Relational Semantics Revisited

Eike Best[1] and Kerstin M. Richter[2]

[1] Carl-von-Ossietzky–Universität Oldenburg, D-26111 Oldenburg
[2] Universität Hildesheim, D-31141 Hildesheim

Abstract. We examine in this paper to what extent flat domains containing a 'bottom' element, domain constructions such as Plotkin's, and special orderings such as the Egli-Milner ordering or the Smyth ordering can be eschewed (i.e., replaced, respectively, by sets, powersets and the subset relation) in the denotational definition of the relational semantics of (possibly infinitely) nondeterministic sequential programs.

1 Introduction

A common way [3, 17] of defining the semantics of a loop such as

$$DO \quad = \quad \mathbf{do}\ B \rightarrow c\ \mathbf{od}$$

is to consider the flat domain $Z_\perp = Z \mathbin{\dot{\cup}} \{\perp\}$, where Z is the state space, and give the semantics $[\![DO]\!]$ as a function from $Z_\perp$ to $P(Z_\perp)$, where $P(Z_\perp)$ is the set of all nonempty subsets of $Z_\perp$ which are finite or contain $\perp$ [15]. On $P(Z_\perp)$, the Egli-Milner ordering, is defined as follows [1], for $X, Y \in P(Z_\perp)$:

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

This yields a complete partial order $(P(Z_\perp), \sqsubseteq_{EM})$ with least element $\{\perp\}$, which can be extended to the set of functions from $Z_\perp$ to $P(Z_\perp)$ by pointwise definition: $f \sqsubseteq_{EM} g$ iff $\forall s {\in} Z_\perp : f(s) \sqsubseteq_{EM} g(s)$. We may then consider the following function $K = K_{DO}$, which depends on DO:

$$K : \begin{cases} (Z_\perp \to P(Z_\perp)) & \to \quad (Z_\perp \to P(Z_\perp)) \\ \qquad f & \mapsto \quad K(f) : \begin{cases} Z_\perp \to P(Z_\perp) \\ s' \ \mapsto \begin{cases} \{s'\} & \text{if } B(s'){=}\mathit{false} \\ \bigcup_{t \in [\![c]\!]} f(t) & \text{if } B(s'){=}\mathit{true} \\ \{\perp\} & \text{if } B(s'){=}\perp. \end{cases} \end{cases} \end{cases} \tag{1}$$

Then K is monotonic with respect to $\sqsubseteq_{EM}$, and the least fixpoint of K describes the semantics $[\![DO]\!]$. Similar constructions apply traditionally if only partial correctness semantics or only total correctness semantics – rather than full semantics, as above – are to be considered, leading to Hoare and Smyth powerdomains (cf. appendix A).

Mathematically and intuitively, this approach is fine, but æsthetically, we would argue that it has some weak points:

(1) The assumption that nondeterminism is finite is already built into the definition of $P(Z_\perp)$. A typical infinitely nondeterministic command is $x :=?$, setting x to any integer value greater or equal to zero. The set of final states is then infinite but does not contain $\perp$, and hence is not a member of $P(Z_\perp)$.
(2) We would like to interpret the artificial state $\perp$ as 'error' or as 'nontermination'. However, it is then counterintuitive to view commands such as, for example, **skip**, to be executable in this state. Also, even for terminating programs p, the calculation of $[\![p]\!]$ involves the error state $\perp$; this indicates that, at least, the $\perp$ state should be interpreted more widely.
(3) Functions $[\![p]\!]: Z_\perp \to P(Z_\perp)$ – which can alternatively be viewed as relations $[\![p]\!] \subseteq (Z_\perp \times Z_\perp)$ – are not the 'leanest' objects describing the input/output behaviour of p. It is shown in [6] that $[\![p]\!]$ can be replaced by a 'smaller' relation $m(p) \subseteq (Z \times Z_\perp)$; this gets rid of the possibility of starting a command in the error state. Moreover, two 'even smaller' relations $m_1(p) \subseteq Z \times Z$ and $m_2(p) \subseteq Z \times Z$ can be derived from $m(c)$ such that neither is equivalent with $m(c)$ (but both together are), and such that $m_1(p)$ is the exact formal equivalent of Hoare's (original) proof rules [13], and thus captures partial correctness semantics, and $m_2(p)$ is the exact equivalent of Dijkstra's wp semantics [9], and thus captures total correctness semantics.
(4) We consider using the Egli-Milner ordering – or any other ordering which is nonstandard in terms of set theory– to be fully justified only if more standard partial orderings (such as plain subset inclusion) demonstrably do not work. In this paper, we show that set inclusion can be used in almost all cases and that the Egli-Milner ordering is based on a systematic lattice-theoretic principle.

These æsthetic problems (may) lead to mathematical complications, and it is therefore reasonable to ask whether they can be removed. Such questions have already been discussed in the literature. As to item (1), Hesselink shows in [12] that the Egli-Milner ordering can be generalised in order to encompass infinite nondeterminism, but it is not discussed whether $\perp$ can be avoided. As to item (3), in [10] Doornbos has argued that $\perp$ can be avoided in special cases, but no operational or other general formal justifications have been given.
In this paper, we proceed as follows. First, in section 2, we introduce the three relations $m(p)$, $m_1(p)$ and $m_2(p)$ operationally as the smallest ways of capturing the, respectively, full, partial correctness, and total correctness relational semantics of sequential nondeterministic programs. In section 3, we ask (and answer) the question whether there are other ways of defining $m(p)$, $m_1(p)$ and $m_2(p)$ by denotational means, using a minimum amount of non-standard set-theoretic domain constructions. The main result of this paper is that, indeed, there are such ways. More precisely, we show that:

- The relations $m_1(p)$ and $m_2(p)$ can be defined purely set-theoretically, and moreover, by resorting only to standard complete lattices rather than any more general domains.
- A similar remark is true for the relation $m(p)$, but the lattice theoretic constructions are slightly more involved.

All of our results hold for programs with arbitrary (including infinite) nondeterminism. Proofs are given in the appendices. In appendix A, we also recall the traditional denotational definitions of the above three relations using, respectively, the Egli-Milner, the Hoare, and the Smyth orderings. These definitions rely on well-known powerdomain constructions.

2 Syntax and operational / relational semantics

Section 2.1 defines syntax, section 2.2 contains some of the mathematical notation we use, and sections 2.3, 2.4 and 2.5 give operational definitions for full, partial and total correctness semantics, respectively.

2.1 Syntax and assumptions

We use the following small syntax for programs p (see e.g. [6, 9]):

p ::= **skip** | **abort** | x:=E | x:=? | p_1;p_2 | **if** $B_1 \rightarrow p_1 \,\Box\, B_2 \rightarrow p_2$ **fi** | **do** $B \rightarrow p$ **od**.

We often abbreviate **if** $true \rightarrow p_1 \,\Box\, true \rightarrow p_2$ **fi** by p_1 **or** p_2. Using this syntax, one may express infinite nondeterminism as well as iteration. E and B are expressions and Boolean expressions, respectively. We assume that the evaluation of an expression in a state never leads to an error (errors may be introduced by **abort**, by non-applicable **if** commands, or by nonterminating **do** commands).
All variables are assumed to be typed, so that we can associate, as usual, a state space $Z(p)$ (the set of functions from variables to values) with every program p. Unless stated otherwise, we assume a program to have only a single variable, x, whose type is the set $\mathbb{Z}$ of integers. Also, to simplify formulae, we will often write Z instead of $Z(p)$ and m instead of $m(p)$ if the program p in question is understood. Also, $Z_\perp(p) = Z(p) \uplus \{\perp\}$ and, as before, $Z_\perp = Z \uplus \{\perp\}$. We call $s \in Z_\perp$ 'proper' if $s \neq \perp$. We will use the following notation consistently:

p	denotes an arbitrary program
DO	denotes a loop with entry condition B and body c.

2.2 Notations and Definitions

Let S be any set. The powerset of S is denoted by 2^S, and the disjoint union of two sets is denoted by $\uplus$. Let $\rho \subseteq S \times S$ be any relation on S. The set $dom(\rho)$ is defined as $\{x \in S \mid \exists y \in S \colon (x, y) \in \rho\}$. 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 $\rho_1, \rho_2 \subseteq S \times S$ be any two relations on S. Relational composition is defined by $\rho_1 \circ \rho_2 \subseteq S \times S$ and $(a, b) \in \rho_1 \circ \rho_2 \Leftrightarrow \exists x \in S \colon (a, x) \in \rho_1 \wedge (x, b) \in \rho_2$.
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'.

2.3 Full relational semantics $m(p)$

We associate a meaning relation $m(p) \subseteq Z(p) \times Z_\perp(p)$ with every program p, such that $(s', s) \in m(p)$ expresses the following operational intuition: *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 $s = \perp$).* We define $m(p) \subseteq Z \times Z_\perp$ by induction on the syntax of p:

$$
\begin{array}{ll}
m(\mathbf{skip}) & = \{(s', s') \mid s' \in Z\} \\
m(\mathbf{abort}) & = \{(s', \perp) \mid s' \in Z\} \\
(s', s) \in m(x{:=}E) & \Leftrightarrow s = s'[x \leftarrow E] \text{ (section 2.2 explains notation)} \\
(s', s) \in m(x{:=}?) & \Leftrightarrow \exists i \in \mathbb{N}\colon s = s'[x \leftarrow i] \\
(s', s) \in m(p_1 ; p_2) & \Leftrightarrow ((s', s) \in m(p_1) \circ m(p_2)) \\
 & \quad \vee \ (s = \perp \wedge (s', \perp) \in m(p_1)) \\
(s', s) \in m(\mathbf{if}\ B_1 \rightarrow p_1 \ \Box\ B_2 \rightarrow p_2\ \mathbf{fi}) & \Leftrightarrow \exists j \in \{1, 2\}\colon B_j(s') = \mathit{true} \wedge (s', s) \in m(p_j) \\
 & \quad \vee \ (s = \perp \wedge B_1(s') = B_2(s') = \mathit{false}) \\
(s', s) \in m(\mathbf{do}\ B \rightarrow c\ \mathbf{od}) & \Leftrightarrow \text{see below.}
\end{array}
$$

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

$$
\begin{array}{ll}
 & (s', s) \in m(DO) \\
\Leftrightarrow & [s \neq \perp \wedge \exists \text{ valid } m\text{-sequence } s_0, \ldots, s_r\colon s' = s_0, s_r = s \text{ and } \neg B(s_r)] \\
 & \vee\ [s = \perp \wedge \exists \text{ valid } m\text{-sequence } s_0, \ldots, s_r\colon s' = s_0 \text{ and } s_r = \perp] \\
 & \vee\ [s = \perp \wedge \exists \text{ infinite valid } m\text{-sequence } s_0, s_1, \ldots\colon s' = s_0].
\end{array}
$$

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.

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

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

Later, we will need to use $((s', s) \in m(DO) \wedge \perp \notin s' m(DO))$, which we therefore rewrite as follows:

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

(A) rewrites $(s', s) \in m(DO)$ and (B)$\wedge$(C) rewrites $\perp \notin s' m(DO)$.
From $m(p)$, we derive two relations $m_1(p), m_2(p) \subseteq Z(p) \times Z(p)$ as follows:

$$\begin{array}{l} m_1(p) = m(p) \cap (Z(p) \times Z(p)) \\ m_2(p) = m(p) \cap ((Z(p) \backslash (m(p) \perp)) \times Z(p)), \end{array} \tag{2}$$

where $m(p)\perp = \{s \in Z(p) \mid (s, \perp) \in m(p)\}$ (see section 2.2).
The relation m_1 describes 'partial correctness semantics'; it simply forgets about nontermination and thus equates, for instance, the two programs **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] that $m_1(p)$ is semantically equivalent with Hoare's proof rules [13], while $m_2(p)$ is equivalent with 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 individually weaker than m, both both together are equivalent with m. In the next two subsections we show that, instead of deriving them from $m(p)$, the two relations $m_1(p)$ and $m_2(p)$ can also be defined inductively.

2.4 Partial correctness semantics $m'_1(p) = m_1(p)$

We define $m'_1(p) \subseteq Z \times Z$ by induction on the syntax of p (giving only those clauses where the definition differs from that of $m(p)$).

$$\begin{array}{lll} m'_1(\mathbf{skip}) & = & id|_Z \\ m'_1(\mathbf{abort}) & = & \emptyset \\ m'_1(p_1 ; p_2) & = & m(p_1) \circ m(p_2) \\ (s', s) \in m'_1(\mathbf{if}\ B_1 \rightarrow p_1 \ \Box\ B_2 \rightarrow p_2\ \mathbf{fi}) & \Leftrightarrow & \exists j \in \{1, 2\} : B_j(s') = true \wedge (s', s) \in m'_1(p_j) \\ (s', s) \in m'_1(\mathbf{do}\ B \rightarrow c\ \mathbf{od}) & \Leftrightarrow & \exists \text{ valid } m'_1\text{-sequence } s_0, \ldots, s_r : \\ & & \quad s' = s_0,\, s_r = s \text{ and } B(s_r) = \mathit{false}, \end{array}$$

where $s_0, \ldots, s_r$ is called a valid m'_1-sequence if $B(s_j) \wedge (s_j, s_{j+1}) \in m'_1(c)$ for all $0 \leq j < r$. Note how m'_1 equates **skip** and (**skip or abort**): the **abort** simply does not contribute to its definition. The following lemma shows that m_1 and m'_1 (and hence also the notion of an m'_1-sequence and an m_1-sequence, defined similarly using $m_1(c)$ instead of $m'_1(c)$) are the same.

Lemma 2.1.

Let p be any program. Then $m'_1(p) = m_1(p)$. ■ 2.1

2.5 Total correctness semantics $m_2'(p) = m_2(p)$

We define $m_2'(p) \subseteq Z \times Z$ by induction on the syntax of p (giving only those clauses where the definition differs from that of $m_1'(p)$).

$$
\begin{array}{ll}
(s',s) \in m_2'(p_1;p_2) & \Leftrightarrow ((s',s) \in m_2'(p_1) \circ m_2'(p_2)) \\
 & \quad \wedge \ (s' m_2'(p_1) \subseteq dom(m_2'(p_2))) \\
(s',s) \in m_2'(\mathbf{if}\, B_1 \to p_1 \,\Box\, B_2 \to p_2 \,\mathbf{fi}) & \Leftrightarrow \exists j \in \{1,2\}\colon B_j(s') = true \wedge (s',s) \in m_2'(p_j) \\
 & \quad \wedge \forall j \in \{1,2\}\colon B_j(s') = true \Rightarrow s' m_2'(p_j) \neq \emptyset \\
(s',s) \in m_2'(\mathbf{do}\, B \to c\, \mathbf{od}) & \Leftrightarrow \text{see below.}
\end{array}
$$

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 $0 \leq j$). Note that every valid m_2'-sequence is also a valid m-sequence, but not necessarily conversely. Then,

$$
\begin{array}{lr}
(s',s) \in m_2'(DO) & \\
\Leftrightarrow \quad [\exists \text{ valid } m_2'\text{-sequence } s_0, \ldots, s_r \colon s' = s_0, s_r = s \text{ and } \neg B(s_r)] & \text{(A2)} \\
\quad \wedge\, [\neg\exists \text{ valid } m_2'\text{-sequence } s_0, \ldots, s_r \colon s' = s_0, B(s_r) \text{ and } s_r m_2(c) = \emptyset] & \text{(B2)} \\
\quad \wedge\, [\neg\exists \text{ infinite valid } m_2'\text{-sequence } s_0, s_1, \ldots \colon s' = s_0]. & \text{(C2)}
\end{array}
$$

Note how the semantics of the **if** command takes the intended meaning of m_2 into account: a pair (s',s) is included only if all possible alternatives – not just the actually chosen one – terminate. This leads to the identification – by m_2' as well as m_2 – of **abort** and (**skip or abort**).

Lemma 2.2.

Let p be any program. Then $m_2'(p) = m_2(p)$. ■ 2.2

3 Three variations of denotational 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 $\widetilde{m}(p)$ equals $m(p)$, $\widetilde{m_1}(p)$ equals $m_1(p)$ and $\widetilde{m_2}(p)$ equals $m_2(p)$. We proceed in a different order than before, because $\widetilde{m_1}(p)$ and $\widetilde{m_2}(p)$ can be treated uniformly and do not need any special nonstandard set-theoretic construction. The situation is slightly different for $\widetilde{m}(p)$.
Observe first that m_1 and m_2 do not involve the error state $\perp$; both are elements of $2^{Z \times Z}$. Since $2^{Z \times Z}$ is already provided with a canonical partial order, namely $\subseteq$, we will investigate this ordering. Thus, we now consider the following function, which should be compared with (1). We consider $A(c)$ and $\boxtimes$ to be parametric and the ordering $\subseteq$ to be fixed:

$$
\widetilde{N} : \begin{cases} (2^{Z \times Z}, \subseteq) \to (2^{Z \times Z}, \subseteq) \\ \qquad \rho \qquad \mapsto \widetilde{N}(\rho) = (\, (\neg B \times Z) \cap id_Z) \,\cup\, ((B \times Z) \cap (A(c) \boxtimes \rho)\,). \end{cases} \tag{3}
$$

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 (3) is that: *either B is false and the loop stops with unchanged initial state, or B is true and the body $A(c)$ is executed, 'after' (i.e., $\boxtimes$) which the same situation is repeated.*

3.1 Partial correctness semantics $\widetilde{m_1}(p) = m_1(p)$

In this section, we specialise equation (3) as follows:

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

Let us call $\widetilde{N}$, with these definitions, $\widetilde{N1}$. 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, and we define $\widetilde{m_1}(DO)$ to be this least fixpoint. The other clauses of $\widetilde{m_1}$ follow the same definition as that of m'_1.

Proposition 3.1.

$\widetilde{m_1}(p) = m_1(p)$ for all programs p. ■ 3.1

3.2 Total correctness semantics $\widetilde{m_2}(p) = m_2(p)$

Let us now search for a suitable relational composition $\boxtimes$ – if there exists one – such that the minimal fixpoint of $\widetilde{N}$, so defined, is exactly $\widetilde{m_2}(c)(DO)$. We claim that the composition $*$, defined as follows, does the job.

Definition 3.2. *Demonic composition* [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 $(a, b) \in (\sigma * \tau) \Leftrightarrow ((a, b) \in (\sigma \circ \tau) \wedge (a\sigma \subseteq dom(\tau)))$. ■ 3.2

The operation $*$ is associative and monotonic in the second argument (see appendix C). We now specialise equation (3) as follows:

$$\begin{array}{rcll} A(c) & = & \widetilde{m_2}(c) & \text{(known inductively)} \\ \boxtimes & = & * & \text{(definition 3.2).} \end{array}$$

Let $\widetilde{N2}$ be defined as $\widetilde{N}$ with this choice. By the above, $\widetilde{N2}$ is, in its turn, monotonic with respect to $\subseteq$, and hence has a unique minimal fixpoint. We define relation $\widetilde{m_2}(DO)$ 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:

Proposition 3.3.

$\widetilde{m_2}(p) = m_2(p)$ for all programs p. ■ 3.3

In section 3.3, we give three examples which will (hopefully) illustrate why $*$ had to be defined in the above, rather than any other, way.

3.3 Examples

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

$$\begin{aligned}\gamma_2 &= \delta_2 \cup \{((x{=}k),(x{=}0)) \mid k < 0\}\\ \delta_2 &= \{((x{=}i),(x{=}i)) \mid i \geq 0\}.\end{aligned}$$

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, $m1(p_2) = \gamma_2$ and $m2(p_2) = \delta_2$. Note how the pair $((x{=}{-1}),(x{=}0))$ is added to $m1(p_2)$ 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 $m1(p_2)$ 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.

3.4 Full relational semantics $\widetilde{m}(p)$

Since $m(p)$ is a subset of $Z \times Z_\perp$, (3) is no longer applicable, and we are led to consider the set of relations $2^{Z\times Z_\perp}$ and a suitable partial order (as well as a suitable relational composition such as $\boxtimes$) on it. Keeping in mind that both $\boxtimes = \circ$ in section 3.1 and $\boxtimes = *$ in section 3.2 precisely mimic the sequential composition in the definitions of m'_1 (section 2.4) and m'_2 (section 2.5), the following is an immediate candidate, since it mimics the sequential composition as defined in section 2.3.

Definition 3.4. *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 $(a,b) \in (\sigma \diamond \tau) \Leftrightarrow ((a,b) \in (\sigma \circ \tau) \vee ((a,\perp) \in \sigma \wedge b{=}\perp)$. ■ 3.4

Using this composition and a parametric partial ordering relation $\sqsubseteq$ on $2^{Z\times Z_\perp}$, we are thus led to consider the following function:

$$N : \begin{cases} (2^{Z\times Z_\perp}, \sqsubseteq) \to (2^{Z\times Z_\perp}, \sqsubseteq) \\ \qquad \rho \qquad\quad \mapsto N(\rho) = ((\neg B \times Z) \cap id_Z) \cup ((B \times Z_\perp) \cap (\widetilde{m}(c) \diamond \rho)) \end{cases} \tag{4}$$

where $\widetilde{m}(c)$ can be assumed known by induction. This is to be understood such that N actually maps $2^{Z\times Z_\perp}$ to $2^{Z\times Z_\perp}$, and that $\sqsubseteq\ \subseteq\ 2^{Z\times Z_\perp} \times 2^{Z\times Z_\perp}$ is the relation with respect to which the fixpoint of N will be taken.
The example $p_4 =$ **var** $x : \{0\}$; **do** $x{=}0 \to$ **skip od** shows that neither $\subseteq$ nor $\supseteq$ can be chosen as $\sqsubseteq$. This is because the m semantics of p_4 is $m(p_4) = \{(x{=}0, \perp)\}$, but the least fixpoints of N under $\sqsubseteq = \subseteq$ and under $\sqsubseteq = \supseteq$ 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, let us provide the set $2^{Z_\perp}$ with the following ordering. For $X, Y \in 2^{Z_\perp}$:

$$X \sqsubseteq Y \quad \Leftrightarrow \quad (X \backslash \{\perp\} \subseteq Y) \wedge (\perp \in X \vee \perp \notin Y).$$

Then,

Lemma 3.5.

$(2^{Z_\perp}, \sqsubseteq)$ is a complete lattice.

The proof of this lemma shows that $(2^{Z_\perp}, \sqsubseteq)$ 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 $\subseteq$ has the same meaning as before. The fact that $\{\perp\}$ lies 'below' $\emptyset$ can be interpreted as saying that the former represents less definite information than the latter. This corresponds to the idea behind the Egli-Milner ordering; indeed, on the subset $P(Z_\perp)$ of $2^{Z_\perp}$, $\sqsubseteq$ generalises $\sqsubseteq_{EM}$ in the sense of satisfying $\sqsubseteq_{EM}\ \subseteq\ \sqsubseteq$.
The partial order $\sqsubseteq$ can be lifted to relations $\rho \subseteq Z \times Z_\perp$ by putting

$$\rho_1 \sqsubseteq \rho_2 \quad \Leftrightarrow \quad \forall s \in Z : s\rho_1 \sqsubseteq s\rho_2 .$$

Then $(2^{Z \times Z_\perp}, \sqsubseteq)$ is another complete lattice. In particular, $\sqsubseteq$ is reflexive, transitive, antisymmetric. It is also a congruence with respect to $\cap$ and $\cup$, and moreover, $\diamond$ is monotonic with respect to $\sqsubseteq$ in its second argument. Hence we may define $\widetilde{m}(DO)$ as the least fixpoint of N in (4), and $\widetilde{m}(p)$ for all other program constructs p as in the definition of $m(p)$.
Then we get the second main result of this paper:

Proposition 3.6.

$\widetilde{m}(p) = m(p)$ for all programs p. ■ 3.6

4 Conclusion

We have shown in sections 3.1 and 3.2 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 come as a surprise, since the traditional Hoare ordering is the same as subset ordering (see appendix A.2). The latter (and the concomitant result, proposition 3.3) came as more of a surprise to us, since prevailing opinion links total correctness semantics to the Smyth ordering which uses the artificial $\perp$ state (see appendix A.3).
For full relational semantics (section 3.4), we have shown that pure subset or superset orderings are insufficient if $\diamond$ is taken as the underlying relational composition, but that it suffices to consider a lattice product of two lattices, one of which uses subset ordering and the other one of which uses superset ordering. In fact, the Egli-Milner ordering can be seen as an abridged form of this ordering. However, because of its simplicity and its systematic connection to an ordering derived from a lattice product, we find the $\sqsubseteq$ more pleasing than the traditional $\sqsubseteq_{EM}$ defined in the introduction (and used in appendix A.1). It remains to be checked whether a relational composition different from $\diamond$ can be found which makes an analogue of proposition 3.6 true for pure subset or superset orderings. The result of section 3.4, proposition 3.6, is (after the work of Hesselink [12]) perhaps less surprising than that of section 3.2.
Some of the reasoning that we have used to motivate this work has been voiced before. For instance, arguments such as we have made in the introductory section can be found in [4] (pages 128 pp.) and in [10]. The demonic composition $*$ has been defined previously: e.g., implicitly in [5] and explicitly in [14]. In [5, 11], moreover, 'functionals' such as $\widetilde{N1}$ and $\widetilde{N2}$ are used.
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 (continuous)

function $\widetilde{N2}$, coming from the present paper, applied to the same example, which yields stability of approximation after only two steps, using the empty relation as starting point.

Acknowledgments

We would like to thank Jaco W. de Bakker, Rudolf Berghammer, Wim H. Hesselink and Burghard von Karger for their willingness to discuss this subject and for helping out with pointers to their work. Burghard von Karger also detected a mistake and suggested the simple proof of lemma 3.5.

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. T.T. Nguyen: A Relational Model of Demonic Nondeterministic Programs. *Intern. J. Found. of Comp. Sci.* 2, 101–131 (1991).
15. G.D. Plotkin: A Powerdomain Construction. *SIAM J. Comp.* 5, 452–486 (1967).
16. K.M. Richter: Denotationale Formulierungen für relationale Semantiken. *Diplomarbeit*, Universität Hildesheim (1998).
17. D. Schmidt: *Denotational Semantics: a Methodology for Language Development.* Allyn and Bacon (1986).

A Standard powerdomain constructions

The purpose of this section is to provide Egli-Milner, Hoare and Smyth constructions for, respectively, $m(p)$, $m_1(p)$ and $m_2(p)$. We focus on the loop $DO = \mathbf{do}\, B \rightarrow c\, \mathbf{od}$, 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. The 'standard' denotational semantics of DO, to be recalled in this section, employs the fixpoint scheme that has been presented in section 1. However, we will reformulate it in terms of relations, because that is what will make these considerations uniform with those of section 3. Moreover, in the 'standard' definitions, the relational composition that has been a parameter+ in the previous sections (variously called $\boxtimes$, $*$, $\diamond$ etc.) is uniformly the relational composition $\circ$. This implies that the main variations in the definitions must be in the domain(s) and its partial orderings. The proofs of the claims in this section are standard, and are therefore omitted.
First, consider the following function $\widehat{N}$, where D, $\sqsubseteq$ and $\widehat{A}(c)$ are parametric:

$$\widehat{N} : \begin{cases} (D, \sqsubseteq) \rightarrow (D, \sqsubseteq) \\ \quad \rho \quad \mapsto (\ (\neg B \times Z) \cap id_Z)\ \cup\ (B_\perp \times \{\perp\})\ \cup\ ((B \times Z_\perp) \cap (\widehat{A}(c) \circ \rho)\ )\ . \end{cases} \tag{5}$$

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 non-$\perp$ initial state does not lead to any error, equals $\{\perp\}$), and B is the set of states for which B evaluates to *true*.

A.1 Full relational semantics and the Egli-Milner ordering

In this section we use (5) in order to define a relation $\widehat{m}(p) \subseteq Z \times Z_\perp$ which is exactly the same as $m(p)$. Since we are dealing with infinite nondeterminism, we incorporate an immediate generalisation, along the lines of [12], of the setup defined in section 1. Let $P(Z_\perp)$ now 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 as in section 1. 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, we may extend $\sqsubseteq_{EM}$ 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 $P(Z_\perp)$. We specialise (5) by specifying D, $\sqsubseteq$ and $\widehat{A}(c)$, as follows:

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

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 w.r.t. $\sqsubseteq_{EM}$. Hence $\widehat{N}$ has a minimal fixpoint, and we define $M(DO)$ as this smallest fixpoint. However, $M(DO) \subseteq Z_\perp \times Z_\perp$ is not of the same type as $m(DO) \subseteq Z \times Z_\perp$, and hence we define $\widehat{m}(DO) = M(DO) \cap (Z \times Z_\perp)$. The other clauses of $\widehat{m}(p)$ (i.e., when p is not a loop) follow the same definition as for $m(p)$.

Lemma A.1.

Let p be any program. Then $m(p) = \widehat{m}(p)$. ■ A.1

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, $M(DO) = \widehat{m}(DO) \cup \{(\perp, \perp)\}$ can be derived.

A.2 Partial correctness semantics and the Hoare ordering

In this section we use (5) in order to define a relation $\widehat{m_1}(p) \subseteq Z \times Z$ which is exactly the same as $m_1(p)$ (and hence also as $m_1'(p)$). Let $Q(Z_\perp)$ be defined as 2^Z (without $\perp$) and $\sqsubseteq_H \subseteq Q(Z_\perp) \times Q(Z_\perp)$ as ordinary inclusion $\subseteq$ on 2^Z. Let $Q(Z_\perp \times Z_\perp)$ denote the set of all relations $\rho \in 2^{Z_\perp \times Z_\perp}$ such that for all $s \in Z_\perp$, $s\rho \in Q(Z_\perp)$. This set is closed under relational composition. Moreover, we may extend $\sqsubseteq_H$ as follows to $Q(Z_\perp \times Z_\perp)$:

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

This is the relational analogue of extending $\sqsubseteq_H$ to functions from $Z_\perp$ to $Q(Z_\perp)$. In this section, we specialise equation (5) as follows:

$$\begin{aligned} D &= Q(Z_\perp \times Z_\perp) \\ \sqsubseteq &= \sqsubseteq_H \\ \widehat{A}(c) &= M_1(c). \end{aligned}$$

Let us call $\widehat{N}$, with these definitions, $\widehat{N1}$. Then $\widehat{N1}$ 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 w.r.t. $\sqsubseteq_H$. Hence $\widehat{N1}$ has a minimal fixpoint. We define $M_1(DO)$ as the smallest fixpoint of $\widehat{N1}$ with these parameters. Since $M_1(DO) \subseteq Z_\perp \times Z_\perp$ is not of the same type as $m_1(DO) \subseteq Z \times Z$, we define $\widehat{m_1}(DO) = M_1(DO) \cap (Z \times Z)$. The other clauses of $\widehat{m_1}(p)$ follow the same definition as for $m_1'(p)$.

Lemma A.2.

Let p be any program. Then $m_1(p) = \widehat{m_1}(p)$. ■ A.2

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

A.3 Total correctness semantics and the Smyth ordering

Finally, in this section we use (5) in order to define a relation $\widehat{m_2}(p) \subseteq Z\times Z$ which is exactly the same as $m_2(p)$ (and hence also as $m_2'(p)$). Let $R(Z_\perp)$ be defined as $(2^Z\backslash\{\emptyset\})\cup\{Z_\perp\}$ and $\sqsubseteq_S \subseteq R(Z_\perp)\times R(Z_\perp)$ as reverse inclusion $\supseteq$ on $R(Z_\perp)$. Let $R(Z_\perp\times Z_\perp)$ denote the set of all relations $\rho\in 2^{Z_\perp\times Z_\perp}$ such that for all $s\in Z_\perp$, $s\rho\in R(Z_\perp)$. Again, this set is closed under relational composition. Moreover, we may extend $\sqsubseteq_S$ to $R(Z_\perp\times Z_\perp)$:

$$\rho_1 \sqsubseteq_S \rho_2 \Leftrightarrow \forall s\in Z_\perp\colon s\rho_1 \sqsubseteq_S s\rho_2.$$

We specialise equation (5) as follows:

$$\begin{aligned} D &= R(Z_\perp\times Z_\perp) \\ \sqsubseteq &= \sqsubseteq_S \\ \widehat{A}(c) &= M_2(c). \end{aligned}$$

Let us call $\widehat{N}$, with these definitions, $\widehat{N2}$. 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. $\sqsubseteq_S$. Hence $\widehat{N2}$ has a minimal fixpoint. 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 $m_2(DO) \subseteq Z\times Z$, and we define $\widehat{m_2} \subseteq Z\times Z$ as follows: for $s'\in Z$,

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

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

Lemma A.3.

Let p be any program. Then $m_2(p) = \widehat{m_2}(p)$. ■ A.3

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

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

A.4 Discussion

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, had to be trimmed down to the 'operational' interpretation of failure, as described by m_2 (yielding $\emptyset$ as set of final states). The paper shows in section 3.2 that despite the above use of the Smyth ordering, yielding chaotic interpretation of failure, it is possible to give denotational semantics yielding non-chaotic (and thus operationally more satisfying) interpretation of failure.

B Proofs relating to section 2

Proof of lemma 2.1:
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 $m_1 = m \cap (Z \times Z)$. ■ 2.1

Proof of lemma 2.2:
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).) ■ 2.2

C Proofs relating to section 3

Proof of Proposition 3.1:
Easier than the proof of proposition 3.3, but along the same lines [16].

Proof that $*$ is associative, i.e., satisfies $(\rho * \tau) * \sigma = \rho * (\tau * \sigma)$:
By definition 3.2, $(a, b) \in (\rho * \tau) * \sigma$ can be rewritten as follows:

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

$$\wedge\ [\ \underbrace{\forall d''}_{(12)} : [\underbrace{(\exists c'' : (a,c'') \in \rho \wedge (c'', d'') \in \tau)}_{(121)} \wedge \underbrace{(\forall c''' : (a,c''') \in \rho \Rightarrow \exists d''' : (c''', d''') \in \tau)}_{(122)}]$$
$$\Rightarrow\ \underbrace{\exists b' : (d'', b') \in \sigma}_{(123)}\ ].$$

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

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

$$\wedge\ [\ \underbrace{\forall c'}_{(22)} : \underbrace{(a,c') \in \rho}_{(221)}\ \Rightarrow$$
$$(\ \exists b' : \underbrace{[\exists d'' : (c', d'') \in \tau \wedge (d'', b') \in \sigma]}_{(222)} \wedge [\underbrace{\forall d'''}_{(223)} : \underbrace{(c', d''') \in \tau}_{(2231)} \Rightarrow \exists\, \underbrace{b''' : (d''', b''') \in \sigma]}_{(2232)}\ )\ ].$$

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 3.3:
We may prove the result by induction on program p; however, we restrict ourselves to the loop $DO = \mathbf{do}\, B \rightarrow c\, \mathbf{od}$, 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 we have $((s', s) \in m(DO) \wedge \bot \notin s' m(DO)) \Leftrightarrow (s', s) \in m_2(DO)$, we may use, for the former, the three lines (A)–(C) in the definition of $m(DO)$, cf. section 2.3, and for the latter, the three lines (A2)–(C2) in the definition of $m_2'(DO)$, cf. section 2.5.
Before proving $\widetilde{m_2}(DO) = m_2(DO)$, we define some auxiliary notions. The notion of a valid $\widetilde{m_2}$-sequence (in what follows abbreviated by $\widetilde{m_2}$vseq) is defined as that of a valid m_2-sequence, except that m_2 is replaced by $\widetilde{m_2}$. Moreover, we define the set of states which start an infinite $\widetilde{m_2}$-execution, as follows:

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

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 2.2($\Leftarrow$).

(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.,

$$B(t') \;\wedge\; (t', t) \in (\widetilde{m_2}(c) \circ \widetilde{m_2}(DO)) \;\wedge\; t' \widetilde{m_2}(c) \subseteq dom(\widetilde{m_2}(DO)). \tag{6}$$

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 (6), $t'\widetilde{m_2}(c) \subseteq dom(\widetilde{m_2}'(DO))$ 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 (6), $B(t')\wedge(t',t)\in(\widetilde{m_2}(c)*\widetilde{m_2}'(DO))$, and hence, by the definition of $\widetilde{N}2$, $(t',t)\in\widetilde{N}2(\widetilde{m_2}'(DO))$.
($\supseteq$:) Assume $(t',t)\in\widetilde{N}2(\widetilde{m_2}'(DO))$. We intend to show $(t',t)\in\widetilde{m_2}'(DO)$. By the definition of $\widetilde{N}2$, 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.,

$$B(t') \ \wedge\ (t',t)\in(\widetilde{m_2}(c)\circ\widetilde{m_2}'(DO)) \ \wedge\ t'\widetilde{m_2}(c)\subseteq(dom(\widetilde{m_2}(DO))\backslash INF(DO)), \quad (7)$$

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 (7), $(t',t)\in(\widetilde{m_2}(c)*\widetilde{m_2}(DO))$. From this, $B(t')$, and the definition of $\widetilde{N}2$, it follows that $(t',t)\in\widetilde{N}2(\widetilde{m_2}(DO))$, and hence, since $\widetilde{m_2}(DO)$ is a fixpoint of $\widetilde{N}2$, $(t',t)\in\widetilde{m_2}(DO)$ (we use direction $\widetilde{m_2}(DO)\supseteq\widetilde{N}2(\widetilde{m_2}(DO))$ 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 (7). 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 $\widetilde{N}2$ 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 $(s_0,s)\in\widetilde{m_2}(DO)\subseteq\widetilde{N}2(\widetilde{m_2}(DO))$. 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 $\widetilde{m_2}(DO)\subseteq\widetilde{N}2(\widetilde{m_2}(DO))$, there are again two cases to be considered for s_1. Because, 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

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

Base $j{=}0$: We know that $(s_0, s){\in}\widetilde{m_2}(DO){=}\widetilde{N}2(\widetilde{m_2}(DO))$. 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 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 (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 2.2($\Leftarrow$), 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 2.2($\Rightarrow$), 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:[1]

- 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 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

[1] It is unique up to isomorphism.

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 see that $(t, v)\in\widetilde{N2}(\widetilde{m_2}(DO))$, and hence $(t, v)\in\widetilde{m_2}(DO)$ by the fixpoint equation ($\supseteq$).
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. ■ 3.3
Proof of lemma 3.5:
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. ■ 3.5
More precisely, let $\{X_j \mid j\in J\}$ be any set of subsets of $2^{Z_\perp}$ (with index set J). Then

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

are the greatest lower and least upper bounds, respectively, of $\{X_j \mid j\in I\}$.
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 $\underbrace{a(\mu \diamond \rho_1)\backslash\{\perp\} \subseteq a(\mu \diamond \rho_2)}_{(1)}$ and $\underbrace{\perp \in a(\mu \diamond \rho_1) \vee \perp \notin a(\mu \diamond \rho_2)}_{(2)}$.

$$
\begin{array}{lrl}
\text{Ad (1):} & b \in a(\mu \diamond \rho_1)\backslash\{\perp\} & \Rightarrow \text{(definition of } \diamond) \\
 & & \quad b \in a(\mu \circ \rho_1)\backslash\{\perp\} \\
 & & \Rightarrow \text{(definition of } \circ) \\
 & & \quad \exists x\colon x \in a\mu \wedge b \in x\rho_1\backslash\{\perp\} \\
 & & \Rightarrow (\rho_1 \sqsubseteq \rho_2) \\
 & & \quad x \in a\mu \wedge \exists x\colon b \in x\rho_2\backslash\{\perp\} \\
 & & \Rightarrow \text{(definition of } \circ) \\
 & & \quad b \in a(\mu \circ \rho_2)\backslash\{\perp\} \\
 & & \Rightarrow \text{(definition of } \diamond) \\
 & & \quad b \in a(\mu \diamond \rho_2)\backslash\{\perp\}. \\
\text{Ad (2):} & \perp \in a(\mu \diamond \rho_2) & \Rightarrow \text{(definition of } \diamond) \\
 & & \quad (a,\perp)\in\mu \vee \exists x\colon (a,x)\in\mu \wedge (x,\perp)\in\rho_2 \\
 & & \Rightarrow (\rho_1 \sqsubseteq \rho_2) \\
 & & \quad (a,\perp)\in\mu \vee \exists x\colon (a,x)\in\mu \wedge (x,\perp)\in\rho_2 \wedge (x,\perp)\in\rho_1 \\
 & & \Rightarrow \text{(definition of } \circ) \\
 & & \quad (a,\perp)\in\mu \vee (a,\perp)\in\mu \circ \rho_1 \\
 & & \Rightarrow \text{(definition of } \diamond \text{ and } \mu \circ \rho_1 \subseteq \mu \diamond \rho_1) \\
 & & \quad \perp \in a(\mu \diamond \rho_2).
\end{array}
$$

Proof of Prop. 3.6:
We only need to prove $\widetilde{m}(DO) = m(DO)$ for $DO = \mathbf{do}\ B \to c\ \mathbf{od}$, assuming as an induction hypothesis that $\widetilde{m}(c) = m(c)$.
Proof of $\widetilde{m}(DO) \sqsubseteq m(DO)$:
The smallest fixpoint of N can be expressed as follows:

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

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

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

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

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

Ad (1): Let $s \in s'm(DO)\backslash\{\perp\}$. 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 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 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 $\perp \in s'\widetilde{m}(DO)$. We prove $\perp \in s'm(DO)$. Using $\perp \in s'\widetilde{m}(DO)$, the fixpoint equation $\widetilde{m}(DO) = N(\widetilde{m}(DO))$ 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 $(s', \perp) \in m(DO)$. Otherwise, the construction continues indefinitely and then we have (C) and hence, again, $(s', \perp) \in m(DO)$.
In both cases, $\perp \in s'm(DO)$, which proves (2).
This ends the proof of $\widetilde{m}(DO) \sqsupseteq'_{EM} m(DO)$.
The main result now follows from the antisymmetry of $\sqsubseteq$. ■ 3.6

[2] One may be tempted to use the same idea in order to simplify the first part of the proof of proposition 3.3. However, this does not seem to work as nicely as it does here.