Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 229 (2009) 27–47
www.elsevier.com/locate/entcs

Relational Semantics Revisited

Eike Best[1]

Parallel Systems, Department of Computing Science
Carl von Ossietzky Universität, D-26111 Oldenburg, Germany

Kerstin Strecker

Max-Planck Gymnasium, D-37073 Göttingen, Germany

Abstract

This paper describes the systematic use of fixpoints for the relational semantics of unboundedly nondeterministic sequential programs. The focus is on analysing the expressiveness of various semantic definitions and of powerdomains including or excluding an 'error' element $\perp$.

Keywords: Denotational semantics, Egli-Milner ordering, fixpoints, Hoare ordering, powerdomains, relational semantics, Smyth ordering.

1 Introduction

This paper describes a set of fixpoint schemes suitable for defining the relational semantics of sequential nondeterministic programs, while focussing on the *expressiveness* of such semantics. For the purpose of this paper, expressiveness of a semantics is defined as the set of programs that are discriminated by it. For example, traditional partial correctness semantics equates the two programs **skip** and **skip or abort** but distinguishes **skip or abort** from **abort**, while traditional total correctness semantics identifies **skip or abort** with **abort** but distinguishes between **skip or abort** and **skip**.

We consider general nondeterminstic programs, allowing unbounded nondeterminism as well as nonterminating behaviour. Usually, a special 'error' symbol called $\perp$ (the *bottom* symbol) is employed in order to describe the effect of failing programs such as **abort** or of nonterminating programs such as **do true** $\rightarrow$ **skip od**. We investigate the possibility of using denotational (in particular, fixpoint-based)

[1] Email: eike.best@informatik.uni-oldenburg.de

doi:10.1016/j.entcs.2009.07.072

methods while avoiding the $\perp$ symbol as an initial state. This is done for three semantics of different degrees of expressiveness: partial correctness semantics, which is as expressive as Hoare proof rules, total correctness semantics, which is as expressive as Dijkstra weakest preconditions, and full semantics, which is as expressive as both of them in combination. Parametrised fixpoint definitions are introduced in order to accommodate these semantic variants.

The paper is organised as follows. Section 3 contains direct definitions of the three semantics (full, partial correctness, and total correctness). Section 4 gives equivalent fixpoint semantics while using the $\perp$ symbol as little as possible and set theory as much as possible. This is compared with traditional powerdomain approaches described in section 5. The concluding section 6 contains some remarks and a short bibliographic overview. Results whose proofs are believed by the authors not to be readily accessible elsewhere (for instance the proof of proposition 4.3, referring to a Smyth-ordering-free denotational semantics of total correctness) are contained in an appendix.

2 Notation and syntax

Let S be any set. The powerset of S is denoted by 2^S, and the disjoint union of two sets is denoted by $\dot\cup$. Let $\rho \subseteq S\times S$ be any relation on S. The set $dom(\rho)$ is defined as $\{x{\in}S \mid \exists y{\in}S{:}\,(x,y){\in}\rho\}$. For $a{\in}S$, we write $a\rho = \{x{\in}S \mid (a,x){\in}\rho\}$ and $\rho a = \{x{\in}S \mid (x,a){\in}\rho\}$. The relation id_S is defined as $\{(a,a) \mid a{\in}S\}$. Let $\rho_1 \subseteq S\times S'$, $\rho_2 \subseteq S''\times S'''$ with $S'' \subseteq S'$ be any two relations. Relational composition is defined by $\rho_1{\circ}\rho_2 \subseteq S\times S'''$ and $(a,b){\in}\rho_1{\circ}\rho_2 \Leftrightarrow \exists x{\in}S''{:}\,(a,x){\in}\rho_1{\wedge}(x,b){\in}\rho_2$. Programs p are built from the following syntax (see e.g. [6,9]):

$$p \;::=\; \mathbf{skip} \mid \mathbf{abort} \mid x{:=}E \mid x{:=}? \mid p_1;p_2 \mid \mathbf{if}\; B_1{\to}p_1 \,[]\, B_2{\to}p_2\; \mathbf{fi} \mid \mathbf{do}\; B{\to}p\; \mathbf{od}.$$

Using this syntax, one may express nondeterminism (using the **if** ... **fi** construct), infinite nondeterminism (using $x{:=}?$, meaning that x is assigned an arbitrary natural number), and iteration (using **do** ... **od**). E and B are expressions and Boolean expressions, respectively. As a (non-essential) simplification, we assume that the evaluation of an expression in a state never leads to an error. Errors may be introduced explicitly by the **abort** command, implicitly by non-applicable **if** commands, or by nonterminating **do** commands. Program **if** $true{\to}p_1$ [] $true{\to}p_2$ **fi** is abbreviated as p_1 **or** p_2.

All variables are assumed to be typed, so that, as usual, a state space $Z(p)$ (the set of functions from variables to values) can be associated with every program p. To simplify formulae, we write Z instead of $Z(p)$ and m instead of $m(p)$ if the program p is known from the context. Also, $Z_\perp(p) = Z(p)\,\dot\cup\{\perp\}$ and $Z_\perp = Z\,\dot\cup\{\perp\}$. A state $s{\in}Z_\perp$ is called proper if $s{\neq}\perp$. We will use the following notation consistently:

p	to denote an arbitrary program
DO	to denote a loop **do** $B \to c$ **od** with entry condition B and body c.

Let E be an expression, x a program variable, and s' a state. Then $s'[x{\leftarrow}E]$ is defined as the same state as s', except that the value of variable x is the value of E when evaluated in state s'.

3 Direct definitions of relational semantics

Relational semantics (also called 'natural semantics' [19]) associates a relation between initial and final states to a program. Since if a program does not terminate properly, no proper final state is produced, an artefact is used in order to describe (potential) nontermination. This is achieved by the artificial 'error' state $\perp$. The production of such a state as a final state indicates prior, not properly terminating execution. Adding it to the set of initial states may be viewed (intuitively and operationally) as redundant, since error states do not give rise to further executions. Therefore, our investigations start by considering relations ρ of the form $\rho \subseteq Z \times Z_\perp$, where the first factor, Z, denotes the initial states (excluding $\perp$) and the second factor, $Z_\perp$, denotes the final states (including $\perp$).

3.1 Full relational semantics $m(p)$

A semantic relation $m(p) \subseteq Z \times Z_\perp$ is associated with every program p, such that $(s', s) \in m(p)$ expresses the following operational meaning: *When started in (the proper state) $s' \in Z$, program p may end in (proper) state s (if $s \in Z$), or may fail to terminate (if $s = \perp$).* Relation $m(p)$ is defined by induction on the syntax of p:

$$
\begin{aligned}
m(\mathbf{skip}) &= \{(s', s') \mid s' \in Z\} \\
m(\mathbf{abort}) &= \{(s', \perp) \mid s' \in Z\} \\
(s', s) \in m(x{:=}E) &\Leftrightarrow s = s'[x{\leftarrow}E] \text{ (section 2 explains notation)} \\
(s', s) \in m(x{:=}?) &\Leftrightarrow \exists i \in \mathbb{N}\colon s = s'[x{\leftarrow}i] \\
(s', s) \in m(p_1;p_2) &\Leftrightarrow ((s', s) \in m(p_1) \circ m(p_2)) \ \vee \ (s = \perp \wedge (s', \perp) \in m(p_1)) \\
(s', s) \in m(\mathbf{if}\ B_1 \to p_1 \,[]\, B_2 \to p_2\ \mathbf{fi}) &\Leftrightarrow \\
\exists j \in \{1, 2\}\colon B_j(s') = \mathit{true} \wedge (s', s) \in m(p_j) &\ \vee \ (s = \perp \wedge B_1(s') = B_2(s') = \mathit{false})\,.
\end{aligned}
$$

For the definition of $m(DO)$, with $DO = \mathbf{do}\, B \to c\, \mathbf{od}$, let a finite sequence $s_0, \ldots, s_r$ (an infinite sequence $s_0, s_1, \ldots$) of states in $Z_\perp$ be called a valid m-sequence (with respect to $\mathbf{do}\, B \to c\, \mathbf{od}$) if $B(s_j) \wedge (s_j, s_{j+1}) \in m(c)$ for $0 \leq j < r$ (respectively, for $0 \leq j$). Then,

$$
\begin{aligned}
&(s', s) \in m(\mathbf{do}\, B \to c\, \mathbf{od}) \\
\Leftrightarrow \quad &[s \neq \perp \wedge \exists \text{ valid } m\text{-sequence } s_0, \ldots, s_r\colon s' = s_0, s_r = s \text{ and } \neg B(s_r)] \\
\vee\ &[s = \perp \wedge \exists \text{ valid } m\text{-sequence } s_0, \ldots, s_r\colon s' = s_0 \text{ and } s_r = \perp] \\
\vee\ &[s = \perp \wedge \exists \text{ infinite valid } m\text{-sequence } s_0, s_1, \ldots\colon s' = s_0].
\end{aligned}
$$

The first term describes normal termination of the loop, the second term describes the case that the loop body leads to nontermination, and the third term describes infinite execution. Note that, as a consequence of the definition, all s_j, except possibly s_r, are proper.

For later use, the conjunction $((s',s)\in m(DO)\wedge\bot\notin s'm(DO))$ is rewritten as follows:

$$
\begin{aligned}
&((s',s)\in m(DO)\wedge\bot\notin s'm(DO))\\
\Leftrightarrow\quad &[\exists \text{ valid } m\text{-sequence } s_0,\ldots,s_r\colon s'{=}s_0, s_r{=}s \text{ and } \neg B(s_r)] && \text{(A)}\\
\wedge\ &[\neg\exists \text{ valid } m\text{-sequence } s_0,\ldots,s_r\colon s'{=}s_0, B(s_r) \text{ and } \bot\in s_r m(c)] && \text{(B)}\\
\wedge\ &[\neg\exists \text{ infinite valid } m\text{-sequence } s_0,s_1,\ldots\colon s'{=}s_0]. && \text{(C)}
\end{aligned}
$$

(A) rewrites $(s',s)\in m(DO)$ and (B)$\wedge$(C) rewrites $\bot\notin s'm(DO)$.

Examples: If there is only a single integer variable x, every state can be written as $(x{=}i)$, and we have, for instance:

$$
\begin{aligned}
&((x{=}{-}2),(x{=}3)) && \in m(x{:=}x{+}5)\\
&((x{=}{-}2),(x{=}3)) && \in m(x{:=}?)\\
&((x{=}{-}2),(x{=}{-}1)) && \notin m(x{:=}?)\\
&((x{=}{-}2),\bot) && \notin m(x{:=}?)\\
&((x{=}{-}2),(x{=}0)) && \in m(\mathbf{do}\ x{<}0{\to}(x{:=}x{+}1\ \mathbf{or}\ x{:=}{-}x)\ \mathbf{od})\\
&((x{=}{-}2),(x{=}1)) && \in m(\mathbf{do}\ x{<}0{\to}(x{:=}x{+}1\ \mathbf{or}\ x{:=}{-}x)\ \mathbf{od})\\
&((x{=}{-}2),\bot) && \notin m(\mathbf{do}\ x{<}0{\to}(x{:=}x{+}1\ \mathbf{or}\ x{:=}{-}x)\ \mathbf{od})\\
&((x{=}{-}2),\bot) && \in m(\mathbf{do}\ x{<}0{\to}(x{:=}x{+}1\ \mathbf{or\ skip})\ \mathbf{od})\\
& && \text{(since the loop may fail to terminate)}\\
&((x{=}{-}2),\bot) && \notin m(\mathbf{do}\ x{\neq}0{\to}\mathbf{if}\ x{>}0{\to}x{:=}x{-}1\ \square\ x{<}0{\to}x{:=}?\ \mathbf{fi}\ \mathbf{od}).
\end{aligned}
$$

3.2 Partial and total relational semantics $m_1(p)$ and $m_2(p)$

From $m(p)$, two relations $m_1(p), m_2(p) \subseteq Z(p)\times Z(p)$ are derived as follows:

$$
\begin{aligned}
m_1(p) &= m(p)\cap(Z(p)\times Z(p))\\
m_2(p) &= m(p)\cap(\,(Z(p)\backslash(m(p)\bot))\times Z(p)\,).
\end{aligned}
$$

The relation m_1 describes partial correctness semantics; it simply forgets about nontermination and thus equates, for instance, **skip** and (**skip or abort**). By contrast, m_2 describes total correctness semantics; it contains a pair $(s',s)\in m(p)\cap(Z\times Z)$ only if s' under no circumstances leads to nontermination, and thus it equates (**skip or abort**) and **abort**. It can be shown [6] – and is well-known – that $m_1(p)$ is as expressive as Hoare's proof rules [13], while $m_2(p)$ is as expressive as Dijkstra's

weakest precondition function $wp(p)$ [9], and that $m(c)$ can be reconstructed if both $m_1(p)$ and $m_2(p)$ are known. Thus, m_1 and m_2 are incomparable and individually less expressive than m, but both together are as expressive as m.

In the next two subsections we show, briefly, that instead of deriving them from $m(p)$, the two relations $m_1(p)$ and $m_2(p)$ can also be defined inductively.

3.3 Inductively defined partial correctness semantics $m'_1(p) = m_1(p)$

Clauses of $m'_1(p) \subseteq Z \times Z$ that differ from those of $m(p)$ are given below.

$$
\begin{aligned}
m'_1(\mathbf{skip}) &= id_Z \\
m'_1(\mathbf{abort}) &= \emptyset \\
m'_1(p_1;p_2) &= m'(p_1) \circ m'(p_2) \\
(s',s) \in m'_1(\mathbf{if}\ B_1 \to p_1 \,[]\, B_2 \to p_2\ \mathbf{fi}) &\Leftrightarrow \exists j \in \{1,2\}\colon B_j(s') = \mathit{true} \wedge (s',s) \in m'_1(p_j).
\end{aligned}
$$

Let $s_0, \ldots, s_r$ be called a valid m'_1-sequence if $B(s_j) \wedge (s_j, s_{j+1}) \in m'_1(c)$ for all $0 \leq j < r$.

$$
\begin{aligned}
(s',s) \in m'_1(\mathbf{do}\, B \to c\, \mathbf{od}) \Leftrightarrow\ & \exists \text{ valid } m'_1\text{-sequence } s_0, \ldots, s_r\colon \\
& s' = s_0,\ s_r = s \text{ and } B(s_r) = \mathit{false}.
\end{aligned}
$$

Lemma 3.1 *Let p be any program. Then $m'_1(p) = m_1(p)$.*

3.4 Inductively defined total correctness semantics $m'_2(p) = m_2(p)$

Clauses of $m'_2(p) \subseteq Z \times Z$ that differ from those of $m'_1(p)$ are given below.

$$
\begin{aligned}
& m'_2(\mathbf{abort}) = \{(s', \bot) \mid s' \in Z\} \\
& (s',s) \in m'_2(p_1;p_2) \Leftrightarrow ((s',s) \in m'_2(p_1) \circ m'_2(p_2)) \ \wedge\ (s' m'_2(p_1) \subseteq dom(m'_2(p_2))) \\
& (s',s) \in m'_2(\mathbf{if}\ B_1 \to p_1 \,[]\, B_2 \to p_2\ \mathbf{fi}) \ \Leftrightarrow \\
& \quad \exists j \in \{1,2\}\colon B_j(s') = \mathit{true} \wedge (s',s) \in m'_2(p_j) \ \wedge\ \forall j \in \{1,2\}\colon B_j(s') = \mathit{true} \Rightarrow s' m'_2(p_j) \neq \emptyset\,.
\end{aligned}
$$

Let $s_0, \ldots, s_r$ $(s_0, s_1, \ldots)$ be called a finite (resp. infinite) valid m'_2-sequence iff $B(s_j) \wedge (s_j, s_{j+1}) \in m_2(c)$ for all $0 \leq j < r$ (resp., for all $0 \leq j$). Then,

$$
\begin{aligned}
& (s',s) \in m'_2(\mathbf{do}\, B \to c\, \mathbf{od}) \\
\Leftrightarrow\ & [\exists \text{ valid } m'_2\text{-sequence } s_0, \ldots, s_r\colon s' = s_0, s_r = s \text{ and } \neg B(s_r)] && \text{(A2)} \\
& \wedge\ [\neg\exists \text{ valid } m'_2\text{-sequence } s_0, \ldots, s_r\colon s' = s_0, B(s_r) \text{ and } s_r m_2(c) = \emptyset] && \text{(B2)} \\
& \wedge\ [\neg\exists \text{ infinite valid } m'_2\text{-sequence } s_0, s_1, \ldots\colon s' = s_0]. && \text{(C2)}
\end{aligned}
$$

Note how the semantics of the sequential composition takes the intended meaning of m_2 into account: a pair (s', s) is included in $m'_2(p_1;p_2)$ only if for *every* state t with $(s', t) \in m'_2(p_1)$ there is some state s'' with $(t, s'') \in m'_2(p_2)$, i.e., p_2 terminates when started with t.

Lemma 3.2 *Let p be any program. Then $m'_2(p) = m_2(p)$.*

4 Fixpoint definitions of relational semantics

The objective of the present section is to provide alternative (denotational-style) definitions of a triple of relations $\widetilde{m}(p)$, $\widetilde{m_1}(p)$ and $\widetilde{m_2}(p)$, such that they equal $m(p)$, $m_1(p)$ and $m_2(p)$, respectively. To this end, we consider the following function, $\widetilde{\mathcal{N}}$, on relations, intended for describing a loop:

$$(1)\quad \widetilde{\mathcal{N}} : \begin{cases} (2^{Z\times Y}, \widetilde{\sqsubseteq}) \to (2^{Z\times Y}, \widetilde{\sqsubseteq}) \\ \qquad \rho \qquad \mapsto \widetilde{\mathcal{N}}(\rho) = (\ (\neg B\times Z) \cap id_Z)\ \cup\ ((B\times Z) \cap (\widetilde{A}(c) \boxtimes \rho)\ ). \end{cases}$$

Here, $\neg B$ is the set of states for which B evaluates to *false* and B is the set of states for which B evaluates to *true* (as said before, we assume that evaluating B does not lead to any error). The operational intuition behind the last line of (1) is that: *either B is false and the loop stops with unchanged initial state, or B is true and the body $\widetilde{A}(c)$ is executed, 'after' (i.e., $\boxtimes$) which the same situation is repeated.*

We consider $\widetilde{\mathcal{N}}$ to be parametric in terms of Y, $\widetilde{\sqsubseteq}$, $\widetilde{A}(c)$ and $\boxtimes$, with the following intentions. As for using $2^{Z\times Y}$, the set of initial states is always Z, excluding $\perp$, and the set Y of final states should be either Z (for $\widetilde{m_1}, \widetilde{m_2}$) or $Z_\perp$ (for $\widetilde{m}$); this provides for avoiding $\perp$ as much as possible. The ordering $\widetilde{\sqsubseteq}$ should be either $\subseteq$ or $\supseteq$, providing for maximal use of set theory. The relation $\widetilde{A}(c)$ is always given inductively, since c is the loop body. The relational composition $\boxtimes$ may need to vary in order to discriminate between $\widetilde{m}$, $\widetilde{m_1}$ and $\widetilde{m_2}$.

4.1 Fixpoint for partial correctness semantics $\widetilde{m_1}(p)$=$m_1(p)$

In this section, equation (1) is specialised as follows:

$$\begin{array}{rcll} Y & = & Z & \\ \widetilde{\sqsubseteq} & = & \subseteq & \\ \widetilde{A}(c) & = & \widetilde{m_1}(c) & \text{(known inductively)} \\ \boxtimes & = & \circ & \text{(ordinary relational composition).} \end{array}$$

Let $\widetilde{\mathcal{N}}$, with these definitions, be called $\widetilde{N1}$. Then $\widetilde{N1}$ is continuous (and hence monotonic) with respect to $\subseteq$. This follows from the continuity of $\cap$ and $\cup$ in both arguments, as well as the continuity of $\circ$ in its second (as well as its first) argument. Hence $\widetilde{N1}$ has a minimal fixpoint. Let $\widetilde{m_1}(DO)$ be defined as this least fixpoint. The other clauses of $\widetilde{m_1}$ follow the same definition as that of m_1'.

Proposition 4.1 *$\widetilde{m_1}(p) = m_1(p)$ for all programs p.*

4.2 Fixpoint for total correctness semantics $\widetilde{m_2}(p) = m_2(p)$

There is a suitable relational composition $\boxtimes$ such that the minimal fixpoint of $\widetilde{\mathcal{N}}$, so defined, is $\widetilde{m_2}(DO)$:

Definition 4.2 [Demonic composition, cf. [8]] Let σ and τ be two relations $\subseteq Z \times Z$. Then $\sigma * \tau$ is a new relation $\subseteq Z \times Z$ which is defined by $(a,b) \in (\sigma * \tau) \Leftrightarrow ((a,b) \in (\sigma \circ \tau) \wedge (a\sigma \subseteq dom(\tau)))$.

Note that $*$ exactly mimics the sequential definition for m_2' in section 3.4. It is associative and monotonic in the second argument (see appendix A). Equation (1) is specialised as follows:

$$\begin{array}{rcll} Y & = & Z & \\ \widetilde{\sqsubseteq} & = & \subseteq & \\ \widetilde{A}(c) & = & \widetilde{m_2}(c) & \text{(known inductively)} \\ \boxtimes & = & * & \text{(definition 4.2).} \end{array}$$

Let $\widetilde{N2}$ be defined as $\widetilde{\mathcal{N}}$ with this choice. By the above, $\widetilde{N2}$ is, in its turn, monotonic with respect to $\subseteq$, and has a unique minimal fixpoint. Let $\widetilde{m_2}(DO)$ be defined as this least fixpoint. The other clauses of $\widetilde{m_2}$ follow the same definition as that of m_2'.

We can now state the first main result of this paper. Its proof (appendix A) is surprisingly nontrivial:

Proposition 4.3 $\widetilde{m_2}(p) = m_2(p)$ *for all programs* p.

Examples: First, consider $p_1 =$ **var** $x : \{-2,-1,0,1,2\}$; **do** $x{<}0 \to (x{:=}x{+}1$ **or** $x{:=}{-}x)$ **od**. Moreover, consider

$$\begin{aligned} \gamma_1 = \{&((x{=}{-}2),(x{=}2)),((x{=}{-}2),(x{=}1)),((x{=}{-}2),(x{=}0)),((x{=}{-}1),(x{=}1)),\\ &((x{=}{-}1),(x{=}0)),((x{=}0),(x{=}0)),((x{=}1),(x{=}1)),((x{=}2),(x{=}2))\} \\ \delta_1 = \gamma_1. \end{aligned}$$

Then γ_1 is a fixpoint of $\widetilde{N1}$ (with respect to p_1). It is approximated as follows:

$$\begin{aligned} \gamma_1^0 &= \emptyset \qquad \text{(the empty relation)} \\ \gamma_1^1 &= \widetilde{N1}(\emptyset) = \{((x{=}0),(x{=}0)),((x{=}1),(x{=}1)),((x{=}2),(x{=}2))\} \\ \gamma_1^2 &= \widetilde{N1}(\gamma_1^1) = \gamma_1^1 \cup \{((x{=}{-}2),(x{=}2)),((x{=}{-}1),(x{=}1)),((x{=}{-}1),(x{=}0))\} \\ \gamma_1^3 &= \widetilde{N1}(\gamma_1^2) = \gamma_1. \end{aligned}$$

By contrast, for $\widetilde{N2}$, δ_1 (which is the same as γ_1) is approximated as follows:

$$\begin{aligned} \delta_1^0 &= \emptyset \\ \delta_1^1 &= \widetilde{N2}(\emptyset) = \{((x{=}0),(x{=}0)),((x{=}1),(x{=}1)),((x{=}2),(x{=}2))\} \\ \delta_1^2 &= \widetilde{N2}(\delta_1^1) = \delta_1^1 \cup \{(((x{=}{-}1),(x{=}1)),((x{=}{-}1),(x{=}0))\} \\ \delta_1^3 &= \widetilde{N2}(\delta_1^2) = \gamma_1. \end{aligned}$$

The difference is that in δ_1, the pair $(x{=}{-}2),(x{=}2))$ is added only when it is absolutely certain that initial state $(x{=}{-}2)$ does not lead to nontermination.

Consider next $p_2 = \mathbf{do}\ x{<}0 \to (x{:=}x{+}1\ \mathbf{or}\ \mathbf{skip})\ \mathbf{od}$ and

$$\gamma_2 = \delta_2 \cup \{((x{=}k),(x{=}0)) \mid k < 0\}$$
$$\text{with}\quad \delta_2 = \{((x{=}i),(x{=}i)) \mid i \geq 0\}.$$

Then γ_2 is (least) fixpoint of $\widetilde{N1}$ (with respect to p_2), and it happens also to be a (non-minimal) fixpoint of $\widetilde{N2}$. Moreover, δ_2 is (least) fixpoint of $\widetilde{N2}$, but no fixpoint of $\widetilde{N1}$. Thus, $m_1(p_2) = \gamma_2$ and $m_2(p_2) = \delta_2$. Note how the pair $((x{=}{-}1),(x{=}0))$ is added to $m_1(p_2)$ by the definition of $\circ$ and by the facts that $((x{=}{-}1),(x{=}0))$ is one of the possible executions of the loop's body and $((x{=}0),(x{=}0))$ is in $m_1(p_2)$ because the entry condition evaluates to *false* in state $x{=}0$. The definition of $*$, by contrast, prevents $((x{=}{-}1),(x{=}0))$ from being included in the least fixpoint of $\widetilde{N2}$.

Consider $p_3 = \mathbf{do}\ x{\neq}0 \to \mathbf{if}\ x{>}0 \to x{:=}x{-}1\ \square\ x{<}0 \to x{:=}?\ \mathbf{fi}\ \mathbf{od}$ and

$$\delta_3 = \{((x{=}i),(x{=}0)) \mid i \in \mathbb{Z}\}.$$

Then δ_3 is both a (minimal) fixpoint of $\widetilde{N1}$ and a (minimal) fixpoint of $\widetilde{N2}$ with respect to p_3.

4.3 Fixpoint for full relational semantics $\widetilde{m}(p) = m(p)$

Since $m(p)$ is a subset of $Z \times Z_\perp$, we are led to consider $Y = Z_\perp$, i.e. the set of relations $2^{Z \times Z_\perp}$, and a suitable partial order as well as a suitable relational composition for $\boxtimes$ on it. Keeping in mind that $\boxtimes = \circ$ in section 4.1 and $\boxtimes = *$ in section 4.2 simulate, respectively, the sequential composition in the definitions of m_1' (section 3.3) and m_2' (section 3.4), the following is a candidate, since it captures the sequential composition as defined in section 3.1.

Definition 4.4 [Erratic composition] Let σ and τ be two relations $\subseteq Z \times Z_\perp$. Then $\sigma \diamond \tau$ is a new relation $\subseteq Z \times Z_\perp$ defined by $(a,b) \in (\sigma \diamond \tau) \Leftrightarrow ((a,b) \in (\sigma \circ \tau) \vee ((a,\perp) \in \sigma \wedge b{=}\perp)$.

The example $p_4 = \mathbf{var}\ x : \{0\};\ \mathbf{do}\ x{=}0 \to \mathbf{skip}\ \mathbf{od}$ shows that neither $\subseteq$ nor $\supseteq$ can be chosen as $\widetilde{\sqsubseteq}$. This is because the m semantics of p_4 is $m(p_4) = \{(x{=}0,\perp)\}$, but the least fixpoints of $\widetilde{\mathcal{N}}$ under $\subseteq$ and under $\supseteq$ are, respectively, $\emptyset$ and $\{(x{=}0,x{=}0),(x{=}0,\perp)\}$. The same example shows that switching from least to greatest fixpoints does not help, either.

Instead, the set $2^{Z_\perp}$ will be provided with the following ordering. For $X, Y \in 2^{Z_\perp}$:

$$X \sqsubseteq_{full} Y \quad \Leftrightarrow \quad (X \backslash \{\perp\} \subseteq Y) \wedge (\perp \in X \vee \perp \notin Y). \tag{2}$$

Lemma 4.5 $(2^{Z_\perp}, \sqsubseteq_{full})$ *is a complete lattice.*

$(2^{Z_\perp}, \sqsubseteq_{full})$ is actually isomorphic to the lattice product $(2^Z, \subseteq) \otimes (2^{\{\perp\}}, \supseteq)$ with its induced partial ordering. Taking $\subseteq$ in the first lattice and $\supseteq$ in the second can be interpreted as follows. The $\subseteq$ indicates 'less definite information', that is: if X and Y are two sets of (final) proper states and if $X \subseteq Y$, then we know less of the possible results if we know X than if we know Y. Similarly, the fact that $\{\perp\}$ lies 'below'

$\emptyset$ can be interpreted as saying that the former represents less definite information than the latter, that is: if we know that some computation potentially does not terminate, we know less of the possible results than if we know that nontermination is not possible.

The partial order $\sqsubseteq_{full}$ can be lifted to relations $\rho \subseteq Z \times Z_\perp$ by putting

$$\rho_1 \sqsubseteq_{full} \rho_2 \quad \Leftrightarrow \quad \forall s \in Z\colon s\rho_1 \sqsubseteq_{full} s\rho_2.$$

Then $(2^{Z\times Z_\perp}, \sqsubseteq_{full})$ is another complete lattice. In particular, $\sqsubseteq_{full}$ is reflexive, transitive, and antisymmetric. It is also a congruence with respect to $\cap$ and $\cup$, and moreover, $\diamond$ is monotonic with respect to $\sqsubseteq_{full}$ in its second argument. Equation (1) is now specialised as follows:

$$\begin{array}{rcll} Y & = & Z_\perp & \\ \widetilde{\sqsubseteq} & = & \sqsubseteq_{full} & \\ \widetilde{A}(c) & = & \widetilde{m}(c) & \text{(known inductively)} \\ \boxtimes & = & \diamond & \text{(definition 4.4).} \end{array}$$

Let $\widetilde{N}$ be defined as $\widetilde{\mathcal{N}}$ with this choice. $\widetilde{N}$ is monotonic with respect to $\subseteq$ and has a unique minimal fixpoint. Let $\widetilde{m}(p)$ be the least fixpoint of $\widetilde{N}$ for $p = DO$, and $m(p)$ for all other programs p. Then we get the second main result of this paper:

Proposition 4.6 *$\widetilde{m}(p) = m(p)$ for all programs p.*

5 Powerdomain definitions of relational semantics

The purpose of this section is to recall Egli-Milner, Hoare and Smyth constructions for, respectively, $m(p)$, $m_1(p)$ and $m_2(p)$. We focus on the loop

$$DO \quad = \quad \textbf{do}\ B \to c\ \textbf{od},$$

because this is where fixpoints come into play. Our aim is to define $\widehat{m}(DO)$, $\widehat{m_1}(DO)$ and $\widehat{m_2}(DO)$, such that they equal $m(DO)$, $m_1(DO)$ and $m_2(DO)$, respectively. We follow standard powerdomain theory in the sense that we now include $\perp$ in the set of *initial* states, that is, we consider uniformly the set ('flat domain') $Z_\perp$ as the ground set for both initial and final states. The constructions in this section differ (as usual in powerdomain theory) in terms of which subsets of this ground set (and which order on them) are taken as the basis for fixpoint constructions. The proofs of the claims in this section are well-known, or are easily derivable from well-known ones, and are therefore omitted.

Often, the standard denotational semantics of DO employs a functional fixpoint scheme [3,18], that is, a function mapping (partial) functions to functions. We reformulate this scheme in terms of relations, because that is what will make the considerations uniform with those of section 4. Moreover, in the standard theory of powerdomains, the relational composition that has been a parameter in the previous sections (variously called $\boxtimes$, $\circ$, $*$, and $\diamond$) is uniformly the relational composition $\circ$.

Thus, we consider the following function $\widehat{\mathcal{N}}$, where D, $\widehat{\sqsubseteq}$ and $\widehat{A}(c)$ are parametric:

$$(3)\quad \widehat{\mathcal{N}} : \begin{cases} (D, \widehat{\sqsubseteq}) \to (D, \widehat{\sqsubseteq}) \\ \quad \rho \quad \mapsto (\ (\neg B \times Z) \cap id_Z)\ \cup\ (B_\perp \times \{\perp\})\ \cup\ ((B \times Z_\perp) \cap (\widehat{A}(c) \circ \rho)\ )\ . \end{cases}$$

Here, $\neg B$ is the set of states for which B evaluates to *false*, $B_\perp$ is the set of states where B evaluates to $\perp$ (which, by our assumption that evaluating B in a proper initial state does not lead to any error, equals $\{\perp\}$), and B is the set of states for which B evaluates to *true*.

5.1 Full relational semantics $M(p)$ and the Egli-Milner ordering

Commonly, the semantics of DO is given as a function from $Z_\perp$ to $P_0(Z_\perp)$, where $P_0(Z_\perp)$ is the set of all nonempty subsets of $Z_\perp$ which are finite or contain $\perp$ [16]. On $P_0(Z_\perp)$, the Egli-Milner ordering, is defined as follows [1], for $X, Y \in P_0(Z_\perp)$:

$$(4)\qquad X \sqsubseteq_{EM} Y \quad \Leftrightarrow \quad (\perp \notin X \wedge X = Y)\ \vee\ (\perp \in X \wedge (X \setminus \{\perp\}) \subseteq Y).$$

This yields a complete partial order $(P_0(Z_\perp), \sqsubseteq_{EM})$ with least element $\{\perp\}$.

For unbounded nondeterminism a generalisation along the lines of [12] may be incorporated. Let $P(Z_\perp)$ denote the set of all nonempty subsets of $Z_\perp$ (without the restriction of being finite or containing $\perp$). The ordering $\sqsubseteq_{EM} \subseteq P(Z_\perp) \times P(Z_\perp)$ can be defined on this extended set by the same formula, (4). Let $P(Z_\perp \times Z_\perp)$ denote the set of all relations $\rho \in 2^{Z_\perp \times Z_\perp}$ such that for all $s \in Z_\perp$, $s\rho \in P(Z_\perp)$. Clearly, this set is closed under relational composition. Moreover, $\sqsubseteq_{EM}$ may be extended as follows to $P(Z_\perp \times Z_\perp)$:

$$\rho_1 \sqsubseteq_{EM} \rho_2 \quad \Leftrightarrow \quad \forall s \in Z_\perp : s\rho_1 \sqsubseteq_{EM} s\rho_2.$$

This is the relational analogue of extending $\sqsubseteq_{EM}$ to functions from $Z_\perp$ to $P(Z_\perp)$, as in [3,18]. Let (3) be specialised by specifying D, $\widehat{\sqsubseteq}$ and $\widehat{A}(c)$, as follows:

$$\begin{aligned} D &= P(Z_\perp \times Z_\perp) \\ \widehat{\sqsubseteq} &= \sqsubseteq_{EM} \\ \widehat{A}(c) &= M(c) \text{ (known by induction).} \end{aligned}$$

Let $\widehat{N}$ denote $\widehat{\mathcal{N}}$, with this choice. Then $\widehat{N}$ is monotonic with respect to $\sqsubseteq_{EM}$. This follows from the monotonicity of $\cap$ and $\cup$ in both arguments, as well as the monotonicity of $\circ$ in its second (as well as its first) argument with respect to $\sqsubseteq_{EM}$. Hence $\widehat{N}$ has a unique minimal fixpoint. Define $M(DO)$ as this smallest fixpoint. Since $M(DO) \subseteq Z_\perp \times Z_\perp$ is not of the same type as $m(DO) \subseteq Z \times Z_\perp$, let $\widehat{m}(DO) = M(DO) \cap (Z \times Z_\perp)$. When p is not a loop, $\widehat{m}(p)$ follows the same definition as for $m(p)$.

Lemma 5.1 *Let p be any program. Then $m(p) = \widehat{m}(p)$.*

Note that $M(DO)$ and $\widehat{m}(DO)$ carry *exactly* the same information: above, $\widehat{m}(DO)$ has been derived from $M(DO)$; conversely, when $\widehat{m}(DO) \subseteq Z \times Z_\perp$ is given,

$M(DO)$ can be derived as $M(DO) = \widehat{m}(DO) \cup \{(\bot, \bot)\}$. In this sense, adding $\bot$ to the set of initial states can be viewed (mathematically, cf. also section 3) as redundant.

There is a relationship between the orderings $\sqsubseteq_{full}$ in (2) and $\sqsubseteq_{EM}$ in (4). On the subset $P(Z_\bot)$ of $2^{Z_\bot}$, $\sqsubseteq_{full}$ generalises $\sqsubseteq_{EM}$ in the sense of satisfying $\sqsubseteq_{EM} \subseteq \sqsubseteq_{full}$.

5.2 Partial correctness semantics $M_1(p)$ and the Hoare ordering

Let $Q(Z_\bot)$ be defined as 2^Z (without $\bot$) and $\sqsubseteq_H \subseteq Q(Z_\bot) \times Q(Z_\bot)$ as ordinary inclusion $\subseteq$ on 2^Z. Let $Q(Z_\bot \times Z_\bot)$ denote the set of relations $\rho \in 2^{Z_\bot \times Z_\bot}$ such that for all $s \in Z_\bot$, $s\rho \in Q(Z_\bot)$. $\sqsubseteq_H$ may be extended to $Q(Z_\bot \times Z_\bot)$ by

$$\rho_1 \sqsubseteq_H \rho_2 \quad \Leftrightarrow \quad \forall s \in Z_\bot : s\rho_1 \sqsubseteq_H s\rho_2.$$

Let (3) be specialised as follows:

$$\begin{aligned} D &= Q(Z_\bot \times Z_\bot) \\ \widehat{\sqsubseteq} &= \sqsubseteq_H \\ \widehat{A}(c) &= M_1(c) \text{ (known by induction).} \end{aligned}$$

Let $\widehat{N1}$ be $\widehat{\mathcal{N}}$, with this choice. Then $\widehat{N1}$ is clearly continuous (and hence monotonic) with respect to $\subseteq$. Let $M_1(DO)$ be the smallest fixpoint of $\widehat{N1}$ with these parameters. Since $M_1(DO) \subseteq Z_\bot \times Z_\bot$ is not of the same type as $m_1(DO) \subseteq Z \times Z$, let $\widehat{m_1}(DO) = M_1(DO) \cap (Z \times Z)$. The other clauses of $\widehat{m_1}(p)$ follow the same definition as for $m_1'(p)$.

Lemma 5.2 *Let p be any program. Then $m_1(p) = \widehat{m_1}(p)$.*

Again, $M_1(DO)$ and $\widehat{m_1}(DO)$ carry exactly the same information (as it turns out, they are actually equal).

5.3 Total correctness semantics $M_2(p)$ and the Smyth ordering

Finally, let $R(Z_\bot)$ be defined as $(2^Z \setminus \{\emptyset\}) \cup \{Z_\bot\}$ and $\sqsubseteq_S \subseteq R(Z_\bot) \times R(Z_\bot)$ as reverse inclusion $\supseteq$ on $R(Z_\bot)$. Let $R(Z_\bot \times Z_\bot)$ denote the set of all relations $\rho \in 2^{Z_\bot \times Z_\bot}$ such that for all $s \in Z_\bot$, $s\rho \in R(Z_\bot)$. Again, this set is closed under relational composition. $\sqsubseteq_S$ may be extended to $R(Z_\bot \times Z_\bot)$ by

$$\rho_1 \sqsubseteq_S \rho_2 \quad \Leftrightarrow \quad \forall s \in Z_\bot : s\rho_1 \sqsubseteq_S s\rho_2.$$

Let equation (3) be specialised as follows:

$$\begin{aligned} D &= R(Z_\bot \times Z_\bot) \\ \widehat{\sqsubseteq} &= \sqsubseteq_S \\ \widehat{A}(c) &= M_2(c) \text{ (known by induction).} \end{aligned}$$

Let $\widehat{N2}$ be defined as $\widehat{\mathcal{N}}$, with these definitions. Then $\widehat{N2}$ is monotonic with respect to $\sqsubseteq_S$. This follows from the monotonicity of $\cap$ and $\cup$ in both arguments, as well as the monotonicity of $\circ$ in its second (as well as its first) argument w.r.t. $\sqsubseteq_S$. We

define $M_2(DO)$ as the smallest fixpoint of $\widehat{N2}$ with these parameters. Once more, $M_2(DO) \subseteq Z_\perp \times Z_\perp$ is not of the same type as $m_2(DO) \subseteq Z \times Z$. Therefore, let $\widehat{m_2} \subseteq Z \times Z$ be defined as follows: for $s' \in Z$,

$$s'\widehat{m_2}(DO) \quad = \quad \begin{cases} \emptyset & \text{if } s'M_2 = Z_\perp \\ s'M_2 & \text{if } s'M_2 \neq Z_\perp. \end{cases}$$

The other clauses of $\widehat{m_2}(p)$ follow the same definition as for $m_2'(p)$.

Lemma 5.3 *Let p be any program. Then $m_2(p) = \widehat{m_2}(p)$.*

Again, $M_2(DO)$ and $\widehat{m_2}(DO)$ carry exactly the same information: above, $\widehat{m_2}(DO)$ has been derived from $M_2(DO)$; conversely, when $\widehat{m_2}(DO) \subseteq Z \times Z_\perp$ is given,

$$\begin{aligned} M_2(DO) = \quad & \{(s', s) \in \widehat{m_2}(DO) \mid s'\widehat{m_2}(DO) \neq \emptyset\} \\ & \cup \{(s', s) \in Z_\perp \times Z_\perp \mid s'\widehat{m_2}(DO) = \emptyset\}. \end{aligned}$$

In all three cases, the full state space $Z_\perp$ has been used both for initial and for final states. The 'leaner' relations, which do not allow $\perp$ to be an initial state, have been derived *a posteriori*. The derivation has been particularly noticeable in the case of $\widehat{m_2}$, where the 'chaotic' interpretation of failure (yielding $Z_\perp$ as set of final states), as described by M_2, was trimmed down to the 'operational' interpretation of failure, as described by m_2 (yielding $\emptyset$ as set of final states). When sections 4.2 and 5.3 are compared, it appears that denotational semantics can be given both for the chaotic interpretation of failure induced by the Smyth ordering and for the non-chaotic interpretation of failure induced by the notion of demonic composition. The Smyth fixpoint described above uses a least fixpoint on reverse subset ordering (or, equivalently, a maximal fixpoint on ordinary ordering) while the construction described in section 4.2 uses a least fixpoint on ordinary ordering, and yet they are equally expressive, because the distinction in terms of their orderings is neutralised by the distinction in terms of their relational composition operations.

6 Conclusion

In sections 4.1 and 4.2, it was shown how the relational correspondents of Hoare's (original) proof rules for partial correctness semantics and Dijkstra's *wp* function for total correctness semantics can be defined denotationally without needing to introduce the $\perp$ state or using powerset constructions other than provided by set theory. The former should hardly be surprising, since the traditional Hoare ordering is the same as subset ordering (cf. section 5.2). The latter (and the concomitant result, proposition 4.3) came as slightly more of a surprise, since total correctness semantics is frequently thought to be intimately linked to the Smyth ordering, which does use the $\perp$ state (section 5.3). For full relational semantics (section 4.3 and proposition 4.6), we have used a double generalisation of the original Egli-Milner ordering. A first generalisation (as also discussed in [12]) allows for infinite state sets not containing the $\perp$ element and thus encompasses unbounded nondeterminism. A second generalisation uses a product of two lattices.

Since $\sqsubseteq_{full}$ is a mixture of subset and superset orderings, reduction to pure set theory has not completely been achieved for full semantics. However, consider replacing the $\perp$ symbol by $\top$, indicating 'certain termination' and yielding a semantic relation $m_{top}(p) \subseteq Z \times Z_\top$ for programs p, with $Z_\top = (Z \dot\cup \{\top\})$, such that, for instance:

$$\begin{aligned} m_{top}(\textbf{skip}) &= \{(s', s') \mid s' \in Z\} \cup \{(s', \top) \mid s' \in Z\} \\ m_{top}(\textbf{skip or abort}) &= \{(s', s') \mid s' \in Z\} \\ m_{top}(\textbf{abort}) &= \emptyset. \end{aligned}$$

Going through the motions of section 4.3 and all previous sections with $\top$ rather than $\perp$, will quite likely lead to a lattice $(2^{Z_\top}, \sqsubseteq'_{full})$ which is isomorphic to the lattice product $(2^Z, \subseteq) \otimes (2^{\{\top\}}, \subseteq)$ rather than $(2^Z, \subseteq) \otimes (2^{\{\perp\}}, \supseteq)$, and thus also to the lattice $(2^{Z_\top}, \subseteq)$, yielding, by $\sqsubseteq'_{full} = \subseteq$, a more complete reduction to set theory.

While Hesselink has shown in [12] that the Egli-Milner ordering can be generalised in order to encompass infinite nondeterminism, it is not discussed there whether or not $\perp$ can be avoided as an initial state. Discussions about eschewing $\perp$ are contained in [4] (pages 128 ff.) and in [10]. In [10], Doornbos has argued that $\perp$ can be avoided in special cases, but no operational or other general formal justifications have been given.

The demonic composition $*$ has been defined previously: e.g., implicitly in [5] and explicitly in [15]. In [5,11], moreover, 'functionals' such as $\widetilde{N1}$ and $\widetilde{N2}$ have been used. In [14], a pairing construction is used to capture total/partial correctness semantics.

In [8], we find a relational definition of the loop employing the $*$ operator, which, we believe, is not the best choice. Instead of set theoretic ones, this definition uses various demonic versions of set-theoretic operators in order to construct a domain in which a *maximal* fixpoint can be used. This may lead to complications. For instance, the very involved calculation of the maximal fixpoint of the example given on page 175 of [8] may be compared with the function $\widetilde{N2}$, coming from the present paper (which is actually continuous, although we have neither proved nor used this fact in this paper). When applied to the same example, $\widetilde{N2}$ yields stability of approximation after only two steps from the empty relation as starting point, using a much easier calculation. Nevertheless, it might be possible and interesting to conduct a study similar to the one in the present paper, using the union and intersection operators as parameters.

The semantics given in this paper suggests an approximation relation between nondeterministic programs, which in terms of (partial or total) correctness formulae, is antimonotonic. That is, if c_1 approximates c_2, then the set of valid (partial or total) correctness formulae pertaining to c_2 is a subset of those pertaining to c_1. This holds in all cases, as opposed to the usual Smyth ordering, where the set of valid total correctness formulae pertaining to c_1 is a subset of those pertaining to c_2, provided c_1 approximates c_2 (i.e., the logical characterisation is monotonic).

In future work, we propose to investigate the following two questions. First,

the generalised Egli-Milner ordering (2) of section 4.3 may appear to be rather underived. We would like to actually derive it, that is, show that there are no good alternatives. Second, we would like to strengthen the above remark about partial or total correctness formulae by showing – if possible – that the set of such formulae actually characterise the corresponding relational semantics.

Acknowledgement

This paper has benefitted from discussions with Jaco W. de Bakker, Rudolf Berghammer, Wim H. Hesselink and Burghard von Karger, to whom the observation that (2) corresponds to a lattice product is due. We would also like to thank the anonymous reviewers and non-anonymous participants of SOS'2008, in particular Bartek Klin, Matthew Hennessy and Peter Mosses, for comments.

References

[1] J.W. de Bakker: Recursive Programs as Predicate Transformers. Proc. Working Conference on Formal Description of Programming Concepts, IFIP, North Holland, Erich Neuhold (ed.), 15 pages (1977).

[2] J.W. de Bakker: *Mathematical Theory of Program Correctness*. Prentice Hall (1980).

[3] R. Berghammer: *Semantik von Programmiersprachen*. Lecture Notes, Universität Kiel (1996/97).

[4] R. Berghammer, B. v. Karger: Relational Semantics of Functional Programs. Chapter 8, pages 115–130, of [7].

[5] R. Berghammer, H. Zierer: Relational Algebraic Semantics of Deterministic and Nondeterministic Programs. *Theoretical Computer Science* 43, 123–147 (1986).

[6] E. Best: *Semantics of Sequential and Parallel Programs*. Prentice Hall (1996).

[7] C. Brink, W. Kahl, G. Schmidt (eds): *Relational Methods in Computer Science*. Advances in Computing Science, Springer-Verlag (Wien, New York) (1996).

[8] J. Desharnais, A. Mili, T.T. Nguyen: Refinement and Demonic Semantics. Chapter 11, pages 166–183, of [7].

[9] E.W. Dijkstra: *A Discipline of Programming*. Prentice Hall (1976).

[10] H. Doornbos: A Relational Model of Programs Without the Restriction to Egli-Milner Monotone Constructs. Proc. *ProCoMet'94* (E.R. Olderog, ed.). IFIP Transactions A-56, 363–382, North Holland (1994).

[11] Th.F. Gritzner, R. Berghammer: A Relation Algebraic Model of Robust Correctness. *Theoretical Computer Science* 159, 245–270 (1996).

[12] W.H. Hesselink: Interpretations of Recursion under Unbounded Nondeterminacy. *Theoretical Computer Science* 59, 211—234 (1988).

[13] C.A.R. Hoare: An Axiomatic Basis for Computer Programming. *Communications of the ACM* 12, 576–580 (1969).

[14] R. Maddux: Relation-algebraic Semantics. *Theoretical Computer Science* 160, 1—85 (1996).

[15] T.T. Nguyen: A Relational Model of Demonic Nondeterministic Programs. *Intern. J. Found. of Comp. Sci.* 2, 101–131 (1991).

[16] G.D. Plotkin: A Powerdomain Construction. *SIAM J. Comp.* 5, 452–486 (1967).

[17] K.M. Richter: Denotationale Formulierungen für relationale Semantiken. *Diplomarbeit*, Universität Hildesheim (1998).

[18] D. Schmidt: *Denotational Semantics: a Methodology for Language Development*. Allyn and Bacon (1986).

[19] G. Winskel: *The Formal Semantics of Programming Languages*. Foundations of Computing. The MIT Press, Cambridge, Massachusetts (1993).

A Some proofs

Proof of lemma 3.1: This follows directly by comparing the clauses in the definition of m_1' with the corresponding clauses in the definition of m, keeping in mind that $m_1 = m \cap (Z \times Z)$. □

Proof of lemma 3.2: This follows directly, except for the loop, for which we may use the following six implications, the first three of which are true because every valid m_2-sequence is also a valid m-sequence: (A2) implies (A); (B) implies (B2); (C) implies (C2); (A)∧(B) implies (A2); (B2) implies (B); and finally, (B2)∧(C2) implies (C). (Note that the r in (B) is the $r-1$ in (B2).) □

Proof of Proposition 4.1: Along the same lines as the proof of proposition 4.3, cf. [17]. □

Proof that $*$ is associative, i.e., satisfies $(\rho * \tau) * \sigma = \rho * (\tau * \sigma)$: By definition 4.2, $(a,b) \in (\rho * \tau) * \sigma$ can be rewritten as follows:

$$[\ \underbrace{\exists d}_{(11)}\colon [\underbrace{(\exists c\colon (a,c)\in\rho \wedge (c,d)\in\tau)}_{(111)} \wedge \underbrace{(\forall c'\colon (a,c')\in\rho \Rightarrow \exists d'\colon (c',d')\in\tau)}_{(112)}] \wedge \underbrace{(d,b)\in\sigma}_{(113)}\ ]$$

$$\wedge\ [\ \underbrace{\forall d''}_{(12)}\colon [\underbrace{(\exists c''\colon (a,c'')\in\rho \wedge (c'',d'')\in\tau)}_{(121)} \wedge \underbrace{(\forall c'''\colon (a,c''')\in\rho \Rightarrow \exists d'''\colon (c''',d''')\in\tau)}_{(122)}]$$

$$\Rightarrow\ \underbrace{\exists b'\colon (d'',b')\in\sigma}_{(123)}\ ].$$

By the same definition, $(a,b) \in \rho * (\tau * \sigma)$ becomes:

$$[\ \underbrace{\exists c}_{(21)}\colon [\underbrace{(a,c)\in\rho}_{(211)} \wedge \underbrace{(\exists d\colon (c,d)\in\tau \wedge (d,b)\in\sigma)}_{(212)} \wedge \underbrace{(\forall d'\colon (c,d')\in\tau \Rightarrow \exists b''\colon (d',b'')\in\sigma)}_{(213)}]\ ]$$

$$\wedge\ [\ \underbrace{\forall c'}_{(22)}\colon \underbrace{(a,c')\in\rho}_{(221)}\ \Rightarrow$$

$$(\ \exists b'\colon \underbrace{[\exists d''\colon (c',d'')\in\tau \wedge (d'',b')\in\sigma]}_{(222)} \wedge [\underbrace{\forall d'''}_{(223)}\colon \underbrace{(c',d''')\in\tau}_{(2231)} \Rightarrow \exists\, \underbrace{b'''\colon (d''',b''')\in\sigma}_{(2232)}]\ )\ ].$$

We prove that the first formula implies the second. To verify (21), take the c that exists by (111). It satisfies (211) by the first part of (111). It satisfies (212) by (111) and (113), taking the d that exists by (11). It also satisfies (213). To see this, consider any d' with $(c,d')\in\tau$. Then (121) is satisfies with d' for d'' and c for c'';

also, (122) is satisfied because of (112). The conclusion (123) yields the conclusion of (213). To verify (22), consider any c' with $(a,c')\in\rho$. From (112), there is a $\overline{d}$ with $(c',\overline{d})\in\tau$. From (12), with $\overline{d}$ for d'', there is a $\overline{b}$ with $(\overline{d},\overline{b})\in\sigma$. To satisfy (22), take $\overline{b}$ for b'. Then (222) is satisfied by $\overline{d}$ for d'', and (223) is satisfied because if d''' satisfies (2231), the existence of a b''' satisfying (2232) follows from (123).

Next, we prove the converse, viz. that the second formula implies the first. To verify (11), we may take c and d as in (21) and (212). This satisfies (111), (112) (using (221)) and (113) (using 212). To verify (12), consider any d'' with (121) and (122). The c'' that exists by (121) satisfies (221) (by c'' for c'), and hence there is a b' with ((222) and) (223). The chosen d'' satisfies (2231), hence (2232), which implies (123). □

Proof that $*$ is monotonic in the second argument: If $\tau_1\subseteq\tau_2$ then $\sigma\circ\tau_1\subseteq\sigma\circ\tau_2$ and $dom(\tau_1)\subseteq dom(\tau_2)$. The claim follows. □

Proof of proposition 4.3: By induction on program p. We restrict ourselves to the loop $DO = \mathbf{do}\, B \to c\, \mathbf{od}$, because that is the most involved case. On the one hand, we have the operational semantics $m(DO)$, from which we get $m_2(DO)$, and on the other hand, we have the fixpoint semantics $\widetilde{m_2}(DO)$. Since $((s',s)\in m(DO)\wedge\bot\notin s'm(DO)) \Leftrightarrow (s',s)\in m_2(DO)$, we may use, for the former, lines (A)–(C) in the definition of $m(DO)$, cf. section 3.1, and for the latter, lines (A2)–(C2) in the definition of $m_2'(DO)$, cf. section 3.4.

Let the notion of a valid $\widetilde{m_2}$-sequence (in what follows abbreviated by $\widetilde{m_2}$vseq) be defined as that of a valid m_2-sequence, except that m_2 is replaced by $\widetilde{m_2}$. Moreover, let the set of states which start an infinite $\widetilde{m_2}$-execution be defined as follows:

$$INF(DO) \;=\; \{s'\in Z \mid \exists \text{ infinite } \widetilde{m_2}\text{vseq } s_0, s_1, \ldots : s'=s_0\}.$$

For the proof of $\widetilde{m_2}(DO) = m_2(DO)$, we may assume, as an inductive step, that $\widetilde{m_2}(c) = m_2(c)$. Therefore, the notions of an $\widetilde{m_2}$-sequence and an m_2-sequence coincide, and $INF(DO)$ is the same set, independently of whether $\widetilde{m_2}$-sequences or m_2-sequences are considered. □

Proof of $\widetilde{m_2}(DO)\subseteq m_2(DO)$: Assume that $(s',s)\in\widetilde{m_2}(DO)$. We prove $(s',s)\in m_2(DO)$ by verifying (C2), (A2) and (B2), in that order, and using Lemma 3.2($\Leftarrow$).
(C2): $s'\notin INF(DO)$.
Proof: By contradiction. We assume that $s'\in INF(DO)$ and define $\widetilde{m_2}'(DO) = \widetilde{m_2}(DO)\backslash(INF(DO)\times Z)$. We show that $\widetilde{m_2}'(DO)$ (which is strictly contained in $\widetilde{m_2}(DO)$ by $(s',s)\in(\widetilde{m_2}(DO)\cap(INF(DO)\times Z))$) is also a fixpoint of $\widetilde{N2}$, contradicting the fact that $\widetilde{m_2}(DO)$ was defined as the smallest one. To this end, we show the two directions of $\widetilde{m_2}'(DO) = \widetilde{N2}(\widetilde{m_2}'(DO))$ separately:
($\subseteq$:) Assume $(t',t)\in\widetilde{m_2}'(DO)$. We intend to show $(t',t)\in\widetilde{N2}(\widetilde{m_2}'(DO))$. By definition of $\widetilde{m_2}'(DO)$, $(t',t)\in\widetilde{m_2}(DO)\wedge t'\notin INF(DO)$. Because $\widetilde{m_2}(DO)$ is a fixpoint, we also have $(t',t)\in\widetilde{N2}(\widetilde{m_2}(DO))$ (we use part $\widetilde{m_2}(DO)\subseteq\widetilde{N2}(\widetilde{m_2}(DO))$ of the fixpoint equation). By definition of $\widetilde{N2}$, there are two cases.

Case 1: $\neg B(t')\wedge t'=t$. Then, by definition of $\widetilde{N2}$, $(t',t)\in\widetilde{N2}(\widetilde{m_2}'(DO))$.

Case 2: $B(t') \wedge (t',t) \in (\widetilde{m_2}(c) * \widetilde{m_2}(DO))$, i.e.,

$$B(t') \ \wedge \ (t',t) \in (\widetilde{m_2}(c) \circ \widetilde{m_2}(DO)) \ \wedge \ t'\widetilde{m_2}(c) \subseteq dom(\widetilde{m_2}(DO)). \tag{A.1}$$

By the second conjunct, there is a state u with $(t',u) \in \widetilde{m_2}(c)$ and $(u,t) \in \widetilde{m_2}(DO)$. If $u \in INF(DO)$ then also $t' \in INF(DO)$, because any infinite sequence starting with u could be prefixed by t'. However, this contradicts a prior assumption, so that $u \notin INF(DO)$. Hence $(u,t) \in \widetilde{m_2}'(DO)$. Also, using the third conjunct of (A.1), $t'\widetilde{m_2}(c) \subseteq dom(\widetilde{m_2}'(DO))$ since $dom(\widetilde{m_2}'(DO)) = dom(\widetilde{m_2}(DO)) \backslash INF(DO)$, and because $v \in t'\widetilde{m_2}(c)$ implies, as before, $v \notin INF(DO)$. Hence, using also the first conjunct of (A.1), $B(t') \wedge (t',t) \in (\widetilde{m_2}(c) * \widetilde{m_2}'(DO))$, and hence, by the definition of $\widetilde{N2}$, $(t',t) \in \widetilde{N2}(\widetilde{m_2}'(DO))$.

($\supseteq$:) Assume $(t',t) \in \widetilde{N2}(\widetilde{m_2}'(DO))$. We intend to show $(t',t) \in \widetilde{m_2}'(DO)$. By the definition of $\widetilde{N2}$, we have two cases:

Case 1: $\neg B(t') \wedge t'=t$. Then, by the definition of $\widetilde{m_2}$, $(t',t) \in \widetilde{m_2}(DO)$. By $\neg B(t')$, no infinite $\widetilde{m_2}$vseq can start at t'. Hence $t' \notin INF(DO)$, and hence, by definition of $\widetilde{m_2}'$, $(t',t) \in \widetilde{m_2}'(DO)$.

Case 2: $B(t') \wedge (t',t) \in (\widetilde{m_2}(c) * \widetilde{m_2}'(DO))$, i.e.,

$$B(t') \ \wedge \ (t',t) \in (\widetilde{m_2}(c) \circ \widetilde{m_2}'(DO)) \ \wedge \ t'\widetilde{m_2}(c) \subseteq (dom(\widetilde{m_2}(DO)) \backslash INF(DO)), \tag{A.2}$$

using also $dom(\widetilde{m_2}'(DO)) = dom(\widetilde{m_2}(DO)) \backslash INF(DO)$.

First, we prove that $(t',t) \in \widetilde{m_2}(DO)$. By the monotonicity of $*$ in its second argument, $\widetilde{m_2}(c) * \widetilde{m_2}'(DO) \subseteq \widetilde{m_2}(c) * \widetilde{m_2}(DO)$, whence, by the second conjunct of (A.2), $(t',t) \in (\widetilde{m_2}(c) * \widetilde{m_2}(DO))$. From this, $B(t')$, and the definition of $\widetilde{N2}$, it follows that $(t',t) \in \widetilde{N2}(\widetilde{m_2}(DO))$, and hence, since $\widetilde{m_2}(DO)$ is a fixpoint of $\widetilde{N2}$, $(t',t) \in \widetilde{m_2}(DO)$ (using direction $\widetilde{m_2}(DO) \supseteq \widetilde{N2}(\widetilde{m_2}(DO))$ of the fixpoint equation). Secondly, we prove that $t' \notin INF(DO)$. For, suppose that $t' \in INF(DO)$. Then there is $\bar{t}$ with $(t',\bar{t}) \in \widetilde{m_2}(c)$ and $\bar{t} \in INF(DO)$, contradicting the third conjunct of (A.2). Hence $(t',t) \in \widetilde{m_2}'(DO)$, which was to be shown.

Thus we have proved that, under the assumption that $s' \in INF(DO)$, we can find a fixpoint of $\widetilde{N2}$ which is strictly contained in $\widetilde{m_2}(DO)$, in contradiction to the definition of $\widetilde{m_2}(DO)$. This proves that $s' \notin INF(DO)$.

(A2): There is an $\widetilde{m_2}$vseq (and hence a valid m_2-sequence) $s_0, \ldots, s_r$ with $s'=s_0$, $s_r=s$ and $\neg B(s_r)$. (In particular, it follows that $\neg B(s)$.)

Proof: We construct a valid $\widetilde{m_2}$-sequence starting with s', which cannot be infinite because, by the above, $s' \notin INF(DO)$. Initially, put $s'=s_0$. Then we use $(s_0,s) \in \widetilde{m_2}(DO) \subseteq \widetilde{N2}(\widetilde{m_2}(DO))$. Again there are two cases.

Case 1: $\neg B(s_0) \wedge s_0=s$. Then we may put $r=0$ to obtain the desired sequence.

Case 2: $B(s_0) \wedge (s_0,s) \in (\widetilde{m_2}(c) * \widetilde{m_2}(DO))$, i.e.

$$B(s_0) \ \wedge \ (s_0,s) \in (\widetilde{m_2}(c) \circ \widetilde{m_2}(DO)) \ \wedge \ s_0\widetilde{m_2}(c) \subseteq dom(\widetilde{m_2}(DO)).$$

Using the second conjunct, choose s_1 with $(s_0,s_1) \in \widetilde{m_2}(c)$ and $(s_1,s) \in \widetilde{m_2}(DO)$. Using $\widetilde{m_2}(DO) \subseteq \widetilde{N2}(\widetilde{m_2}(DO))$, there are again two cases to be considered for s_1. Be-

cause, by (C2), s' is not in $INF(DO)$, this construction will eventually stop with a desired sequence (validity follows from the construction and the fact that $B(s_j)$ holds of all states except the last one).

(B2): There is no $\widetilde{m_2}$vseq (and hence no valid m_2-sequence) $s_0, \ldots, s_r$ with $s'{=}s_0$, $B(s_r)$ and $s_r\widetilde{m_2}(c){=}\emptyset$.

Proof: Consider any arbitrary valid $\widetilde{m_2}$-sequence $s_0, \ldots, s_r$ with $s'{=}s_0$ and $B(s_r)$. We shall prove that $s_r\widetilde{m_2}(c){\neq}\emptyset$. By induction on j, we prove that

$$\forall j, 0{\leq}j{\leq}r\colon s_j\widetilde{m_2}(c){\subseteq}dom(\widetilde{m_2}(DO)).$$

Base $j{=}0$: We know that $(s_0, s){\in}\widetilde{m_2}(DO){=}\widetilde{N2}(\widetilde{m_2}(DO))$. By $B(s_0)$, this implies $(s_0, s){\in}(\widetilde{m_2}(c){*}\widetilde{m_2}(DO))$, and by the definition of $*$, $s_0\widetilde{m_2}(c){\subseteq}dom(\widetilde{m_2}(DO))$ as desired.

Step $j{\to}j{+}1$ ($j{<}r$): By the induction hypothesis, $s_j\widetilde{m_2}(c){\subseteq}dom(\widetilde{m_2}(DO))$. Since $(s_j, s_{j+1}){\in}\widetilde{m_2}(c)$, this implies $s_{j+1}{\in}dom(\widetilde{m_2}(DO))$. That is, there is some u such that $(s_{j+1}, u){\in}\widetilde{m_2}(DO)$. By $B(s_{j+1})$, we conclude (similarly as in the base case) that $(s_{j+1}, u){\in}(\widetilde{m_2}(c){*}\widetilde{m_2}(DO))$ and $s_{j+1}\widetilde{m_2}(c){\subseteq}dom(\widetilde{m_2}(DO))$.

Take $j{=}r$. Then the above yields $s_r\widetilde{m_2}(c){\subseteq}dom(\widetilde{m_2}(DO))$, i.e. there is some v with $(s_r, v){\in}\widetilde{m_2}(DO)$. By the previously shown result, i.e. (A2), there is a valid $\widetilde{m_2}$-sequence leading from s_r to v, and, in particular, $\neg B(v)$. Hence, by $B(s_r)$, this sequence must be of length at least 1 (where the length of a sequence $s_0, \ldots, s_r$ is defined as r), leading to $s_r\widetilde{m_2}(c){\neq}\emptyset$, which was to be proved.

Thus far, we have proved (A2), (B2) and (C2) for the pair $(s', s){\in}\widetilde{m_2}(DO)$. Together with Lemma 3.2($\Leftarrow$), it follows that $(s', s){\in}m_2(DO)$.

Proof of $\widetilde{m_2}(DO) \supseteq m_2(DO)$: Now assume that $(s', s){\in}m_2(DO)$. We shall prove that $(s', s){\in}\widetilde{m_2}(DO)$. By Lemma 3.2($\Rightarrow$), all of (A2), (B2) and (C2) are true and may be used in the proof. We will use a standard tree construction (e.g., [2]). We will consider trees whose nodes are labelled by states in Z. The same state may occur twice or more as a label of such a tree, but the children of a given node will always be labelled by mutually different states. More precisely, starting from s', we construct (i.e. define inductively) the following tree, called the s'-tree: [2]

- The root of the s'-tree is a node labelled by s'.
- If k is a node of the s'-tree labelled by a state t, then the set of children (i.e. direct successors) of k is a set of nodes which is in 1–1 correspondence with the set $\{t'{\in}Z \mid B(t) \wedge (t, t'){\in}m_2(c)\}$ and carries the states from this set as labels.

As a consequence of this definition, the leaves of the s'-tree are the set of nodes k for whose labels t we have $\neg B(t) \vee tm_2(c){=}\emptyset$. Property (B2) implies that the states t at the leaves of the s'-tree – and, by the definition of the tree, only those – satisfy $\neg B(t)$. Property (A2) implies that s actually occurs as a label of one of the leaves of the tree. Property (C2) implies that the s'-tree has no infinite paths (however, the s'-tree could still be infinite, or even have no upper bound for the

[2] It is unique up to isomorphism.

lengths of paths). It also has another pleasant property: from each node there is at least one finite path to some leaf (the length of a path is defined as the number of nodes on it, minus 1). This follows immediately from the absence of infinite paths: a node is either already a leaf (in which case a path of length 0 leads from it to itself), or a path can be constructed by successively following through children and children's children, a construction which must stop eventually because there is no infinite path.

We will now show $(s', s) \in \widetilde{m_2}(DO)$ by proving two claims, (Ca) and (Cb), simultaneously. Let k be any node on the s'-tree and let t be its label. Then,

(Ca) if $B(t)$ then $tm_2(c) \subseteq dom(\widetilde{m_2}(DO))$;

(Cb) if x is a leaf with label u and a path leads from k to x, then $(t, u) \in \widetilde{m_2}(DO)$.

We prove (Ca) and (Cb) by induction on the smallest length, l, of a path from k to a leaf; this number is well-defined because at least one such path exists.

Base l=0: Then k is itself a leaf. Then, for the label t of k, we have $\neg B(t)$, which implies that (Ca) is satisfied for k and t. Moreover, by definition of $\widetilde{N2}$, $(t, t) \in \widetilde{N2}(\widetilde{m_2}(DO))$ by the fact that $(t, t) \in ((\neg B \times Z) \cap id_Z)$. By the fixpoint equation, $(t, t) \in \widetilde{m_2}(DO)$. Hence (Cb) is also satisfied, since the path from k to k (of length 0) is the only one to be considered.

Step $l \to l+1$: Let k be a node with label t, whose smallest distance to a root is $l+1$. Then $B(t)$, since k is not a leaf. Let k' be any child of k and t' its label; then and only then, by definition of the tree, $t' \in tm_2(c)$. We will prove $t' \in dom(\widetilde{m_2}(DO))$. Consider any path from k' to some leaf x with label u. By (Cb) for k' and t' (which is true by the induction hypothesis), $(t', u) \in \widetilde{m_2}(DO)$. Hence $t' \in dom(\widetilde{m_2}(DO))$, which proves (Ca) for k and t. To prove (Cb) for k and t, consider any path from t to some leaf y with label v and let k'' with label t'' be the child of k on that path (such a child exists by the fact that k is not a leaf, and it is unique by the general properties of trees). For k'' and t'', the induction hypothesis applies again, and hence $(t'', v) \in \widetilde{m_2}(DO)$ by (Cb). By the definition of $*$ and by the already proved property (Ca) for t, we have $(t, v) \in (m_2(c) * \widetilde{m_2}(DO))$, hence $(t, v) \in (\widetilde{m_2}(c) * \widetilde{m_2}(DO))$ by the general induction hypothesis that $m_2(c) = \widetilde{m_2}(c)$. Adding to this the fact that $B(t)$, we have $(t, v) \in \widetilde{N2}(\widetilde{m_2}(DO))$ and $(t, v) \in \widetilde{m_2}(DO)$ by the fixpoint equation, part $(\supseteq)$.

The above may be applied to the root with label s' and a particular leaf with label s (such a leaf exists by $(s', s) \in m_2(DO)$ and (A2)) of the s'-tree. Then (Cb) yields $(s', s) \in \widetilde{m_2}(DO)$, which was to be proved. □

Proof of Lemma 4.5: It is easy to check that $(2^{Z_\perp}, \sqsubseteq)$ is isomorphic to the direct lattice product between $(2^Z, \subseteq)$ and $(2^{\{\perp\}}, \supseteq)$ (reverse ordering for the latter lattice). Such products inherit all the nice properties from their components. □

More precisely, let $\{X_j \mid j \in J\}$ be any set of subsets of $2^{Z_\perp}$ (with index set J). Then

$$\sqcap X_j = \begin{cases} \bigcap (X_j \cup \{\perp\}) & \text{if } \exists X_j \colon \perp \in X_j \\ \bigcap X_j & \text{if } \forall X_j \colon \perp \notin X_j \end{cases} \quad \text{and} \quad \sqcup X_j = \begin{cases} \bigcup (X_j \setminus \{\perp\}) & \text{if } \exists X_j \colon \perp \notin X_j \\ \bigcup X_j & \text{if } \forall X_j \colon \perp \in X_j \end{cases}$$

are the greatest lower and least upper bounds, respectively, of $\{X_j \mid j \in J\}$.

Proof of the properties of $\sqsubseteq$: Because $\sqsubseteq$ is the ordering of the direct product of two lattices enjoying these properties, they are bequeathed upon $\sqsubseteq$.

Proof that $\diamond$ is monotonic with respect to $\sqsubseteq$ in its second argument:

Let μ, ρ_1, ρ_2 be relations $\subseteq Z \times Z_\perp$ with $\rho_1 \sqsubseteq \rho_2$. We show $(\mu \diamond \rho_1) \sqsubseteq (\mu \diamond \rho_2)$ by proving that for arbitrary $a \in Z$, $a(\mu \diamond \rho_1) \sqsubseteq a(\mu \diamond \rho_2)$. More precisely, we need to show that (1): $a(\mu \diamond \rho_1)\backslash\{\perp\} \subseteq a(\mu \diamond \rho_2)$, and (2): $\perp \in a(\mu \diamond \rho_1) \vee \perp \notin a(\mu \diamond \rho_2)$.

$$
\begin{aligned}
\text{Ad (1): } b \in a(\mu \diamond \rho_1)\backslash\{\perp\} &\Rightarrow \text{(definition of } \diamond) && b \in a(\mu \circ \rho_1)\backslash\{\perp\} \\
&\Rightarrow \text{(definition of } \circ) && \exists x\colon x \in a\mu \wedge b \in x\rho_1\backslash\{\perp\} \\
&\Rightarrow (\rho_1 \sqsubseteq \rho_2) && x \in a\mu \wedge \exists x\colon b \in x\rho_2\backslash\{\perp\} \\
&\Rightarrow \text{(definition of } \circ) && b \in a(\mu \circ \rho_2)\backslash\{\perp\} \\
&\Rightarrow \text{(definition of } \diamond) && b \in a(\mu \diamond \rho_2)\backslash\{\perp\}.
\end{aligned}
$$

$$
\begin{aligned}
\text{Ad (2): } \perp \in a(\mu \diamond \rho_2) &\Rightarrow \text{(definition of } \diamond)\ (a,\perp) \in \mu \vee \exists x\colon (a,x) \in \mu \wedge (x,\perp) \in \rho_2 \\
&\Rightarrow (\rho_1 \sqsubseteq \rho_2)\ (a,\perp) \in \mu \vee \exists x\colon (a,x) \in \mu \wedge (x,\perp) \in \rho_2 \wedge (x,\perp) \in \rho_1 \\
&\Rightarrow \text{(definition of } \circ)\ (a,\perp) \in \mu \vee (a,\perp) \in \mu \circ \rho_1 \\
&\Rightarrow \text{(definition of } \diamond \text{ and } \mu \circ \rho_1 \subseteq \mu \diamond \rho_1)\ \perp \in a(\mu \diamond \rho_2).
\end{aligned}
$$

Proof of Proposition 4.6: We only need to prove $\widetilde{m}(DO) = m(DO)$ for $DO =$ **do** $B \to c$ **od**, assuming as an induction hypothesis that $\widetilde{m}(c) = m(c)$.

Proof of $\widetilde{m}(DO) \sqsubseteq m(DO)$: The smallest fixpoint of $\widetilde{N}$ can be expressed as follows:

$$\widetilde{m}(DO) \quad = \quad \sqcap\{\rho \mid \widetilde{N}(\rho) \sqsubseteq \rho\}.$$

Hence $\widetilde{m}(DO)$ is below (in the sense of $\sqsubseteq$) every element of the set $\{\rho \mid \widetilde{N}(\rho) \sqsubseteq \rho\}$. Therefore, in order to prove the claim, it suffices to show that $m(DO)$ is a member of this set. Thus, we will show $\widetilde{N}(m(DO)) \sqsubseteq m(DO)$, and more precisely:

$$\forall s' \in Z\colon \underbrace{(s'\widetilde{N}(m(DO))\backslash\{\perp\} \subseteq s'm(DO))}_{(1)} \wedge \underbrace{(\perp \in s'\widetilde{N}(m(DO)) \vee \perp \notin s'm(DO))}_{(2)}.$$

Ad (1): Let $s \in s'\widetilde{N}(m(DO)\backslash\{\perp\}$. By the definition of $\widetilde{N}$, either $\neg B(s')$ or $B(s') \wedge (s',s) \in \widetilde{m} \diamond m(DO)$.

In the first case, clearly, $s \in s'm(DO)$, since a valid m-sequence (of length 0) leads from s' to s, and hence (A) of the definition of $m(DO)$ holds true.

In the second case, $s \neq \perp$ implies that $B(s') \wedge (s',s) \in \widetilde{m} \circ m(DO)$. The induction hypothesis $\widetilde{m}(c) = m(c)$ then further implies that there is some $t \in Z$ with

$$B(s') \wedge (s',t) \in m(c) \wedge (t,s) \in m(DO).$$

$(t,s) \in m(DO)$ implies (A), (B) or (C) for the pair (t,s), and no matter which one of these is the case, it will remain the case for (s',s) because of $B(s') \wedge (s',t) \in m(c)$.

Hence $(s', s) \in m(DO)$, and (1) is proved.

Ad (2): Assume that $\bot \in s'm(DO)$. We prove that $\bot \in s'\widetilde{N}(m(DO))$. By $(s', \bot) \in m(DO)$ and the definition of $m(DO)$, we have either (B) or (C) for the pair $(s', \bot)$.

In the first case, there is a sequence of states $s_0, s_1, \ldots, s_r$ with $r \geq 1$, $s'=s_0$, $s_r=\bot$ and $B(s_j) \wedge (s_j, s_{j+1}) \in m(c)$ for all $0 \leq j < r$. If $r=1$ and $s_1=\bot$, the definition of $\diamond$ yields $(s', \bot) \in m(c) \diamond m(DO)$, and hence (taking into account $B(s')$ and the induction hypothesis $\widetilde{m}(c) = m(c)$) also $(s', \bot) \in \widetilde{N}(m(DO))$. If $r>1$, then by (B) applied to the pair $(s_1, \bot)$, $(s_1, \bot) \in m(DO)$; hence the definition of $\diamond$ (together with $(s', s_1 \in m(c))$ yields again that $(s', \bot) \in m(c) \diamond m(DO)$, and together with $B(s')$ and the induction hypothesis we have $(s', \bot) \in \widetilde{N}(m(DO))$. In all cases, $\bot \in s'\widetilde{N}(m(DO))$, which proves (2). This ends the proof of $\widetilde{m}(DO) \sqsubseteq m(DO)$. [3]

Proof of $\widetilde{m}(DO) \sqsupseteq m(DO)$: We need to show:

$$\forall s' \in Z: \underbrace{(s'm(DO) \backslash \{\bot\} \subseteq s'\widetilde{m}(DO))}_{(1)} \wedge \underbrace{(\bot \in s'm(DO) \vee \bot \notin s'\widetilde{m}(DO))}_{(2)}.$$

Ad (1): Let $s \in s'm(DO) \backslash \{\bot\}$. Then (A) holds for the pair (s', s). That is, there is a valid m-sequence $s_0, s_1, \ldots, s_r$ with $s'=s_0$, $s_r=s$ and $B(s_j) \wedge (s_j, s_{j+1}) \in m(c)$ for all $0 \leq j < r$. If $r=0$, then clearly $(s', s) \in \widetilde{m}(DO)$ because of the first disjunct in the definition of $\widetilde{N}$. If $r>0$, then $B(s')$ and $(s_1, s) \in m(DO)$, and again $(s', s) \in \widetilde{m}(DO)$ because of the second disjunct of the definition of $\widetilde{N}$ (and the induction hypothesis $\widetilde{m}(c) = m(c)$). In all cases, $s \in s'\widetilde{m}(DO)$, which proves (1).

Ad (2): Assume $\bot \in s'\widetilde{m}(DO)$. We prove $\bot \in s'm(DO)$. Using $\bot \in s'\widetilde{m}(DO)$, the fixpoint equation $\widetilde{m}(DO) = \widetilde{N}(\widetilde{m}(DO))$ and the definition of $\diamond$, we may construct $s_0, s_1, \ldots$ starting with $s'=s_0$. If we ever get to the second disjunct of $\diamond$, then we will have (B) and thus $(s', \bot) \in m(DO)$. Otherwise, the construction continues indefinitely and then we have (C) and hence, again, $(s', \bot) \in m(DO)$. In both cases, $\bot \in s'm(DO)$, which proves (2). This ends the proof of $\widetilde{m}(DO) \sqsupseteq m(DO)$.

The main result now follows from the antisymmetry of $\sqsubseteq$. □

[3] One may be tempted to use the same idea in order to simplify the first part of the proof of proposition 4.3. However, this does not seem to work as nicely as it does here.