University of Groningen

Interpretations of Recursion under Unbounded Nondeterminacy

Hesselink, Wim H.

Published in:
Theoretical Computer Science

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
1988

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Hesselink, W. H. (1988). Interpretations of Recursion under Unbounded Nondeterminacy. *Theoretical Computer Science*, *59*(3), 211-234.

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 14-02-2023

Theoretical Computer Science 59 (1988) 211-234
North-Holland

INTERPRETATIONS OF RECURSION UNDER UNBOUNDED NONDETERMINACY

Wim H. HESSELINK

Department of Mathematics and Computer Science, Groningen University, 9700 AV Groningen, The Netherlands

Communicated by J. de Bakker
Received June 1986
Revised July 1987

Abstract. A language is constructed that supports arbitrary atomic statements, composition, alternatives, and mutual recursion in the presence of unbounded nondeterminacy. The concept of interpretation is defined axiomatically. By operational means a standard interpretation is constructed, which is proved to be the smallest interpretation with respect to the Egli-Milner ordering. The corresponding weakest precondition and weakest liberal precondition are characterized as smallest and largest preparator functions.

We characterize the interpretations which upon divergence cannot deliver unjustified meaningful values. These interpretations form a spectrum that ranges from standard semantics, over various forms of fair semantics, to the so-called friendly semantics where error is signalled only if divergence can become inevitable after finitely many steps. We determine the corresponding preparator functions.

0. Introduction

0.0. Unbounded nondeterminacy

In programming, data structures and control structures are brought together. Data structures consist of sets and operations on the elements of the sets. Control structures are language constructs that prescribe which sequences of operations are to be performed.

The avoidance of unnecessary or premature choices in the specification of a data structure leads to nondeterminate specifications. If the data structure is unbounded, the nondeterminacy can also be unbounded. Therefore, the theory of control structures must be able to deal with unbounded nondeterminacy.

This kind of nondeterminacy is called loose, cf. [19, p. 514]. It is a property of a specification, which does not exclude deterministic implementations. Thus we see no reason to restrict to countable nondeterminacy. This has the advantage that some technical problems of [2] disappear.

0.1. The operational abstraction of a data structure

From the point of view of the control structure, the number of data values and their types are irrelevant. So we put all data values together in one value, which is called the state. The set of the conceivable states is called the state space X. It may be thought of as the cartesian product of the relevant data domains.

An operation f is a way to establish a certain postcondition, which usually depends on the initial state. If the initial state is called x, we use $f(x)$ to denote the set of the states which satisfy the postcondition. It may be that the operation f is permitted to result in divergence. Therefore, we introduce an error symbol ψ that is not element of X, and we prescribe that $f(x)$ is a nonempty subset of $X \cup \{\psi\}$.

Remark. It is not satisfactory to model divergence by means of an empty result set $f(x)$. For, in that case, a nondeterminate choice between divergence and a meaningful result cannot be specified. The same argument can be used against Hehner's proposal (cf. [10]) to model divergence by means of a result set $f(x) = X$. The condition that $f(x)$ be nonempty, is Dijkstra's law of the excluded miracle, cf. [6]. We see no need to drop this law as is done in [17].

0.2. Unification of alternative command and routine call

The control structure needs tests on the state in order to choose between alternative actions. Some authors treat tests as operations that in case of a negative answer deliver an empty set of results, cf. [4, Theorem 7.8(b)]. This may lead to the occurrence of blind alleys so that the semantics require a kind of backtracking mechanism, cf. [13, p. 470].

Instead of this unification of tests and statements, we have chosen a unification of Dijkstra's alternative command (cf. [6]) and the routine call. The declaration of a routine specifies a set of alternative refinements each guarded by its own test. If a routine is called, it is replaced nondeterminately by one of the refinements with a test that succeeds. The formalism enforces the existence of at least one such test. In this way no backtracking is required.

0.3. Interpretations

The semantics of a language is determined by the interpretation. The concept of interpretation is defined axiomatically in Section 1. Recursion and loops lead to freedom of interpretation. Typical cases are the following loops, where initially the integer variable x is positive.

(A) **do** $x \neq 0 \rightarrow x := x - 1$ ▯ $x \neq 0 \rightarrow$ skip **od**,

(B) **do** $x \neq 0 \rightarrow x := x - 1$ ▯ $x \neq 0 \rightarrow x := x + 1$ **od**,

(C) **do** $x \neq 0 \rightarrow x := x - 1$ ▯ $x \neq 0 \rightarrow x := x - 2$ **od**.

In case (A), the standard semantics gives the possibility of divergence, but a fair treatment of the alternatives in the loop would guarantee convergence. In case (B), fairness is not sufficient for convergence, but the loop converges under friendly semantics. In case (C), even friendly semantics admits a possible divergence since divergence can become inevitable after finitely many steps.

0.4. Standard semantics

In Section 2, the standard interpretation is constructed by means of an operational model, cf. [2]. We prove in Section 3 that this standard interpretation is the smallest interpretation with respect to the Egli–Milner ordering, cf. [2, 20]. In Section 4, we introduce preparator functions as a generalization of the weakest precondition wp and the weakest liberal precondition wlp, cf. [6]. It is proved that wp and wlp are the two extreme preparator functions, cf. [7].

In these three sections the results may not be new, but proofs of this generality seem to have not yet appeared in the literature. Our principal goal has been to provide complete proofs in a most general and elementary setting.

0.5. Operationally justified interpretations

The axiomatic definition in Section 1 admits interpretations which upon divergence may deliver meaningful but erroneous results. For example, it is not excluded that after an initialization $x := 3$, the loop

$$\textbf{do}\ x \neq 0 \rightarrow x := 1\ \textbf{od}$$

is interpreted to deliver a state with $x = 0$, or $x = 1$, or $x = 2$. Such an interpretation is considered to be not operationally justified. The concept of operational justification is defined in Section 5. The operationally justified interpretations are constructed explicitly in terms of sets of potential divergence. They form a spectrum that ranges from standard semantics, over various forms of fair semantics, up to friendly semantics (compare Section 0.3). It turns out that the operationally justified interpretations are in a one-to-one correspondence with a nice class of preparator functions.

0.6. Semantics of fair termination

In Section 6 we show that the semantics of termination for all fair execution sequences is an operationally justified interpretation. More specifically, we introduce T-fairness, where T is a finite family of transition types that are to be treated fairly. The T-fair interpretation signals divergence in the case of an infinite, T-fair calculation or of a finite calculation that ends in a diverging statement.

0.7. Elementary tools

Our main tool is the elementary calculus of set-valued functions. Several ingredients of other presentations are missing. We do not use adjoint functors, cpo's, or complete metric spaces, cf. [0, 2, 14]. We do not need transfinite induction or König's Lemma. The recursion is modeled by means of refinements, so we do not need substitutions or a block structure.

0.8. Additional remarks

We got interested in this field by reading [7]. Our interest was revived when investigating observable equivalence in nondeterministic data types, cf. [11, Theorem 3.5]. The results on standard semantics were first obtained by means of the metric space of infinite sequences, cf. [14]. Later we realized that configurations are more convenient, cf. [2]. Friendly semantics was invented as a classroom example. The description of the operationally justified interpretations by means of sets of potential divergence emerged when the first version was almost completed. It turned out, however, that that section contained a grave error. This was pointed out by one of the referees. In the meantime we had learned to appreciate Dijkstra's rigor of simple and explicit proofs. Therefore, the revision became an exercise in balancing between almost conflicting styles. Especially, Section 5 was completely rewritten.

1. A language and its interpretations

1.0. Operations on the state space X

Let X be an arbitrary set, not necessarily finite. The set X is called the *state space.* The elements of X are called *states.* We form the disjoint union

$$X_+ = X \cup \{\psi\}$$

where ψ is a formal symbol, not in X. An *operation* f on X is defined to be a function which assigns to every state x in X a nonempty subset $f(x)$ of X_+. We let $\mathrm{Op}(X)$ denote the set of all operations f on X. As examples of operations we provide skip, abort, and havoc, given by

$$\mathrm{skip}(x) = \{x\}, \qquad \mathrm{abort}(x) = \{\psi\}, \qquad \mathrm{havoc}(x) = X.$$

Remarks. We use the symbol ψ instead of the conventional bottom in order to indicate that X_+ is not regarded as a flat cpo, cf. [2, p. 481]. We will use $\mathcal{P}(U)$ to denote the powerset of a set U. So an operation f on X is a function $f : X \to \mathcal{P}(X_+)$ with nonempty values $f(x)$. The condition that the sets $f(x)$ be nonempty, is Dijkstra's law of the excluded miracle, cf. [6]. As argued in [3, 20], the functional approach to nondeterminacy is equivalent to the relational approaches, cf. [7, 10, 17, 19]. We use the functional approach for personal convenience and for consistency with [11].

1.1. Composition of operations

If $f \in \mathrm{Op}(X)$, then we define $f(\psi) = \{\psi\}$. If V is a subset of X_+, we define $f_*(V) = \bigcup_{x \in V} f(x)$. The composition $f \circ g$ of operations f and g is defined by $f \circ g(x) = f_*(g(x))$.

It is easy to see that composition of operations is associative, and that skip is the neutral element for the composition, cf. [2, Proposition 2.3].

1.2. Statements

We assume that S is a given alphabet, not necessarily finite. The elements of S are called *statement symbols.* The semantics is defined by a fixed function

$$k : S \to \mathrm{Op}(X).$$

If $s \in S$, then the operation $k(s)$ is to be regarded as the meaning of the statement symbol s.

1.3. Routine symbols, strings, and refinements

Let H be a second alphabet. The elements of H are called *routine symbols*. Let A denote the disjoint union of the alphabets S and H. Let A^* denote the set of the strings of elements of A. The empty string is denoted as ε. Concatenation of strings is denoted by means of a semicolon. A *refinement function* is defined to be a set-valued function

$$R : H \times X \to \mathcal{P}(A^*)$$

such that the set $R(h, x)$ is nonempty for any $h \in H$ and $x \in X$. The strings $r \in R(h, x)$ are called the *refinements* of the routine symbol h that are *acceptable* to the state x.

1.4. Interpretations

Let a refinement function R be fixed. We define an *interpretation* to be a function $p : A^* \to \mathrm{Op}(X)$ that satisfies the following conditions:

(a) $p(\varepsilon) = \mathrm{skip}$;

(b) if $t, u \in A^*$, then $p(t; u) = p(u) \circ p(t)$;

(c) if $s \in S$, then $p(s) = k(s)$;

(d) if $h \in H$ and $x \in X$, then $p(h)(x) = \bigcup_{r \in R(h,x)} p(r)(x)$.

If p is an interpretation and $y \in p(t)(x)$, we say that p signals the possibility of result y for an execution of string t that starts in state x. In view of the conditions (a), (b), (c), an interpretation p is completely determined by the operations $p(h)$ with $h \in H$. Condition (d) is a recursive one. In general, the conditions are not strong enough to determine the operations $p(h)$ uniquely. In the next section we give an example with four different interpretations.

Remarks. Condition (b) is not imposed by Hehner, cf. [10, Definition 1.2]. Hehner's interpretations can violate condition (b) only in case of a finite state space, but that is because of his restriction to bounded nondeterminism.

The condition in Definition 1.3 that the sets $R(h, x)$ are nonempty, is motivated by the elegance of the above condition (d). In Dijkstra's alternative command (cf. [6]) the absence of a test that succeeds leads to abortion. In our setting this effect can be accomplished by putting $R(h, x) = \{s0\}$ where $s0$ is a statement symbol with $k(s0) = \mathrm{abort}$.

1.5. Example of a refinement function with four interpretations

In an imperative language with nondeterminacy we consider a boolean variable x and the loop

$$\textbf{do } x \to \text{choose } x \textbf{ od}.$$

In order to translate the loop into our language we introduce $X = \text{boolean}$ with the elements ff and tt. We put $S = \{s\}$ with $k(s) = \text{havoc}$. The loop is represented by the routine symbol h with

$$R(h, \mathrm{ff}) = \{\varepsilon\} \quad \text{and} \quad R(h, \mathrm{tt}) = \{(s; h)\}.$$

Put $H = \{h\}$. All interpretations p are easily determined. It suffices to determine the operation $p(h)$. It is clear that

$$p(h)(\mathrm{ff}) = p(\varepsilon)(\mathrm{ff}) = \{\mathrm{ff}\}.$$

The other case leads to the recursive equation

$$p(h)(\mathrm{tt}) = p(h)(\mathrm{ff}) \cup p(h)(\mathrm{tt}).$$

Therefore, the set $p(h)(\mathrm{tt})$ can be any subset of X_+ that contains the element ff. So it may contain tt, or ψ, or neither one, or both together. This shows that there are four different interpretations in this case.

Remark. One may argue that an interpretation p with $\mathrm{tt} \in p(h)(\mathrm{tt})$ is not justified. This suggests a definition, cf. Definition 5.1 below.

2. Operational semantics: the configuration graph

2.0. The operational construction

The operational semantics is constructed by means of a directed graph of configurations, cf. [2, p. 486]. A configuration consists of a state and a string of statement and routine symbols. The transitions in the graph correspond to the execution of a statement symbol or the choice of a refinement of a routine symbol. The main result of this section is that the semantic function thus constructed is an interpretation in the sense of Definition 1.4.

We have two reasons for considering operational semantics. Although the specification of semantics had better be axiomatic or denotational (cf. Definition 1.4, Theorem 3.3, Definition 4.3 and Theorem 4.3), implementations are likely to be operational. An abstract operational model seems to be necessary to verify the correctness of operational implementations. A second reason is that the operational semantics yields an existence proof, which is more simple than the standard methods associated with denotational semantics.

2.1. The configuration graph

A *configuration* is defined to be a pair $\langle x, t\rangle$ with $x \in X_+$ and $t \in A^*$. So the set of configurations is the cartesian product $Z = X_+ \times A^*$. A configuration is called *terminal* if it is of the form $\langle x, \varepsilon\rangle$ with $x \in X$. A terminal configuration $\langle x, \varepsilon\rangle$ will be identified with the state x. In this way, the set X is identified with a subset of Z. Configurations of the form $\langle \psi, t\rangle$ with $t \in A^*$ are called *aborted*. We let $Z0$ denote the set of the aborted configurations.

The set Z is made into a directed graph by specifying, for each configuration $y=\langle x, t\rangle$, a set of transitions $y \to z$. If y is aborted or terminal (i.e., if $x=\psi$ or $t=\varepsilon$), then y has no transitions. Otherwise, it holds that $x \in X$ and $t=(a;u)$ with $a \in A$ and $u \in A^*$. Then we distinguish two cases:

Case (a) If $a \in S$ and $x \neq \psi$, the configuration $\langle x, a;u\rangle$ has the transitions $\langle x, a;u\rangle \to \langle x0, u\rangle$ with $x0 \in k(a)(x)$.

Case (b) If $a \in H$ and $x \neq \psi$, the configuration $\langle x, a;u\rangle$ has the transitions $\langle x, a;u\rangle \to \langle x, r;u\rangle$ with $r \in R(a, x)$.

The set Z together with the binary relation "$\to$" is called the configuration graph Z.

2.2. *The postulate of operational semantics*

Let $\xrightarrow{*}$ denote the reflexive and transitive closure of the transition relation $\to$ on the configuration graph Z. Let us write $y \xrightarrow{*} \infty$ to indicate that $y \xrightarrow{*} \langle \psi, t\rangle$ for some aborted configuration $\langle \psi, t\rangle$ or that y is the starting point of some infinite path in the graph Z. We define the semantic function $m: Z \to \mathcal{P}(X_+)$ by

$$m(y)=\{x \in X \mid y \xrightarrow{*} x\} \cup \{\psi \mid y \xrightarrow{*} \infty\}.$$

If a path in Z cannot be extended, it ends in a configuration which is either terminal or aborted. Therefore, the set $m(y)$ is always nonempty. If $y=\langle \psi, t\rangle$ then $m(y)=\{\psi\}$. It follows that m can be identified with a function $m: A^* \to \mathrm{Op}(X)$ such that $m(t)(x)=m(\langle x, t\rangle)$ holds for every configuration $\langle x, t\rangle$. Below in Theorem 2.4 we will prove that the function m is an interpretation. After that it will be called the standard interpretation.

2.3. *Example*

The relevant part of the configuration graph of Example 1.5 consists of four configurations with four possible transitions:

$$\langle \mathrm{tt}, h\rangle \leftrightarrows \langle \mathrm{tt}, s;h\rangle \longrightarrow \langle \mathrm{ff}, h\rangle \longrightarrow \langle \mathrm{ff}, \varepsilon\rangle.$$

It follows that $m(h)(\mathrm{tt})=\{\mathrm{ff}, \psi\}$.

2.4.

The configuration graph Z has the following regularity property.

Lemma 2.1. (a) *If $\langle x, t\rangle \to \langle x0, u\rangle$ is a transition and $v \in A^*$, then $\langle x, t;v\rangle \to \langle x0, u;v\rangle$ is a transition.*

(b) *If $\langle x, t;v\rangle \to \langle x0, w\rangle$ is a transition, then $x \in X$, and either $t=\varepsilon$ or w has a decomposition $w=(u;v)$ such that $\langle x, t\rangle \to \langle x0, u\rangle$ is a transition.*

Proof. For configurations $\langle x, t\rangle$ and $\langle x0, w\rangle$, Definition 2.1 implies that

$$
\begin{aligned}
(*)\quad & \langle x, t\rangle \rightarrow \langle x0, w\rangle \\
&\equiv (\exists a \in A, p \in A^*: t = a;p \\
&\qquad \wedge ((a \in S \wedge x0 \in k(a)(x) \wedge w = p) \\
&\qquad\quad \vee (a \in H \wedge x0 = x \wedge (\exists r \in R(a, x): w = r;p)))) \\
&\equiv (\exists a \in A, p, r \in A^*: t = a;p \wedge w = r;p \wedge \langle x, a\rangle \rightarrow \langle x0, r\rangle).
\end{aligned}
$$

For $t \neq \varepsilon$, it follows that

$$
\begin{aligned}
& \langle x, t;v\rangle \rightarrow \langle x0, w\rangle \\
&\equiv (\exists a \in A, p, r \in A^*: t;v = a;p \wedge w = r;p \wedge \langle x, a\rangle \rightarrow \langle x0, r\rangle) \quad [\text{by } (*)] \\
&\equiv (\exists a \in A, q, r \in A^*: t = a;q \wedge w = r;q;v \wedge \langle x, a\rangle \rightarrow \langle x0, r\rangle) \\
&\hfill [\text{use } p = q;v] \\
&\equiv (\exists u \in A^*: w = u;v \wedge \langle x, t\rangle \rightarrow \langle x0, u\rangle) \qquad [\text{use } (*) \text{ and } u = r;q].
\end{aligned}
$$

Since, in general, $\langle x, t\rangle$ only has transitions if $x \in X$ and $t \neq \varepsilon$, the assertions (a) and (b) follow. □

2.5. *Ingredients of the composition*

If $y = \langle x, t\rangle$ is a configuration and $v \in A^*$ is a string, we define the postfixed configuration

$$y;v = \langle x, t;v\rangle.$$

Note that the identification of x in X with $\langle x, \varepsilon\rangle$ in Z leads to the equality $x;t = \langle x, t\rangle$ for any $x \in X$ and $t \in A^*$. The next result describes the paths that start in a postfixed configuration $y;v$. It is a kind of generalization of Lemma 2.1.

Proposition 2.2. *For configurations $y, z \in Z$ and a string $v \in A^*$ we have*

(a) $(y;v \xrightarrow{*} z) \equiv (\exists y0 \in Z: y \xrightarrow{*} y0 \wedge y0;v = z) \vee (\exists x \in X: y \xrightarrow{*} x \wedge \langle x, v\rangle \xrightarrow{*} z)$,

(b) $(y;v \xrightarrow{*} \infty) \equiv (y \xrightarrow{*} \infty) \vee (\exists x \in X: y \xrightarrow{*} x \wedge \langle x, v\rangle \xrightarrow{*} \infty)$.

Proof. In both cases the implication "$\Leftarrow$" is an easy consequence of Lemma 2.1(a) together with the transitivity of the relation "$\xrightarrow{*}$" on Z.

As for the implication "$\Rightarrow$", assume that $y;v \xrightarrow{*} z$, or that $y;v \xrightarrow{*} \infty$ respectively. Then there is a finite or infinite path $(z_i \mid i \in I)$ starting in $z_0 = (y;v)$ with $z_{i-1} \rightarrow z_i$ for all indices $i \in I \backslash \{0\}$. In case (a) the path ends in the configuration $z = z_m$, say, so that $I = \{i \mid 0 \leqslant i \leqslant m\}$. In case (b) the path ends in an aborted state $z_m = \langle \psi, t\rangle$, or the path is infinite so that $I = \mathbb{N}$. Write $y = \langle x, t_0\rangle$ and $z_i = \langle x_i, w_i\rangle$ with $x, x_i \in X_+$ and $t_0, w_i \in A^*$. Since $z_0 = (y;v)$, we have $x = x_0$ and $w_0 = (t_0;v)$. If $w_{i-1} = (t_{i-1};v)$, then Lemma 2.1(b) implies $x_{i-1} \in X$, and $t_{i-1} = \varepsilon$ or $w_i = (t_i;v)$ with $\langle x_{i-1}, t_{i-1}\rangle \rightarrow \langle x_i, t_i\rangle$. As long as the second alternative applies, induction yields a sequence of configurations $y_i = \langle x_i, t_i\rangle$ with $y_0 = y$, and $y_{i-1} \rightarrow y_i$, and $z_i = (y_i;v)$. If this construction covers the entire range I, we have obtained the first disjunct of the right-hand side of (a) or (b). Otherwise, the construction stops at some index $r \in I$ with $x = x_{r-1} \in X$ and $t_{r-1} = \varepsilon$, and hence $y \xrightarrow{*} x$. In that case, we have $\langle x, v\rangle = z_{r-1}$, and therefore, $\langle x, v\rangle \xrightarrow{*} z$ or $\langle x, v\rangle \xrightarrow{*} \infty$ respectively so that the second disjunct of the right-hand side of (a) or (b) holds. □

Corollary 2.3. *For $y \in Z$, and $x0 \in X$, and $v \in A^*$, it holds that*

$$(y; v \xrightarrow{*} x0) \;\equiv\; (\exists x \in X: y \xrightarrow{*} x \wedge \langle x, v\rangle \xrightarrow{*} x0).$$

Proof. This follows from Proposition 2.2(a) with $z = x0$, in which case the first disjunct of Proposition 2.2(a) implies the second one:

$$\begin{aligned} &(\exists y0 \in Z: y \xrightarrow{*} y0 \wedge y0; v = x0) \\ &\equiv\; v = \varepsilon \wedge y \xrightarrow{*} x0 && [\text{because } x0 \in X] \\ &\Rightarrow (\exists x \in X: y \xrightarrow{*} x \wedge \langle x, v\rangle \xrightarrow{*} x0) && [\text{choose } x = x0]. \end{aligned}$$ □

Theorem 2.4. *The semantic function $m: A^* \to \mathrm{Op}(X)$ is an interpretation.*

Proof. We verify the conditions of Definition 1.4. Let $x \in X$ be given.

(a) The configuration $\langle x, \varepsilon\rangle$ belongs to X. It has no transitions, and it does not satisfy $\langle x, \varepsilon\rangle \xrightarrow{*} \infty$. Therefore, we have $m(\varepsilon)(x) = m(\langle x, \varepsilon\rangle) = \{x\}$. This proves that $m(\varepsilon) = \mathrm{skip}$.

(b) For $t, v \in A^*$, we observe

$$\begin{aligned} &m(t; v)(x) \\ &= \{x0 \in X \mid \langle x, t; v\rangle \xrightarrow{*} x0\} \cup \{\psi \mid \langle x, t; v\rangle \xrightarrow{*} \infty\} && [\text{by Def. 2.2}] \\ &= \{x0 \in X \mid \exists x1 \in X: \langle x, t\rangle \xrightarrow{*} x1 \wedge \langle x1, v\rangle \xrightarrow{*} x0\} && [\text{by Cor. 2.3}] \\ &\quad \cup \{\psi \mid \exists x1 \in X: \langle x, t\rangle \xrightarrow{*} x1 \wedge \langle x1, v\rangle \xrightarrow{*} \infty\} \cup \{\psi \mid \langle x, t\rangle \xrightarrow{*} \infty\} \\ &&& [\text{by Prop. 2.2(b)}] \\ &= (m(t)(x) \cap \{\psi\}) \cup \bigcup_{z \in X \cap m(t)(x)} m(v)(z) && [\text{by Def. 2.2}] \\ &= m(v) \circ m(t)(x) && [\text{by Def. 1.1}]. \end{aligned}$$

(c) For any statement symbol s, we have

$$\begin{aligned} &m(s)(x) \\ &= \{x0 \in X \mid \langle x, s\rangle \xrightarrow{*} x0\} \cup \{\psi \mid \langle x, s\rangle \xrightarrow{*} \infty\} && [\text{by Def. 2.2}] \\ &= \{x0 \in X \mid x0 \in k(s)(x)\} \cup \{\psi \mid \psi \in k(s)(x)\} && [\text{by Def. 2.1}] \\ &= k(s)(x). \end{aligned}$$

(d) For any routine symbol h, we have

$$\begin{aligned} &m(h)(x) \\ &= \{x0 \in X \mid \langle x, h\rangle \xrightarrow{*} x0\} \cup \{\psi \mid \langle x, h\rangle \xrightarrow{*} \infty\} && [\text{by Def. 2.2}] \\ &= \bigcup_{r \in R(h,x)} (\{x0 \in X \mid \langle x, r\rangle \xrightarrow{*} x0\} \cup \{\psi \mid \langle x, r\rangle \xrightarrow{*} \infty\}) && [\text{by Def. 2.1}] \\ &= \bigcup_{r \in R(h,x)} m(r)(x) && [\text{by Def. 2.2}]. \end{aligned}$$ □

Remark. Henceforward the semantic function m will be called the *standard interpretation.*

3. Denotational semantics without cpo's or limits

3.0.

In this section we show that the standard interpretation constructed by operational means in Section 2 is also a natural one from the point of view of denotational semantics. In fact, we prove that it is the smallest interpretation with respect to the Egli-Milner ordering, cf. [2, 20].

3.1.

Let $p:A^*\to \mathrm{Op}(X)$ be an interpretation. It can be identified with a function $p:Z\to \mathcal{P}(X_+)$ given by $p(\langle x,t\rangle)=p(t)(x)$.

Lemma 3.1. (a) *If* $y\in Z0$, *then* $p(y)=\{\psi\}$.
(b) *If* $y\in X$, *then* $p(y)=\{y\}$.
(c) *If* $y\notin X\cup Z0$, *then* $p(y)=\bigcup_{z:y\to z}p(z)$.

Proof. We write $y=\langle x,t\rangle$, so that $p(y)=p(t)(x)$. If $x=\psi$, then $p(y)=\{\psi\}$ by Definition 1.1. If $t=\varepsilon$, then $p(y)=\{y\}$ by Definition 1.4(a). By Definition 2.1, this proves the cases (a) and (b).

In case (c) we have $x\in X$ and $t\neq\varepsilon$. Write $t=(a;u)$ with $a\in A$ and $u\in A^*$. By Definitions 1.4(b) and 1.1, it holds that $p(y)=p(u)_*(p(a)(x))$. We calculate the union. If $a\in S$, then

$$\begin{aligned}&\bigcup_{z:y\to z}p(z)\\ &=\bigcup_{x0\in k(a)(x)}p(\langle x0,u\rangle)\quad&&[\text{by Def. 2.1(a)}]\\ &=\bigcup_{x0\in p(a)(x)}p(u)(x0)&&[\text{by Def. 1.4(c)}]\\ &=p(u)_*(p(a)(x))&&[\text{by Def. 1.1}].\end{aligned}$$

If $a\in H$, then

$$\begin{aligned}&\bigcup_{z:y\to z}p(z)\\ &=\bigcup_{r\in R(a,x)}p(\langle x,r;u\rangle)\quad&&[\text{by Def. 2.1(b)}]\\ &=\bigcup_{r\in R(a,x)}p(u)_*(p(r)(x))&&[\text{by Def. 1.4(b)}]\\ &=p(u)_*\Big(\bigcup_{r\in R(a,x)}p(r)(x)\Big)&&[\text{set calculus}]\\ &=p(u)_*(p(a)(x))&&[\text{by Def. 1.4(d)}].\quad\square\end{aligned}$$

3.2

Theorem 3.2. *Let y be a configuration.*

(a) *Every interpretation p satisfies $\{x \in X \mid y \xrightarrow{*} x\} \subset p(y)$.*

(b) *Let p and q be interpretations with $p(y) \neq q(y)$. Then there is an infinite path $(z_i \mid i \in \mathbb{N})$ starting in $z_0 = y$ such that $p(z_i) \neq q(z_i)$ for all indices $i \in \mathbb{N}$.*

Proof. (a): If $y \to z$, then $p(z) \subset p(y)$, by Lemma 3.1(c). Therefore, $y \xrightarrow{*} z$ implies $p(z) \subset p(y)$. By Lemma 3.1(b), it follows that $x \in p(y)$ for any $x \in X$ with $y \xrightarrow{*} x$.

(b): If $p(y) \neq q(y)$, then Lemma 3.1 implies that $y \notin X \cup Z0$, and that $p(z) \neq q(z)$ for some transition $y \to z$. This fact is used in an inductive construction. □

3.3. The Egli-Milner relation

The powerset $\mathcal{P}(X_+)$ of X_+ is equipped with the *Egli-Milner relation* $\leq$ given by

$$U \leq V \;\equiv\; (U \subset V \cup \{\psi\}) \wedge (U = V \vee \psi \in U).$$

It is a well known (but peculiar) fact that this relation is an ordering of $\mathcal{P}(X_+)$ (by ordering we always mean partial ordering). This ordering induces an ordering on the set of the interpretations by

$$p \leq q \;\equiv\; (\forall y \in Z : p(y) \leq q(y)).$$

3.4.

Theorem 3.3. *The standard interpretation is the smallest interpretation with respect to the ordering $\leq$.*

Proof. Let p be an arbitrary interpretation. Let $y \in Z$. By Theorem 3.2(a) we have $m(y) \subset p(y) \cup \{\psi\}$. By Theorems 2.4 and 3.2(b), we have $m(y) = p(y) \vee \psi \in m(y)$. This proves that $m(y) \leq p(y)$, and hence that $m \leq p$. □

3.5. Independent construction

Some readers may prefer an abstract construction of the smallest interpretation. This can be done, cf. [2, 3]. In fact, the powerset $\mathcal{P}(X_+)$ with the Egli-Milner ordering is a cpo. The set Y of the functions $p : H \times X \to \mathcal{P}(X)$ gets an induced cpo structure. The interpretations are brought in bijective correspondence with the fixed points of a certain order-preserving function $F: Y \to Y$. Using transfinite induction one shows that F has a smallest fixed point in Y. The identification of this smallest fixed point with the semantic function m of Definition 2.2 still requires the arguments of Theorems 2.4 and 3.3. If X is not countable, we have to accept the fact that countable directed subsets of these cpo's need not be eventually constant, cf. [3, Fact 2.4].

4. The preparator functions wp and wlp

4.0. Weakest preconditions

In denotational semantics the results of an execution of a string of commands are expressed in terms of the string and the initial state. For programming methodology, however, it is more important to know the weakest precondition such that execution of the string is guaranteed to establish a required postcondition. One distinguishes between partial correctness and total correctness. In the first case one speaks of the weakest liberal precondition. This condition does not guarantee termination of the string, but if termination occurs then the postcondition is established, cf. [6].

According to a suggestion of Wadsworth (cf. [20, p. 477]) the weakest precondition of a recursive procedure can be expressed as a minimal solution of a certain equation. The present section contains a formalization of this fact, and of its liberal analogue. It turns out that partial correctness and total correctness represent the two extreme solutions. In the case of the repetition, this result is due to Dijkstra and Scholten, cf. [7]. For other references we refer to the last part of [16].

4.1. *Predicate transformers: set transformers*

A predicate as may be used in programming practice, corresponds to the subset of the state space where the predicate holds. Therefore, a predicate transformer corresponds to a function that maps subsets of X to subsets of X, cf. [20]. Let $\mathcal{F}(X)$ denote the set of such functions $f: \mathcal{P}(X) \to \mathcal{P}(X)$. We order the set $\mathcal{F}(X)$ in terms of set inclusion by

$$f \subset f' \equiv (\forall V \in \mathcal{P}(X): f(V) \subset f'(V)).$$

4.2. *The weakest precondition functions*

The *weakest precondition function* wp and the *weakest liberal precondition function* wlp are defined as the functions wp, wlp: $A^* \to \mathcal{F}(X)$ given by

$$\mathrm{wp}(t)(V) = \{x \in X \mid m(t)(x) \subset V\},$$

$$\mathrm{wlp}(t)(V) = \{x \in X \mid m(t)(x) \subset V \cup \{\psi\}\}.$$

4.3. *Preparator functions*

In order to analyse the functions wp and wlp, we define preparator functions as a kind of objects dual to the interpretations. A function $w: A^* \to \mathcal{F}(X)$ is called a *preparator function* if it satisfies the following conditions:

(a) $w(\varepsilon)$ is the identity functon of $\mathcal{P}(X)$;

(b) if $t, u \in A^*$, then $w(t; u) = w(t) \circ w(u)$;

(c) if $s \in S$, then $\mathrm{wp}(s) \subset w(s) \subset \mathrm{wlp}(s)$;

(d) if $h \in H$ and $V \in \mathcal{P}(X)$, then $w(h)(V) = \{x \in X \mid \forall r \in R(h, x): x \in w(r)(V)\}$.

The ordering of the set $\mathcal{F}(X)$ induces an ordering of the set of preparator functions by

$$w \subset w' \equiv (\forall t \in A^*: w(t) \subset w'(t)).$$

4.4. An interpretation induces two preparator functions

Let $p: A^* \to \text{Op}(X)$ be an arbitrary interpretation. We define two associated functions

$$p^0, p^1 : A^* \to \mathscr{F}(X).$$

A little trick is used to avoid case analysis. If $V \in \mathscr{P}(X)$, we write $V^1 = V \cup \{\psi\}$, and $V^0 = V$. Let i be 0 or 1. The functions p^0 and p^1 are defined by

$$p^i(t)(V) = \{x \in X \mid p(t)(x) \subset V^i\}.$$

Remark. We have $\text{wp} = m^0$, and $\text{wlp} = m^1$, where m is the standard interpretation.

Proposition 4.1. *The functions p^0 and p^1 are preparator functions.*

Proof. We verify the conditions of Definition 4.3. Let $V \in \mathscr{P}(X)$.

(a) $p^i(\varepsilon)(V) = X \cap V^i = V$ [by 1.4(a)].

(b) $p^i(t;u)(V)$

$= \{x \in X \mid p(u)_*(p(t)(x)) \subset V^i\}$ [by 1.4(b) and 1.1]

$= \{x \in X \mid p(t)(x) \subset (p^i(u)(V))^i\}$ [set calculus]

$= p^i(t) \circ p^i(u)(V)$.

(c) The operations $m(s)$ and $p(s)$ are both equal to $k(s)$. It follows that $p^0(s) = \text{wp}(s)$ and that $p^1(s) = \text{wlp}(s)$. This implies that $\text{wp}(s) \subset p^i(s) \subset \text{wlp}(s)$.

(d) $p^i(h)(V)$

$= \{x \in X \mid p(h)(x) \subset V^i\}$ [definition $p^i(h)$]

$= \{x \in X \mid \forall r \in R(h,x): p(r)(x) \subset V^i\}$ [by cond. 1.4(d)]

$= \{x \in X \mid \forall r \in R(h,x): x \in p^i(r)(V)\}$ [definition $p^i(r)$]. □

4.5.

Proposition 4.2. *Let w be a preparator function. Let $t \in A^*$ with $t \neq \varepsilon$. Let $x \in X$ and $V \in \mathscr{P}(X)$.*

(a) $x \in w(t)(V) \Leftarrow (\forall x0 \in X_+, u \in A^*: (\langle x, t\rangle \to \langle x0, u\rangle \Rightarrow x0 \in w(u)(V)))$.

(b) *If $w(s) = \text{wp}(s)$ for every statement symbol $s \in S$, the implication "$\Leftarrow$" of part* (a) *is an equivalence.*

(c) $x \in w(t)(V) \Rightarrow (\forall x0 \in X_+, u \in A^*: (\langle x, t\rangle \to \langle x0, u\rangle \Rightarrow (x0 = \psi \vee x0 \in w(u)(V))))$.

Proof. Write $t = a; v$ with $a \in A$ and $v \in A^*$. We distinguish the cases $a \in H$ and $a \in S$. If $a \in H$, then

$$\begin{aligned}
&(\forall x0 \in X_+, u \in A^*: (\langle x, t\rangle \to \langle x0, u\rangle \Rightarrow x0 \in w(u)(V)))\\
&\equiv (\forall r \in R(a, x): x \in w(r; v)(V)) \quad \text{[because of Def. 2.1(b)]}\\
&\equiv x \in w(a; v)(V) \quad \text{[by Def. 4.3(b, d)].}
\end{aligned}$$

This proves (a), (b), (c) for the case that $a \in H$.

If $a \in S$, then

$$\begin{aligned}
&(\forall x0 \in X_+, u \in A^*: (\langle x, t\rangle \to \langle x0, u\rangle \Rightarrow x0 \in w(u)(V)))\\
&\equiv (\forall x0 \in k(a)(x): x0 \in w(v)(V)) \quad \text{[by Def. 2.1(a)]}\\
&\equiv x \in \mathrm{wp}(a) \circ w(v)(V) \quad \text{[by Defs. 4.2 and 4.3(b)]}\\
&\Rightarrow x \in w(a; v)(V) \quad \text{[by Def. 4.3(b, c)].}
\end{aligned}$$

This proves part (a). If $\mathrm{wp}(a) = w(a)$, the last implication is an equivalence. This proves (b). Case (c) follows from

$$\begin{aligned}
&(\forall x0 \in X_+, u \in A^*: (\langle x, t\rangle \to \langle x0, u\rangle \Rightarrow (x0 = \psi \vee x0 \in w(u)(V))))\\
&\equiv (\forall x0 \in k(a)(x): x0 = \psi \vee x0 \in w(v)(V)) \quad \text{[by Def. 2.1(a)]}\\
&\equiv x \in \mathrm{wlp}(a) \circ w(v)(V) \quad \text{[by Defs. 4.2 and 4.3(b)]}\\
&\Leftarrow x \in w(a; v)(V) \quad \text{[by Def. 4.3(b, c)].} \quad \Box
\end{aligned}$$

Remark. The isolation of Proposition 4.2(a) out of the proof of Theorem 4.3 below is due to one of the referees.

4.6.

Theorem 4.3. *The function* wp *is the smallest preparator function.*

Proof. By Proposition 4.1, the function wp is a preparator function. Let w be an arbitrary preparator function. Let $t \in A^*$ and $V \in \mathcal{P}(X)$ be given. We have to show that $\mathrm{wp}(t)(V) \subset w(t)(V)$. Assume the existence of a state

$$x \in \mathrm{wp}(t)(V) \setminus w(t)(V).$$

We use the configuration $\langle x, t\rangle$ as a starting point for an infinite path in the configuration graph Z. Since $\mathrm{wp}(t) \neq w(t)$, we have $t \neq \varepsilon$. Therefore, by Proposition 4.2(a), there is a transition $\langle x, t\rangle \to \langle x0, u\rangle$ with $x0 \notin w(u)(V)$. By Proposition 4.2(b) we have $x0 \in \mathrm{wp}(u)(V)$. This proves

$$x0 \in \mathrm{wp}(u)(V) \setminus w(u)(V).$$

By induction, this gives us an infinite path starting in $\langle x, t\rangle$. It follows that $\psi \in m(t)(x)$, so that $x \notin \mathrm{wp}(t)(V)$, a contradiction. This proves that $\mathrm{wp}(t)(V) \subset w(t)(V)$. $\Box$

4.7.

Theorem 4.4. *The function* wlp *is the largest preparator function.*

Proof. It is a preparator function, cf. Definition 4.4. Let w be an arbitrary preparator function. Let $t \in A^*$, and $V \in \mathcal{P}(X)$, and $x \in w(t)(V)$. Consider a result $x0 \in m(t)(x)$ that differs from ψ. Then there is a path $\langle x, t\rangle \xrightarrow{*} \langle x0, \varepsilon\rangle$ in the graph Z. This path does not contain configurations of the form $\langle \psi, u\rangle$. By inductive application of Proposition 4.2(c) we get $x0 \in w(\varepsilon)(V) = V$. This proves that $x \in \mathrm{wlp}(t)(V)$, and hence $w(t)(V) \subset \mathrm{wlp}(t)(V)$. □

4.8. Remarks

Using Proposition 4.1 one can show that Theorem 3.3 is a formal consequence of Theorems 2.4, 4.3 and 4.4 together. Conversely, however, the result in Theorem 3.3 seems to be not strong enough to give direct proofs of Theorems 4.3 and 4.4.

For our purposes it is irrelevant whether or not the preparator functions w satisfy the condition

$$w(t)\left(\bigcap_{i \in I} V_i\right) = \bigcap_{i \in I} w(t)(V_i)$$

for every nonempty family of subsets V_i of the state space X. This condition corresponds to unbounded conjunctivity of predicate transformers, cf. [8]. The condition holds if w is p^0 or p^1 for some interpretation p. There exist however, preparator functions w such that $w(t)$ does not even preserve the inclusion.

5. Operationally justified interpretations

5.0.

The methods developed for the analysis of standard semantics turn out to be sufficiently powerful to determine all interpretations of an interesting class. We will define an interpretation to be operationally justified if every result that is signalled by the interpretation, is also signalled by the standard interpretation. Since the freedom of interpretation is completely due to infinite calculations (cf. Theorem 3.2) the only remaining freedom for an operationally justified interpretation is to choose the infinite calculations that are interpreted as divergence. This choice is formalized in the concept of sets of potential divergence. As an answer to a question of a referee, we characterize a class of preparator functions, which are in perfect duality with the operationally justified interpretations.

5.1. Operational justification

Let an interpretation $p : A^* \to \mathrm{Op}(X)$ be called *operationally justified* if any result allowed by p is also allowed by the standard interpretation m, that is

$$\forall t \in A^*, x \in X\colon \quad p(t)(x) \subset m(t)(x).$$

Example. In the case of Example 1.5, the interpretation p is operationally justified if and only if $\mathrm{tt} \notin p(h)(\mathrm{tt})$.

Remark. Specification oriented approaches (cf. [10, 18]) suggest a preference for the largest interpretation with respect to the inclusion order. The existence of a unique largest interpretation is an easy consequence of the Knaster-Tarski theorem. Usually, this interpretation is not operationally justified.

5.2.
Proposition 5.1. *For an interpretation p the following conditions are equivalent:*
(a) *p is operationally justified.*
(b) $\forall t \in A^*, x \in X: p(t)(x) \cap X = m(t)(x) \cap X$.
(c) *The preparator function p^1 is equal to* wlp.

Proof. We have the following equivalences:

(a) $(\forall t \in A^*, x \in X: p(t)(x) \subset m(t)(x))$

$\equiv (\forall t \in A^*, x \in X: p(t)(x) \cap X \subset m(t)(x))$
[by Theorem 3.2(b) and Def. 2.2]

(b) $\equiv (\forall t \in A^*, x \in X: p(t)(x) \cap X = m(t)(x) \cap X)$
[by Theorem 3.2(a) and Def. 2.2]

$\equiv (\forall t \in A^*, x \in X, V \in \mathcal{P}(X): p(t)(x) \subset V \cup \{\psi\} \equiv m(t)(x) \subset V \cup \{\psi\})$

$\equiv (\forall t \in A^*, V \in \mathcal{P}(X): p^1(t)(V) = \mathrm{wlp}(t)(V))$ [by Prop. 4.1]

(c) $\equiv p^1 = \mathrm{wlp}$. $\square$

5.3. Sets of potential divergence, and their semantic functions

Recall that $Z0$ is the set of the aborted configurations $\langle \psi, t \rangle$ with $t \in A^*$ (cf. Definition 2.1). We define the set of *inevitable divergence*

$$Z1 = \{y \in Z \mid \forall z \in Z: y \xrightarrow{*} z \Rightarrow z \notin X\}.$$

In other words, we have $y \in Z1$ if and only if no path from y leads to a terminal configuration. Clearly, $Z0$ is contained in $Z1$. We define a subset D of Z to be a *set of potential divergence* if it satisfies the following conditions:
(a) $Z1 \subset D$;
(b) $\forall y \in Z: (y \in D \setminus Z0) \equiv (\exists z \in D: y \rightarrow z)$;
(c) $\forall y \in Z, v \in A^*: (y; v \in D) \equiv (y \in D) \vee (\exists x \in X: y \xrightarrow{*} x \wedge \langle x, v \rangle \in D)$.

If D is a set of potential divergence, we define the corresponding semantic function $m_D: Z \rightarrow \mathcal{P}(X_+)$ by

$$m_D(y) = \{x \in X \mid y \xrightarrow{*} x\} \cup \{\psi \mid y \in D\}.$$

By condition (a), the sets $m_D(y)$ are nonempty. If $t \in A^*$, then $m_D(\langle\psi, t\rangle) = \{\psi\}$. As in Definition 2.2 it follows that the function m_D can be regarded as a function $m_D : A^* \to \mathrm{Op}(X)$, with $m_D(t)(x) = m_D(\langle x, t\rangle)$.

5. . *Two extreme cases*

(a) *Standard semantics*: By condition 5.3(b), every set of potential divergence D is contained in the set E of all configurations $z \in Z$ with $z \xrightarrow{*} \infty$. By Proposition 2.2(b), this set E is a set of potential divergence, and hence the largest set of potential divergence. The function m_E is equal to the standard interpretation m.

(b) *Friendly semantics*: Let F be the set of configurations where divergence may become inevitable after finitely many steps. So F consists of the configurations $y \in Z$ such that $y \xrightarrow{*} z$ for some $z \in Z1$. By conditions 5.3(a, b), every set of potential divergence D contains F. Using Proposition 2.2(a) and the fact that for any $y \in Z1$ a relation $y \xrightarrow{*} z$ implies $z \in Z1$ one can prove that F is a set of potential divergence. Therefore, it is the smallest set of potential divergence. The corresponding semantic function m_F is called the *friendly semantic function*. It only signals divergence if divergence can become inevitable after finitely many steps. For examples we refer to Example 0.3.

5.5.

Theorem 5.2. *Let D be a set of potential divergence. Then the function $m_D : A^* \to \mathrm{Op}(X)$ is an operationally justified interpretation.*

Proof. We verify the conditions of Definition 1.4. Let $x \in X$ be given. By condition 5.3(b), we have $\langle x, \varepsilon\rangle \notin D$ so that $m_D(\varepsilon)(x) = \{x\}$. This proves that $m_D(\varepsilon) = \mathrm{skip}$, as required in Definition 1.4(a). For $t, v \in A^*$, we have

$$\begin{aligned}
&m_D(t;v)(x) \\
&= \{\psi \mid \langle x, t;v\rangle \in D\} \cup \{x0 \in X \mid \langle x, t;v\rangle \xrightarrow{*} x0\} && \text{[by Def. } m_D] \\
&= \{\psi \mid \langle x, t\rangle \in D\} \cup \{\psi \mid \exists x1 \in X : \langle x, t\rangle \xrightarrow{*} x1 \wedge \langle x1, v\rangle \in D\} && \text{[by cond. 5.3(c)]} \\
&\quad \cup \{x0 \in X \mid \exists x1 \in X : \langle x, t\rangle \xrightarrow{*} x1 \wedge \langle x1, v\rangle \xrightarrow{*} x0\} && \text{[by Cor. 2.3]} \\
&= (\{\psi\} \cap m_D(t)(x)) \cup \bigcup_{x1 \in X \cap m_D(t)(x)} m_D(v)(x1) && \text{[by Def. } m_D] \\
&= m_D(v) \circ m_D(t)(x), && \text{[by Def. 1.1],}
\end{aligned}$$

thus proving condition 1.4(b). In order to verify conditions 1.4(c, d), we first note that for any symbol $a \in A$ we have $\langle x, a\rangle \notin X \cup Z0$. By condition 5.3(b), it follows that

$$m_D(a)(x) = \bigcup_{z : \langle x, a\rangle \to z} m_D(z).$$

If $a \in S$, we use Definition 2.1(a) to get $m_D(a)(x) = k(a)(x)$, thus proving condition 1.4(c). If $a \in H$, it follows from Definition 2.1(b) that

$$m_D(a)(x) = \bigcup_{r \in R(a,x)} m_D(r)(x).$$

This proves condition 1.4(d), so that m_D is an interpretation. By Example 5.4(a), the set D is contained in E so that $m_D(t)(x) \subset m(t)(x)$ for all t and x. This proves that m_D is operationally justified. □

Remark. Theorem 2.4 is a special case. We have repeated the argument to show the rôle of the conditions of Definition 5.3. One may notice that condition 5.3(a) is used in 5.3 to ensure that the sets $m_D(y)$ are nonempty.

5.6. Special preparator functions

Let a preparator function w be called *special* if it satisfies the following conditions:

(a) for any string $t \in A^*$ the set $w(t)(\emptyset)$ is empty;

(b) for any statement symbol $s \in S$ we have $w(s) = \text{wp}(s)$;

(c) for any string $t \in A^*$ and any subset V of X we have $w(t)(V) = w(t)(X) \cap \text{wlp}(t)(V)$.

Condition (a) is the law of the excluded miracle (cf. [6]). Condition (c) says that total correctness is the conjunction of termination and partial correctness.

Proposition 5.3. *If p is an operationally justified interpretation, the preparator function p^0 is special.*

Proof. For $t \in A^*$ we have

$$\begin{aligned} & p^0(t)(\emptyset) \\ &= \{x \in X \mid p(t)(x) \subset \emptyset\} && \text{[by Prop. 4.1]} \\ &= \emptyset && [p(t)(x) \text{ is nonempty}]. \end{aligned}$$

For $x \in X$, and $s \in S$, and $V \subset X$ we observe

$$\begin{aligned} & x \in p^0(s)(V) \\ &\equiv p(s)(x) \subset V && [\text{definition } p^0] \\ &\equiv k(s)(x) \subset V && \text{[by cond. 1.4(c)]} \\ &\equiv x \in \text{wp}(s)(V) && \text{[by Theorem 2.4 and Def. 4.2].} \end{aligned}$$

If $t \in A^*$, then

$$\begin{aligned} & x \in p^0(t)(V) \\ &\equiv p(t)(x) \subset V \\ &\equiv p(t)(x) \subset X \wedge m(t)(x) \cap X \subset V && \text{[by Prop. 5.1(b)]} \\ &\equiv x \in p^0(t)(X) \cap \text{wlp}(t)(V) && \text{[by Prop. 4.1 and Def. 4.2].} \end{aligned}$$

This proves that p^0 is special. □

Remark. The fact that p is operationally justified is only used in the proof of part (c).

5.7.

Theorem 5.4. *Let w be a special preparator function. There is precisely one set of potential divergence D with $w = m^0_D$.*

Proof. For any set of potential divergence D, we observe

$$\begin{array}{lll} & w = m^0_D & \\ & \equiv (\forall t \in A^*\colon w(t)(X) = m^0_D(t)(X)) & \text{[by Prop. 5.3(c)]} \\ & \equiv (\forall t \in A^*, x \in X\colon x \in w(t)(X) \equiv \psi \notin m_D(t)(x)) & \text{[by Prop. 4.1]} \\ & \equiv (\forall t \in A^*, x \in X_+\colon x \notin w(t)(X) \equiv \langle x, t\rangle \in D) & \text{[by Def. 5.3]} \\ (*) & \equiv D = \{\langle x, t\rangle \in Z \mid x \notin w(t)(X)\}. & \end{array}$$

This proves uniqueness of D. It remains to prove that the set D specified in line (*) is indeed a set of potential divergence. We verify the conditions of Definition 5.3. Condition 5.3(a) is proved in

$$\begin{array}{ll} \langle x, t\rangle \in Z1 & \\ \Rightarrow m(t)(x) = \{\psi\} & \text{[by Defs. 5.3 and 2.2]} \\ \Rightarrow x \in \mathrm{wlp}(t)(\emptyset) & \text{[by Def. 4.2]} \\ \Rightarrow x \notin w(t)(X) & \text{[by conds. 5.6(a) and (c)]} \\ \equiv \langle x, t\rangle \in D & \text{[Def. of } D\text{].} \end{array}$$

Condition 5.3(b) is proved in

$$\begin{array}{ll} \langle x, t\rangle \in D\backslash Z0 & \\ \equiv x \in X\backslash w(t)(X) & \text{[Defs. } D \text{ and } Z0\text{]} \\ \equiv (\exists x0 \in X_+, u \in A^*\colon \langle x, t\rangle \xrightarrow{*} \langle x0, u\rangle \wedge x0 \notin w(u)(X)) & \\ & \text{[by cond. 5.6(b) and Prop. 4.2(b)]} \\ \equiv (\exists z \in D\colon \langle x, t\rangle \rightarrow z) & \text{[Def. of } D\text{].} \end{array}$$

Condition 5.3(c) is proved in

$$\begin{array}{ll} \langle x, t; v\rangle \in D & \\ \equiv x \notin w(t; v)(X) & \text{[Def. of } D\text{]} \\ \equiv x \notin w(t)(w(v)(X)) & \text{[by Def. 4.3(b)]} \\ \equiv x \notin w(t)(X) \cap \mathrm{wlp}(t)(w(v)(X)) & \text{[by cond. 5.6(c)]} \\ \equiv x \notin w(t)(X) \vee (\exists x0 \in X\backslash w(v)(X)\colon \langle x, t\rangle \xrightarrow{*} x0) & \text{[by Defs. 4.2 and 2.2]} \\ \equiv \langle x, t\rangle \in D \vee (\exists x0 \in X\colon \langle x, t\rangle \xrightarrow{*} x0 \wedge \langle x0, v\rangle \in D) & \text{[Def. of } D\text{].} \quad \Box \end{array}$$

5.8.

Theorem 5.5. *Let p be an operationally justified interpretation. Let w be a special preparator function. Let D be the unique set of potential divergence with $w = m^0_D$ (cf. Theorem 5.4). Then we have $w = p^0$ if and only if $p = m_D$.*

Proof. The first steps are similar to those of the uniqueness proof in 5.7. We have

$$\begin{aligned}
& w = p^0 \\
\equiv\ & (\forall t \in A^*: w(t)(X) = p^0(t)(X)) && \text{[by cond. 5.6(c)]} \\
\equiv\ & (\forall t \in A^*, x \in X: x \in w(t)(X) \equiv \psi \notin p(t)(x)) && \text{[by Prop. 4.1]} \\
\equiv\ & (\forall t \in A^*, x \in X: \langle x, t\rangle \in D \equiv \psi \in p(t)(x)) && \text{[by proof 5.7(*)]} \\
\equiv\ & (\forall t \in A^*, x \in X: m_D(t)(x) \cap \{\psi\} = p(t)(x) \cap \{\psi\}) && \text{[by Def. 5.3]} \\
\equiv\ & (\forall t \in A^*, x \in X: m_D(t)(x) = p(t)(x)) \\
& && \text{[by Prop. 5.1(b) and Theorem 5.2]} \\
\equiv\ & m_D = p. \quad \Box
\end{aligned}$$

Remark. Theorems 5.4 and 5.5 establish one-to-one correspondences between:
(i) the set of operationally justified interpretations,
(ii) the set of sets of potential divergence, and
(iii) the set of special preparator functions.
The correspondences are given by $p \mapsto p^0$, and $D \mapsto m_D$, and formula 5.7(*).

6. The semantics of fair termination

6.0.
In this section we show that the semantics of termination under fair execution, as considered for example in [1, 9, 15], can be expressed in terms of a set of potential divergence (cf. Definition 5.3). In the mentioned papers an execution sequence is said to be *fair* (or *strongly* fair) if every transition that is enabled infinitely many times is also taken infinitely many times. A program is said to be fairly terminating if every fair execution sequence is finite.

In each of the papers there is one guarded repetition of a nondeterminate choice between a finite number of statements. In our more general context of arbitrary recursion, there is no fixed finite set of transitions. Therefore, we define fairness relative to a finite family of transition types. The generalization enables us to specify not only a fair treatment of the alternative refinements of a routine symbol, but also of the alternative results of the execution of a statement symbol. It gives a natural way to express the fact that only certain aspects of the execution are fair.

6.1. Transition types
A *transition type* U is defined to be a set of transitions in the configuration graph Z such that

(*) $\quad \forall y, z \in Z, v \in A^*: \quad ((y \to z) \in U \equiv (y; v \to z; v) \in U).$

If U is a transition type, then the *enabling type* $E(U)$ is defined by

$$E(U)=\{y\to z\mid\exists(y\to z')\in U\}.$$

One verifies that $E(U)$ is a transition type, and that it contains U.

Remark. Let a transition $y\to z$ be called *simple* if $y=\langle x, a\rangle$ with $a\in A$. Every transition type U is determined by the simple transitions that it contains. The enabling type $E(U)$ is determined by its source configurations of the form $y=\langle x, a\rangle$ with $a\in A$.

6.2. *Fair divergence*

A path $(z_i\mid i\in I)$ in the configuration graph Z is said to be *maximal* if it cannot be extended, i.e., if it is infinite or if its last configuration is terminal or aborted, cf. Definition 2.1. A path is said to be *divergent* if it is infinite or if its last configuration is aborted. It follows that $y\xrightarrow{*}\infty$ holds if and only if some divergent path starts in y.

Let U be a transition type. A path $(z_i\mid i\in I)$ in the graph Z is said to be *U-infinite* if there are infinitely many indices i such that the transition $z_{i-1}\to z_i$ belongs to the transition type U. The path is said to be *U-fair* if it is U-infinite or if it is not $E(U)$-infinite. Note that every finite path is U-fair.

Let $T=(U_j\mid j\in J)$ be a finite family of transition types U_j. A path in the configuration graph is said to be T-fair if it is U_j-fair for every index $j\in J$. A configuration y is said to be *T-fairly divergent* if there is a divergent and T-fair path that starts in y. We let $D(T)$ denote the set of the T-fairly divergent configurations.

6.3.

Proposition 6.1. *For any configuration y there is a maximal and T-fair path that starts in y.*

Remark. The proof of this result consists of an implementation of a fair strategy without deadlock. Such implementations are known already for a long time, cf. [5].

Proof (*sketch*). The configuration graph Z is enriched with an extra component: a patience vector $q=(q_j\mid j\in J)$. The coordinates q_j of q are nonnegative integers. The idea is that the value q_j decreases whenever a transition of the type U_j is enabled but not taken, and that it can only increase if a transition of type U_j is taken. More specifically, let an integral vector q as above be called a *patience vector* if

$$\forall n\in\mathbb{N}:\quad \text{card}\{j\in J\mid q_j<n\}\leq n.$$

Let Q be the set of all patience vectors. The cartesian product $Q\times Z$ is made into a directed graph by admitting a transition $\langle q, z\rangle\to\langle q', z'\rangle$ if and only if there is a transition $z\to z'$ in Z, and, for every index j such that $z\to z'$ does not belong to the type U_j, we have

$$q'_j\leq q_j\wedge((z\to z')\in E(U_j)\ \Rightarrow\ q'_j\leq q_j-1).$$

Let y be a given configuration. Choose an arbitrary patience vector $q \in Q$. In the graph $Q \times Z$ one constructs a maximal path that starts in the element $\langle q, y\rangle$. Using the structure of the graph $Q \times Z$, one proves that the projection of this path into the graph Z is a T-fair path, which is also maximal. For a detailed proof we refer to [12], where the same scheduler is used to schedule possibly infinitely many processes. The scheduler of [1] is slightly different. □

6.4.
Theorem 6.2. *The set $D(T)$ of the T-fairly divergent configurations is a set of potential divergence.*

Proof. We verify the conditions of Definition 5.3.

(a) Let $y \in Z1$. By Proposition 6.1 there is a maximal and T-fair path that starts in y. Since $y \in Z1$, the path is divergent. This proves that $y \in D(T)$.

(b) A configuration y belongs to $D(T)\backslash Z0$ if and only if some divergent and T-fair path of length ≥ 1 starts in y. The latter condition is equivalent to the existence of a transition $y \to z$ with $z \in D(T)$. This proves condition 5.3(b).

(c) The proof of condition 5.3(c) is an immediate adaptation of the proof of Proposition 2.2(b), based on the property 6.1(*) of the transition types. □

6.5. T-fair semantics

By Theorem 6.2, we can define the T-fair interpretation as the interpretation $m_{D(T)}: A^* \to \mathrm{Op}(X)$. This interpretation signals the possibility of divergence if and only if there exists an infinite T-fair execution path, cf. [1, 9, 15]. The corresponding T-fair preparator function is $m^0_{D(T)}: A^* \to \mathcal{F}(X)$. It gives the weakest precondition such that all T-fair calculations terminate in a state that satisfies the required postcondition.

Remark. We leave it to the reader to construct a T-just interpretation (cf. [15]) or equivalently a weakly T-fair interpretation (cf. [1]). The concept of impartiality of [15] can be modeled only approximately. One must be prepared to admit the result ψ in cases of inevitable divergence such as the program in [15, p. 271].

7. Concluding remarks

We have shown that the elementary configuration model is sufficiently powerful to deduce elegant denotational characterizations of the semantics of recursion under unbounded nondeterminacy. We have characterized the operationally justified interpretations, which include standard semantics as well as various forms of fairness.

If one wants to abolish Dijkstra's law of the excluded miracle (cf. [17]), our formalism can easily be adapted. In Definitions 1.0 and 1.3, one changes the definitions of operations and refinement functions, in the sense that the sets $f(x)$ and $R(h, x)$ are allowed to be empty. In Definitions 2.2 and 5.3 the proofs of nonemptiness are dropped. In Section 5, the set $Z1$ is replaced by $Z0$, and Sections 5.6 and 5.7 are changed accordingly.

Acknowledgment

I had the opportunity to present parts of this material to groups of colleagues and groups of students in Amsterdam, Groningen, and Austin (Texas). This was very stimulating. The suggestions and criticisms of the referees contributed even more.

References

[0] S. Abramsky, On semantic foundations for applicative multiprogramming, in: *Proc. Internat. Conf. on Automata, Languages and Programming*, Lecture Notes in Computer Science **154** (Springer, Berlin, 1983) 1-14.

[1] K.R. Apt and E.-R. Olderog, Proof rules and transformations dealing with fairness, *Sci. Comput. Programming* **3** (1983) 65-100.

[2] K.R. Apt and G.D. Plotkin, A Cook's tour of countable nondeterminism, in: *Proc. Internat. Conf. on Automata, Languages and Programming*, Lecture Notes in Computer Science **115** (Springer, Berlin, 1982) 479-494.

[3] K.R. Apt and G.D. Plotkin, Countable nondeterminism and random assignment, *J.ACM* **33** (1986) 724-767.

[4] J.W. de Bakker, *Mathematical Theory of Program Correctness* (Prentice-Hall, Englewood Cliffs, NJ, 1980).

[5] E.W. Dijkstra, A class of allocation strategies inducing bounded delays only, Tech. Rept. EWD 319, 1971.

[6] E.W. Dijkstra, *A Discipline of Programming* (Prentice-Hall, Englewood Cliffs, NJ, 1976).

[7] E.W. Dijkstra and C.S. Scholten, The operational interpretation of extreme solutions, Tech. Rept. EWD 883, 1984.

[8] E.W. Dijkstra, Predicate transformers (draft of Chapter 3), Tech. Rept. EWD 908, 1985.

[9] O. Grümberg, N. Francez, J.A. Makowsky and W.P. de Roever, A proof rule for fair termination of guarded commands, in: J.W. de Bakker, ed., *Algorithmic Languages* (North-Holland, Amsterdam, 1981) 399-416.

[10] E.C.R. Hehner, Predicative programming, Part I and II, *Comm. ACM* **27** (1984) 134-151.

[11] W.H. Hesselink, Nondeterminism in data types, a mathematical approach, *ACM Trans. Programming Languages and Systems* **10** (1988).

[12] W.H. Hesselink, Deadlock and fairness in morphisms of transition systems, *Theoret. Comput. Sci.* **59** (1988) 235-257 (this issue).

[13] C.A.R. Hoare, Some properties of predicate transformers, *J.ACM* **25** (1978) 461-480.

[14] R. Kuiper, An operational semantics for bounded nondeterminism equivalent to a denotational one, in: J.W. de Bakker, ed., *Algorithmic Languages* (North-Holland, Amsterdam, 1981) 373-398.

[15] D. Lehmann, A. Pnueli and J. Stavi, Impartiality, justice and fairness: the ethics of concurrent termination, in: *Proc. Internat. Conf. on Automata, Languages and Programming*, Lecture Notes in Computer Science **115** (Springer, Berlin, 1981) 264-277.

[16] J.-J.Ch. Meyer, Programming calculi based on fixed-point transformations: semantics and applications, Thesis, Vrije Universiteit, Amsterdam 1985.

[17] G. Nelson, A generalization of Dijkstra's calculus, SRC Rep. (DEC), April 1987.

[18] E.-R. Olderog and C.A.R. Hoare, Specification-oriented semantics for communicating processes, *Acta Inform.* **23** (1986) 9-66.

[19] D. Park, On the semantics of fair parallelism, in: *Abstract Software Specifications,* Lecture Notes in Computer Science **86** (Springer, Berlin, 1980) 504-526.

[20] W.P. de Roever, Dijkstra's predicate transformer, non-determinism, recursion, and termination, in: *Mathematical Foundations of Computer Science,* Lecture Notes in Computer Science **45** (Springer, Berlin, 1976) 472-481.