsupseteq}_{wb_\ominus} U_1 =_{df} \overset{\ominus}{U_0} \fatsemi \vec{S} \subseteq \vec{S} \fatsemi \overset{\ominus}{U_1}$

Obvious introduction and elimination rules follow from this definition.

4 Four Equivalent Theories

In this section, we demonstrate that all four theories of refinement are equivalent and we will clearly see the critical role that the $\perp$ value plays in the three model-theoretic approaches.

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 approaches as theories. 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 T_0 and T_1.

4.1 WB$_\bullet$-Refinement and SB-Refinement are Equivalent

We begin by showing that WB$_\bullet$-refinement implies SB-refinement, by proving that WB$_\bullet$-refinement satisfies both SB-refinement elimination rules. Firstly the rule for pre-conditions.

Proposition 4. *The following rule is derivable:*

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

Proof

$$\dfrac{\dfrac{U_0 \sqsupseteq_{wb_\bullet} U_1 \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg\ Pre\ U_0\ t}\ (1) \quad \dfrac{\dfrac{}{t \in T_0}\ (T)}{t \in T_{0_\perp}}\ (L.\ 4(iii))}{t \star \perp' \in \overset{\bullet}{U_0}} \quad \dfrac{}{\perp \star \perp' \in \overset{\circ}{S}}\ (L.\ 1(ii))}{t \star \perp' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S}}}{t \star \perp' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}} \qquad \begin{array}{c}\delta \\ \vdots \\ false\end{array}}{\dfrac{false}{Pre\ U_0\ t}\ (1)}\ (2)$$

Where δ stands for the following branch:

$$\dfrac{\dfrac{\dfrac{}{t \star y' \in \overset{\circ}{S}}\ (2) \quad \dfrac{\dfrac{}{t \in T_0}\ (T)}{t \neq \perp}}{t \star y' \in S} \qquad \begin{array}{c}\vdots \\ Pre\ U_1\ y\end{array}}{\dfrac{y \star \perp' \in U_1}{false}\ (L.\ 3)}$$

$\square$

There are two observations we can make from the proof. Firstly, note that the ability to distinguish between $\mathcal{Z}_C$ and $\mathcal{Z}_C^\perp$ types is a crucial factor: the proof steps labelled (T) denote the use of proposition 2.3 in [12]. This is an admissible axiom for $\mathcal{Z}_C$, in which every term of type T is a member of the corresponding *carrier set*[6]. It is not admissible for $\mathcal{Z}_C^\perp$ as terms may involve $\perp$ values. Hence, this proof step is valid because SB-refinement is defined in $\mathcal{Z}_C$ (section 3.1). Secondly, notice the explicit use of $\perp$ in the proof. This is reminiscent of our previous investigation of operation refinement, in which the explicit use of $\perp$ is critical for proving that W$_\bullet$-refinement satisfies the precondition elimination rule for S-refinement (see, for example, proposition 4.11 in [5]). Much the same observation can be made here, only that the use of lemmas 4(iii) and 1(ii) in the proof suggests that *both* the lifted-totalisation of the operations and the

[6] In $\mathcal{Z}_C^\perp$ 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).

lifting of the simulation are essential for showing that WB$_\bullet$-refinement guarantees that preconditions do not strengthen in the presence of the simulation.
Turning now to the second elimination rule in SB-refinement.

Proposition 5. *The following rule is derivable:*

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

Proof

$$\dfrac{\dfrac{U_0 \sqsupseteq_{wb_\bullet} U_1 \qquad \dfrac{\dfrac{t_0 \star t_1' \in U_0}{t_0 \star t_1' \in \overset{\bullet}{U_0}}\ (L.\ 4(i)) \qquad \dfrac{t_1 \star t_2' \in S}{t_1 \star t_2' \in \overset{\circ}{S}}\ (L.\ 1(i))}{t_0 \star t_2' \in \overset{\bullet}{U_0} \fatsemi \overset{\circ}{S}} \qquad \overset{\delta}{\vdots}}{t_0 \star t_2' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1} \qquad P}}{P}\ (1)$$

Where δ stands for the following branch:

$$\dfrac{\dfrac{\dfrac{}{t_0 \star y' \in \overset{\circ}{S}}\ (1) \qquad \dfrac{t_0 \star t_1' \in U_0}{t_0 \neq \bot}\ (L.\ 3)}{t_0 \star y' \in S} \qquad \overset{\vdots}{Pre\ U_1\ y}}{t_0 \star y' \in S \wedge y \star t_2' \in U_1}$$

$$\overset{\vdots}{P}$$

□

Theorem 1. $U_0 \sqsupseteq_{wb_\bullet} U_1 \Rightarrow U_0 \sqsupseteq_{sb} U_1$

Proof This follows immediately, by ($\sqsupseteq^+_{sb}$), from propositions 4 and 5[7]. □
We now show that SB-refinement satisfies the WB$_\bullet$-elimination rule.

Proposition 6.

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

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

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

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

Where δ_0 stands for the following branch:

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

Where β_0 is:

$$\dfrac{\dfrac{}{t_0 \star y' \in \overset{\bullet}{U_0}}\ (1) \qquad \dfrac{U_0 \sqsupseteq_{sb} U_1 \quad \overline{\forall z \bullet t_0 \star z' \in S \Rightarrow Pre\ U_1\ z}}{Pre\ U_0\ t_0}\ (2)}{t_0 \star y' \in U_0}$$

and β_1, β_2 are respectively:

$$\dfrac{\dfrac{}{y \star t_1' \in \overset{\circ}{S}}\ (1) \qquad \dfrac{\overset{\beta_0}{\vdots}\ t_0 \star y' \in U_0}{y \neq \bot}}{y \star t_1' \in S}\ (L.\ 3) \qquad \dfrac{\dfrac{\dfrac{\overline{t_0 \star w_0' \in S}\ (3)}{t_0 \star w_0' \in \overset{\circ}{S}}\ (L.\ 1(i)) \qquad \dfrac{\overline{w_0 \star t_1' \in U_1}\ (3)}{w_0 \star t_1' \in \overset{\bullet}{U_1}}\ (L.\ 4(i))}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}}{}$$

δ_1 stands for the following branch:

$$\dfrac{\overline{\exists z \bullet t_0 \star z' \in S \wedge \neg\ Pre\ U_1\ z}\ (2) \qquad \dfrac{\dfrac{\dfrac{\overline{t_0 \star w_1' \in S \wedge \neg\ Pre\ U_1\ w_1}\ (4)}{t_0 \star w_1' \in S}}{t_0 \star w_1' \in \overset{\circ}{S}}\ (L.\ 1(i)) \qquad \overset{\alpha}{\vdots}\ w_1 \star t_1' \in \overset{\bullet}{U_1}}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}\ (4)}{t_0 \star t_1' \in \overset{\circ}{S} \fatsemi \overset{\bullet}{U_1}}$$

Where α is:

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

□

Notice that this proof depends on use of the *law of excluded middle* (see, for example, [18]). We suspect that this result is strictly classical, and there appear to be many other examples of this in refinement theory, so abandoning the *constructive approach* for Z taken in, for example, [10, 11] and [13, 14] may be inevitable.

Theorem 2. $U_0 \sqsupseteq_{sb} U_1 \Rightarrow U_0 \sqsupseteq_{wb_\bullet} U_1$ □

Theorems 1 and 2 together establish that the theories of SB-refinement and $WB_\bullet$-refinement are equivalent.

4.2 WB_ϕ-Refinement and $WB_\ominus$-Refinement are Equivalent to SB-Refinement

We now show that both WB_ϕ-refinement and $WB_\ominus$-refinement are equivalent to SB-refinement.
Proving that WB_ϕ-refinement satisfies both SB-elimination rules leads to proofs identical to propositions 4 and 5, modulo a substitution of $\sqsupseteq_{wb_\phi}$ for $\sqsupseteq_{wb_\bullet}$, $\overset{\ominus}{U}$ for $\overset{\bullet}{U}$, applications of $(\ominus_0^-)$ for $(\bullet_0^-)$ and lemmas 5(iv) and 5(i) in place of lemmas 4(iii) and 4(i) (respectively). Likewise, proving that SB-refinement satisfies WB_ϕ-elimination rule is very similar to the proof of proposition 6. In this case, we require the same general substitutions as above, in addition to a modification in the proof branch labelled α: applying lemma 5(v) in place of lemma 4(iv) requires the variable w_1 to range over the *natural carrier set*, rather than the *extended carrier*; hence, the proof step labelled (♣) is redundant here. From this we have:

Theorem 3. $U_0 \sqsupseteq_{wb_\phi} U_1 \Leftrightarrow U_0 \sqsupseteq_{sb} U_1$ □

A very similar situation arises when we consider $WB_\ominus$-refinement. SB-refinement constitutes our common ground and again we need to make use of the substitutions and amendments to the proofs of propositions 4, 5 and 6 as we did in theorem 3 except that $\sqsupseteq_{wb_\ominus}$ replaces $\sqsupseteq_{wb_\bullet}$, $\vec{S}$ replaces $\mathring{S}$ and we apply lemmas 2(i) and 2(iii) in place of lemmas 1(i) and 1(ii) (respectively). Moreover, applications of $(\rightarrow_0^-)$ and $(\rightarrow_1^-)$ replace $(\circ_0^-)$ and $(\circ_1^-)$ (respectively). From this we immediately get implication in both directions:

Theorem 4. $U_0 \sqsupseteq_{wb_\ominus} U_1 \Leftrightarrow U_0 \sqsupseteq_{sb} U_1$ □

Despite their superficial dissimilarity, all four theories are equivalent. In establishing this we reinforce the results from [5] and [6] showing clearly the significance of $\perp$ (proposition 4). Additionally we have shown that strict lifting of both the operations and the simulation is sufficient for introducing a model based refinement theory that preserves the natural properties of SB-refinement.
The fact that, given the appropriate substitutions, the proofs in this section are identical to the ones in section 4.1 suggests that the *minimal* mathematical properties of the models, which are essential for establishing theorems 1 and 2 are the ones of $\overset{\ominus}{U}$ and $\vec{S}$. To be more specific, the use of lemma 4(iii) (proposition 4) indicates that everything outside the preconditions of the underlying operation, *including* $\perp$, should be mapped onto $\perp$ of the output set; and the use of the proof step labelled (♣) in conjunction with

lemma 4(iv) (proposition 6) indicates that everything outside the preconditions *that is not* $\bot$ should be mapped onto everything in the output set. These observations are precisely the properties of strictly-lifted totalised relations within a non-strict framework. A similar observation can be made for the simulation: the only lemma concerning the lifting of the simulation used in the proofs is 1(ii) (proposition 4); there is no evidence for using lemma 1(iii), which expresses the non-strict lifting.

5 Conclusions and Future Work

In this paper, we introduced four distinct notions of data-refinement founded upon backward simulation. By reformulating these as theories, rather than sufficient conditions, we establish a mathematical framework, based on the logic for Z, which underlies our analysis. We demonstrated that what look like different models of specification and refinement are, in fact, closely related. Having a non-model-theoretic benchmark (SB-refinement) allows us to scrutinise the role of the $\bot$ value in the model-theoretic approaches and evaluate the essence of the relational lifted-totalisation found in the literature.

SB-refinement theory is entirely proof-theoretic, 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 the presence of (backward) simulation. We advocate a different approach to [19] and [4] by taking SB-refinement to be the fundamental characterisation of refinement, rather than (what we denote as) $\mathrm{WB}_\bullet$-refinement. Such approach has two major advantages: first, we establish a clear normative framework based on unquestionable properties. We shall see in future work 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. This aids us in isolating the problem and to construct representative counterexamples, illuminating relational completion, in general, and the non-lifted totalisation (*e.g.* [5]) underlying data-refinement, in particular. Secondly, as we reported in section 4, having a normative theory for investigating the relationships amongst various candidate theories not only simplifies the process (for example, as we have seen: similarities in the proofs), it also enables us to compare details of the proofs. In this paper this has led us to the conclusion that the strict and non-strict relational completion models are interchangeable in the context of backward simulation refinement, and similarly for the strict and non-strict lifting of the simulation.

In this paper we have not provided an analysis of forward simulation data-refinement. The results in this paper cannot be taken as self-evidently analogous in a counterpart investigation of forward simulation theories. Indeed, we will, in future work, show that forward simulation refinement is less permissive: the strict and non-strict relational completion models are still interchangeable in this framework, but the strict lifting of the simulation has a restrictive effect: $\mathrm{WF}_\ominus$-refinement (the forward simulation counterpart of $\mathrm{WB}_\ominus$-refinement) is sound with respect to SF-refinement (the normative theory in this case), but it is not complete because, under certain circumstances, it *prevents weakening of preconditions*.

There is much more to say about data-refinement, particularly generalising results we have detailed in [5] in the context of operation refinement, *e.g.* formulating forward and backward simulation refinement theories based on a *sets of implementations* model or data-refinement theories based on weakest preconditions, and then exploring their relationships with SB and SF-refinement. There is an additional interesting dimension, in which we explore generalisations of *firing conditions* refinement [6, 4, 17] underlying forward and backward simulation techniques. We can investigate their relationships with a variety of data-refinement theories based on the *abortive relational completion* model as given in [6], [4] and [1].

6 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, Eerke Boiten, John Derrick, Lindsay Groves, Ralph Miarka, Greg Reeve, David Streader, Jim Woodcock and Rob Arthan.

References

1. 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.
2. W. P. de Roever and K. Engelhardt. *Data Refinement: Model-Oriented Proof Methods and Their Comparison*. Prentice Hall International, 1998.
3. J. Derrick and E. Boiten. Calculating upward and downward simulations of state-based specifications. *Information and Software Technology*, 41:917–923, July 1999.
4. 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.
5. 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.
6. 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.
7. A. Diller. *Z: An Introduction to Formal Methods*. J. Wiley and Sons, 2nd edition, 1994.
8. 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.
9. 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.
10. 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.
11. 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.

12. M. C. Henson and S. Reeves. Investigating Z. *Logic and Computation*, 10(1):43–73, 2000.
13. S. H. Mirian-Hosseinabadi. *Constructive Z.* PhD thesis, University of Essex, 1997.
14. S. H. Mirian-Hosseinabadi and R. Turner. Constructive Z. *Logic and Computation*, 8(1):49–70, 1998.
15. B. Potter, J. Sinclair, and D. Till. *An Introduction to Formal Specification and Z.* Prentice Hall, 2nd edition, 1996.
16. J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall, 2nd edition, 1992.
17. B. Strulo. How firing conditions help inheritance. In J. P. Bowen and M. G. Hinchey, editors, *ZUM '95: The Z Formal Specification Notation*, volume 967 of *Lecture Notes in Computer Science*, pages 264–275. Springer Verlag, 1995.
18. N. W. Tennant. *Natural Logic*. Edinburgh University Press, 2nd edition, 1990.
19. J. C. P. Woodcock and J. Davies. *Using Z: Specification, Refinement and Proof*. Prentice Hall, 1996.
20. 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.
21. 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 [12] and [5] 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 [12]. 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 z_i are labels (constants) then:

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

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

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

where the term t_i has type T_i.
The symbols $\preceq$, $\wedge$ and $\vee$ 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.* [7, 15, 19]). We can always, then, write the type of such operation schemas as $\mathbb{P}(T^{in} \vee T^{out'})$ 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 [12]). 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 7. *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 6. $t_0^{T_0} =_T t_1^{T_1} =_{df} t_0 \upharpoonright T = t_1 \upharpoonright T \qquad (T \preceq T_0 \text{ and } T \preceq T_1)$.

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 [19]. 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 [12]) 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 7. 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.*[8]

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

Definition 8 (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 a binding satisfying $t.\mathbf{x} = \perp$ for some observation $\mathbf{x}$, we have:

Lemma 3. $t(\perp) \in U \Rightarrow false$ □

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

Definition 9. $T_{\perp} =_{df} T \cup \{\perp^T\}$

[8] 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.

A.2 Preconditions

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

Definition 10. $Pre\ U\ x^{V} =_{df} \exists z \in U \bullet x =_{T^{in}} z \qquad (T^{in} \preceq V)$.

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

B Relational Completion

In this appendix, we review the chaotic relational completion operators discussed in [5] and [6]. Firstly, we define the non-strict-lifted totalisation in line with the intentions described in [19], chapter 16. We will write $T^{\star}$ for the set $T^{in}_{\perp} \star T^{out'}_{\perp}$.

Definition 11. $\overset{\bullet}{U} =_{df} \{z_0 \star z_1' \in T^{\star} \mid Pre\ U\ z_0 \Rightarrow z_0 \star z_1' \in U\}$

Then the following introduction and elimination rules are derivable:

Proposition 8.

$$\frac{t_0 \star t_1' \in T^{\star} \quad Pre\ U\ t_0 \vdash t_0 \star t_1' \in U}{t_0 \star t_1' \in \overset{\bullet}{U}}\ (\bullet^{+}) \qquad \frac{t_0 \star t_1' \in \overset{\bullet}{U} \quad Pre\ U\ t_0}{t_0 \star t_1' \in U}\ (\bullet_0^{-}) \qquad \frac{t_0 \star t_1' \in \overset{\bullet}{U}}{t_0 \star t_1' \in T^{\star}}\ (\bullet_1^{-})$$

□

Lemma 4. *The following extra rules are derivable for lifted-totalised sets:*

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

□

The strict-lifted totalisation is defined as follows:

Definition 12. $\overset{\ominus}{U} =_{df} \{z_0 \star z_1' \in T^{\star} \mid (Pre\ U\ z_0 \Rightarrow z_0 \star z_1' \in U) \wedge (z_0 = \perp \Rightarrow z_1' = \perp')\}$

We obtain obvious introduction and elimination rules, which in this case we will not state explicitly. In addition, we have fairly standard properties:

Lemma 5.

$$\frac{}{U \subseteq \overset{\ominus}{U}}\ (i) \qquad \frac{}{\overset{\ominus}{U} \subseteq \overset{\bullet}{U}}\ (ii) \qquad \frac{}{\perp \in \overset{\ominus}{U}}\ (iii)$$

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

Notice that in (v) t_0 ranges over the natural carrier set, rather than the extended carrier. □