Process Algebra with Four-Valued Logic

Jan A. Bergstra[1,2] Alban Ponse[1]

[1] *University of Amsterdam, Programming Research Group*
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

[2] *Utrecht University, Department of Philosophy*
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

E-mail: `janb@wins.uva.nl` - `alban@wins.uva.nl`

Abstract

We propose a combination of a fragment of four-valued logic and process algebra. This fragment is geared to a simple relation with process algebra via the guarded command, and can easily be extended to a truth-functionally complete logic. We present an operational semantics in SOS-style, and a completeness result for ACP with conditionals and four-valued logic. Completeness is preserved under the restriction to some three-valued logics.

Key words & Phrases: process algebra, four-valued logic, guarded command.
1991 CR Categories: F.3, F.4.3, I.1.

1 Introduction

Three and four-valued logic arise naturally in descriptions of data types with error/exceptions and divergencies. We take this observation as a point of departure. In the case of three-valued logic a reference to [JM94] is illustrative: that paper describes in detail the use of a three-valued logic in the specification language VDM [Jon90]. This logic deals with partial predicates [BCJ84], and stems from Kleene [Kle38]. Another use of three-valued logic evolves if connectives are interpreted sequentially, and stems from McCarthy [McC63]. A third variant was defined by Bochvar [Boc39], in which a third truth value occurs that is strict with respect to all logical operations.

Process expressions (algorithms) can have actions and conditions parameterized by data. If one uses process expressions as a notation for algorithms, one faces the question how to interpret the conditional construct in case a condition evaluates to a truth value different from true or false. In this paper we provide a solution to that question for the four-valued logic of [BBR95] and process algebra in the style of ACP extended with guards as in [BB90]. The reason to use this particular four-valued logic is that it embeds the two typical, non-compatible forms of 'undefinedness' that we distinguish, and various associated logics. We consider this extension of ACP with four-valued logic as an integrated framework for the incorporation of sophisticated exception handling mechanisms in ACP style process algebra. As far as we know, similar work has not been reported about ACP or any other process algebra before. We continue this introduction with a short overview of all ingredients involved.

Four-Valued Logic. We consider four-valued proposition logic as introduced in [BBR95]. It is based on the set of values $\mathbb{T}_4$, which consists of

- M (meaningless),
- T (true),
- F (false), and
- D (divergence).

The following set of logical operations is defined and distinguished as truth-functionally complete:

- $\neg$ (negation),
- $\downarrow$ (definedness: distinguishing F, T from D, M),
- $\wedge$ (conjunction), and
- ${}_{\circ}\!\wedge$ (left sequential conjunction).

Truth tables for $\neg$, $\downarrow$, $\wedge$ and ${}_{\circ}\!\wedge$ are

x	$\neg x$
M	M
T	F
F	T
D	D

x	$\downarrow x$
M	F
T	T
F	T
D	F

$\wedge$	M	T	F	D
M	M	M	M	M
T	M	T	F	D
F	M	F	F	F
D	M	D	F	D

${}_{\circ}\!\wedge$	M	T	F	D
M	M	M	M	M
T	M	T	F	D
F	F	F	F	F
D	D	D	D	D

(Observe that $\wedge$ is monotonic in M.) The resulting logic is denoted as $\Sigma_4(\neg, \downarrow, \wedge, {}_{\circ}\!\wedge)$, where the subscript 4 refers to the four values of $\mathbb{T}_4$. Notice that the connectives $\wedge$, ${}_{\circ}\!\wedge$, and their duals are associative, and that $\wedge$ and $\vee$ are commutative as well. A complete axiomatization of its equational theory can be found in [Rod96]. The sequential fragment $\Sigma_4(\neg, {}_{\circ}\!\wedge)$ was earlier axiomatized in [BP96]. Kleene's three-valued logic is embedded in $\Sigma_4(\neg, \downarrow, \wedge, {}_{\circ}\!\wedge)$ by $\mathsf{T}, \mathsf{F}, \mathsf{D}$ and $\neg, \wedge$, Bochvar's strict three-valued logic by $\mathsf{T}, \mathsf{F}, \mathsf{M}$ and $\neg, \wedge$, and McCarthy's sequential three-valued logic by T, F, one of D, M, and $\neg$, ${}_{\circ}\!\wedge$, where the asymmetric symbol for left sequential conjunction is introduced in [BBR95].

Process Algebra with Two-Valued Logic. A basic construct in the combination of two-valued propositional logic and process algebra is the *conditional* $x \lhd \phi \rhd y$, introduced in [BB90]. Here x, y are processes, and ϕ is a proposition. This operation satisfies (among others) the following axioms:

$$\begin{aligned} x \lhd \mathsf{T} \rhd y &= x, \\ x \lhd \mathsf{F} \rhd y &= y, \\ x \lhd \phi \rhd y &= y \lhd \neg\phi \rhd x. \end{aligned}$$

The notation $\_ \lhd \_ \rhd \_$ stems from [HHJ+87], and in that paper it is argued that $x \lhd \phi \rhd y$ expresses **if** ϕ **then** x **else** y **fi**.

The special constant δ represents the *inactive* or *deadlocking* process, and is axiomatized by $x+\delta = x$ and $\delta \cdot x = \delta$ where $+$ represents 'choice' (so inaction never is an alternative) and $\cdot$ represents sequential composition. With δ the *guarded command* from [Dij76] can be represented in process algebra. This operation is introduced in process algebra in [BB90] with notation $:\rightarrow$, and satisfies

$$\begin{aligned} \phi :\rightarrow x &= x \lhd \phi \rhd \delta, \\ \mathsf{T} :\rightarrow x &= x, \\ \mathsf{F} :\rightarrow x &= \delta. \end{aligned}$$

For the guarded command, the following axiom relates $\vee$ and $+$:

$$\phi \vee \psi :\rightarrow x = \phi :\rightarrow x + \psi :\rightarrow x.$$

Its soundness follows from commutativity of both $\vee$ and $+$, and the axioms $x + x = x$ and $x + \delta = x$. An axiom that reduces repeated application of the guarded command is:

$$\phi :\rightarrow (\psi :\rightarrow x) = \phi \wedge \psi :\rightarrow x$$

(note the symmetry in $\phi \wedge \psi$). A different approach to process algebra with two-valued logic is described in [GP94], where propositions are considered process constants (e.g. $\phi \cdot x + \neg\phi \cdot y$ expresses $x \lhd \phi \rhd y$, and $\phi \cdot \psi$ expresses $\phi \wedge \psi$).

Interpretation of Conditionals with Non-Classical Values. A simple point of departure for defining conditionals over *four-valued* propositional logic is to start from considerations about the interpretation of conditions. We view a meaningless condition as one that destroys all operational behaviour (the process expression contains an irreparable error), and a divergent one as less dramatic in the presence of alternatives. This gives the following new axioms:

$$\begin{aligned} \mathsf{D} :\rightarrow x &= \delta, \\ \mathsf{M} :\rightarrow x &= \mu. \end{aligned}$$

The constant μ is new here, and represents *meaningless* as a process.[1] This constant is axiomatized by

$$\begin{aligned} x + \mu &= \mu \quad \text{“meaningless ruins each alternative,} \\ \mu \cdot x &= \mu \quad \text{and is perpetual”.} \end{aligned}$$

With $\vee$ and ${}_{\circ}\!\!\wedge$ we define

$$\begin{aligned} \phi :\rightarrow x + \psi :\rightarrow x &= \phi \vee \psi :\rightarrow x, \\ \phi :\rightarrow (\psi :\rightarrow x) &= \phi \,{}_{\circ}\!\!\wedge\, \psi :\rightarrow x, \end{aligned}$$

which for $\phi, \psi \in \{\mathsf{T}, \mathsf{F}\}$ both agree with the axioms for conditionals over two-valued proposition logic. The second axiom expresses that also the order of evaluation in $\phi :\rightarrow (\psi :\rightarrow x)$ is crucial, and replaces the earlier mentioned symmetric version.

Structure of the Paper. In the next section we present the fragment of four-valued logic based on $\neg$ and the logical operations $\vee$ and ${}_{\circ}\!\!\wedge$ as introduced above. In Section 3 we combine this fragment with an extension of ACP. In the next two sections we define an operational semantics and bisimulation equivalence, and we prove a completeness result. In Section 6 we consider four examples in process algebra with four-valued logic. The paper is ended with a few words on actions parameterized with non-classical values, and process algebra with three-valued logics.

Acknowledgements. Piet Rodenburg, Alex Sellink, and Wan Fokkink are thanked for discussions. Furthermore, the contents of the paper [BBR95], in which various three-valued logics are integrated turned out to have an unexpected heuristic value for us.

[1]These laws seem also to characterize the *chaos* process χ, which stems from CSP [BHR84]. However, χ characterizes the effect of infinite internal activity, and its laws capture an intuition that is not useful in our set-up. Modeling internal activity explicitly would distinguish μ and χ.

2 A Four-Valued Logical with Propositions

We consider the following set of logical operations on the set $\mathbb{T}_4 = \{\mathsf{M}, \mathsf{T}, \mathsf{F}, \mathsf{D}\}$ of truth values:

$$\begin{array}{rc} \neg : & \mathbb{T}_4 \to \mathbb{T}_4 \\ \wedge, \vee, \mathbin{{}_{\circ}\!\wedge}, \mathbin{{}^{\circ}\!\vee}, \mathbin{\wedge\!{}_{\circ}}, \mathbin{\vee\!{}^{\circ}} : & \mathbb{T}_4 \times \mathbb{T}_4 \to \mathbb{T}_4 \end{array}$$

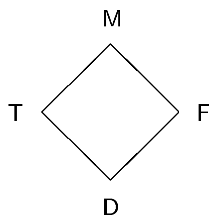
of which $\neg$, $\wedge$, and $\mathbin{{}_{\circ}\!\wedge}$ are defined by the truth tables in the previous section. The remaining operations are all definable:

$$\begin{array}{ll} \textit{Disjunction}: & x \vee y \triangleq \neg(\neg x \wedge \neg y), \\ \textit{Left sequential disjunction}: & x \mathbin{{}^{\circ}\!\vee} y \triangleq \neg(\neg x \mathbin{{}_{\circ}\!\wedge} \neg y), \\ \textit{Right sequential conjunction}: & x \mathbin{\wedge\!{}_{\circ}} y \triangleq y \mathbin{{}_{\circ}\!\wedge} x, \\ \textit{Right sequential disjunction}: & x \mathbin{\vee\!{}^{\circ}} y \triangleq y \mathbin{{}^{\circ}\!\vee} x. \end{array}$$

We represent the resulting logic by $\Sigma_4(\neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$ (thus without definedness, which indeed cannot be defined anymore). We do not provide an axiomatization, and use the identities implied by the truth tables for $\Sigma_4(\neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$. (This gives $4 + 16 + 16 = 36$ axioms, or 40 if we include the dual and right sequential operations). A perhaps convenient representation of these identities (and their duals) is given by the following graphs

$$\begin{array}{ccccccc} \mathsf{M} & & \mathsf{M} & & & & \\ \uparrow & & \uparrow & & & & \\ \mathsf{F} & & \mathsf{T} & & & & \\ \uparrow & & \uparrow & & & & \\ \mathsf{D} & & \mathsf{D} & & \mathsf{M} \quad \mathsf{F} \quad \mathsf{D} & & \mathsf{M} \quad \mathsf{T} \quad \mathsf{D} \\ \uparrow & & \uparrow & & \nwarrow \;\; \uparrow \;\; \nearrow & & \nwarrow \;\; \uparrow \;\; \nearrow \\ \mathsf{T} & & \mathsf{F} & & \mathsf{T} & & \mathsf{F} \\ \hline \wedge & \overset{\neg}{\Longleftrightarrow} & \vee & & \mathbin{{}_{\circ}\!\wedge} & \overset{\neg}{\Longleftrightarrow} & \mathbin{{}^{\circ}\!\vee} \end{array}$$

Clearly, $x \diamondsuit y$ has the value of x, y which is highest in the $\diamondsuit$-graph if x, y are connected, and the value of x otherwise. Observe that all operations in $\Sigma_4(\neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$ are monotone[2] according to the following partial ordering:



In case we use proposition symbols from set $\mathbb{P}$, we shall write $\Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$, and for concise notation we shall identify $\Sigma_4(\neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$ and $\Sigma_4(\emptyset, \neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$. In order to interpret propositions $\phi, \psi, ...$ over $\mathbb{P}$,

[2]A binary operation is *monotone* if $x \leq y$ implies $f(x, c) \leq f(y, c)$ and $f(c, x) \leq f(c, y)$.

we use substitution on single proposition symbols: let $p, q \in \mathbb{P}$, then

$$
\begin{array}{rcl}
[\phi/p]q & \triangleq & q, \\
[\phi/p]p & \triangleq & \phi, \\
[\phi/p]c & \triangleq & c \text{ for } c \in \{\mathsf{M}, \mathsf{T}, \mathsf{F}, \mathsf{D}\}, \\
[\phi/p]\neg\psi & \triangleq & \neg[\phi/p]\psi, \\
[\phi/p](\psi_1 \diamondsuit \psi_2) & \triangleq & [\phi/p]\psi_1 \diamondsuit [\phi/p]\psi_2 \text{ for } \diamondsuit \in \{\wedge, \vee, \mathbin{{}_{\circ}\!\!\wedge}, \mathbin{{}^{\circ}\!\!\vee}, \mathbin{\wedge\!\!_{\circ}}, \mathbin{\vee\!\!^{\circ}}\},
\end{array}
$$

and as a proof rule the *excluded fifth rule*:

$$
\frac{\sigma(\phi) = \sigma(\psi) \qquad \text{for all } \sigma \in \{[\mathsf{M}/p], [\mathsf{T}/p], [\mathsf{F}/p], [\mathsf{D}/p]\}}{\phi = \psi}.
$$

All in all, this yields a complete evaluation system for $\Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\!\wedge})$. We write

$$\Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\!\wedge}) \models \phi = \psi$$

if $\phi = \psi$ follows from the system defined above. If $\mathbb{P}$ is fixed, we often only write $\models \phi = \psi$. The identities stated in the following lemma are used in proofs to come, and can all be easily checked.

Lemma 2.1. *The following identities hold in* $\Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\!\wedge})$*:*

1. $\models (\phi \vee \mathsf{T}) \mathbin{{}_{\circ}\!\!\wedge} \phi = \phi$,
2. $\models (\phi \vee \mathsf{T}) \mathbin{{}_{\circ}\!\!\wedge} (\psi \vee \mathsf{T}) \mathbin{{}_{\circ}\!\!\wedge} \phi = (\psi \vee \mathsf{T}) \mathbin{{}_{\circ}\!\!\wedge} \phi$,
3. $\models \phi \vee \mathsf{D} = \phi \mathbin{{}^{\circ}\!\!\vee} \mathsf{D}$.

3 Process Algebra with Four-Valued Logic

In the left column of Table 1 we present a slight modification of $\mathrm{ACP}(A, \gamma)$ [BK84a, BK84b, BW90]. Here A is a set of atomic actions, and γ a communication function that is commutative and associative. We take γ *total* on $A \times A \to A_\delta$, where $A_\delta = A \cup \{\delta\}$, and the communication merge $\mid$ commutative (CMC) (by which (CM6) and (CM9), the symmetric variants of (CM5) and (CM8) [BW90], become derivable). In the right column, meaningless (μ), pre-abstraction (t_I, i.e., renaming of all actions in I to action t), guarded command ($\phi :\to$ as a unary operation), and conditional are presented. Here, ϕ ranges over $\Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\!\wedge})$. The axioms in Table 1 are parameterized by action set $A_t = A \cup \{t\}$. We mostly suppress the $\cdot$ in process expressions, and brackets according to the following rules: $\cdot$ binds strongest, $:\to$ binds stronger than $\|$, $\mathbin{\|\!\!\lfloor}$, $\mid$, all of which in turn bind stronger than $+$.

Observe that $\mu \mid a = \mu$ for all $a \in A_{t\delta}$ by (GT), (G4) and (GCM6), and likewise $\mu \mid ax = \mu$. The axiom (GCM6) suggests a more general version of (CF1)–(CF2), and (GCM7) and (GCM8) can be seen as generalizations of (CM5) and (CM7), respectively. Also observe that

$$\phi :\to x \mid \psi :\to y \neq \phi \wedge \psi :\to (x \mid y)$$

(set $x = \mathsf{T} :\to \mu$ and $y = \mathsf{F} :\to \delta$). We use the acronym

$$\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$$

both to refer to this axiom system and the signature thus defined.

Table 1: The axiom system $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t,\gamma,\mathbb{P})$, where $a,b\in A_{t\delta}$, $H,I\subseteq A_t$.

(A1)	$x+(y+z)$	$=$	$(x+y)+z$	(M1)	$x+\mu$	$=$	μ
(A2)	$x+y$	$=$	$y+x$	(M2)	$\mu\cdot x$	$=$	μ
(A3)	$x+x$	$=$	x	(M3)	$\mu \mid x$	$=$	μ
(A4)	$(x+y)z$	$=$	$xz+yz$	(GM)	$\mathsf{M} :\rightarrow x$	$=$	μ
(A5)	$(xy)z$	$=$	$x(yz)$	(GT)	$\mathsf{T} :\rightarrow x$	$=$	x
(A6)	$x+\delta$	$=$	x	(GF)	$\mathsf{F} :\rightarrow x$	$=$	δ
(A7)	δx	$=$	δ	(GD)	$\mathsf{D} :\rightarrow x$	$=$	δ
				(Cond)	$x \lhd \phi \rhd y$	$=$	$\phi :\rightarrow x + \neg\phi :\rightarrow y$
(CF1)	$a \mid b$	$=$	$\gamma(a,b)$ if $a,b\in A_t$	(GC1)	$\phi :\rightarrow x + \psi :\rightarrow x$	$=$	$\phi\vee\psi :\rightarrow x$
(CF2)	$a \mid \delta$	$=$	δ	(GC2)	$\phi :\rightarrow x + \phi :\rightarrow y$	$=$	$\phi :\rightarrow (x+y)$
(CM1)	$x \parallel y$	$=$	$(x \mathbin{\|\!\_} y + y \mathbin{\|\!\_} x) + x \mid y$	(GC3)	$(\phi :\rightarrow x)y$	$=$	$\phi :\rightarrow xy$
(CM2)	$a \mathbin{\|\!\_} x$	$=$	ax	(GCL4)	$\phi :\rightarrow (\psi :\rightarrow x)$	$=$	$\phi \mathbin{{}_{\circ}\!\!\wedge} \psi :\rightarrow x$
(CM3)	$ax \mathbin{\|\!\_} y$	$=$	$a(x \parallel y)$	(GC5)	$\phi :\rightarrow x \mathbin{\|\!\_} y$	$=$	$\phi :\rightarrow (x \mathbin{\|\!\_} y)$
(CM4)	$(x+y) \mathbin{\|\!\_} z$	$=$	$x \mathbin{\|\!\_} z + y \mathbin{\|\!\_} z$	(GCM6)	$\phi :\rightarrow a \mid \psi :\rightarrow b$	$=$	$\phi\wedge\psi :\rightarrow a \mid b$
(CMC)	$x \mid y$	$=$	$y \mid x$	(GCM7)	$\phi :\rightarrow ax \mid \psi :\rightarrow b$	$=$	$\phi\wedge\psi :\rightarrow (a \mid b)x$
(CM5)	$ax \mid b$	$=$	$(a \mid b)x$	(GCM8)	$\phi :\rightarrow ax \mid \psi :\rightarrow by$	$=$	$\phi\wedge\psi :\rightarrow (a \mid b)(x \parallel y)$
(CM7)	$ax \mid by$	$=$	$(a \mid b)(x \parallel y)$	(DGC)	$\partial_H(\phi :\rightarrow x)$	$=$	$\phi :\rightarrow \partial_H(x)$
(CM8)	$(x+y) \mid z$	$=$	$x \mid z + y \mid z$	(TGC)	$t_I(\phi :\rightarrow x)$	$=$	$\phi :\rightarrow t_I(x)$
(D1)	$\partial_H(a)$	$=$	a if $a\notin H$	(T1)	$t_I(a)$	$=$	a if $a\notin I$
(D2)	$\partial_H(a)$	$=$	δ if $a\in H$	(T2)	$t_I(a)$	$=$	t if $a\in I$
(D3)	$\partial_H(x+y)$	$=$	$\partial_H(x)+\partial_H(y)$	(T3)	$t_I(x+y)$	$=$	$t_I(x)+t_I(y)$
(D4)	$\partial_H(xy)$	$=$	$\partial_H(x)\partial_H(y)$	(T4)	$t_I(xy)$	$=$	$t_I(x)t_I(y)$

In order to combine process algebra and four-valued logic, we finally introduce the 'rule of equivalence'

$$(\mathrm{ROE})\quad \frac{\models \phi=\psi}{\vdash \phi :\rightarrow x = \psi :\rightarrow x}$$

This rule reflects the 'rule of consequence' in Hoare's Logic (cf. [Apt81]). We write

$$\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t,\gamma,\mathbb{P}) + \mathrm{ROE}_4 \vdash x=y,$$

or shortly $\vdash x=y$, if $x=y$ follows from the axioms of $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t,\gamma,\mathbb{P})$, the axioms and rules for $\Sigma_4(\mathbb{P},\neg,\wedge,{}_{\circ}\!\!\wedge)$, and the appropriate rule of equivalence

$$(\mathrm{ROE}_4)\quad \frac{\Sigma_4(\mathbb{P},\neg,\wedge,{}_{\circ}\!\!\wedge) \models \phi=\psi}{\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t,\gamma,\mathbb{P}) \vdash \phi :\rightarrow x = \psi :\rightarrow x}$$

We end this section with some useful derivabilities.

Lemma 3.1. *The following identities can be derived in* $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4$*:*

1. $\vdash \phi :\to \delta + x = \phi \vee \mathsf{T} :\to x$,
2. $\vdash \phi :\to x + y = \phi :\to x + \phi \vee \mathsf{T} :\to y$,
3. $\vdash \phi :\to x = \phi \mathbin{{}^{\circ}\!\vee} \mathsf{D} :\to x$.

Proof. As for 1. This is just an application of axioms (GT) and (GC1).

As for 2. Use $\phi :\to x = \phi :\to (x + \delta) = \phi :\to x + \phi :\to \delta$, and apply 1 on $\phi :\to \delta + y$.

As for 3. We apply ROE_4 on the identity proved in Lemma 2.1.3:

$$\phi :\to x = \phi :\to x + \delta = \phi :\to x + \mathsf{D} :\to x = \phi \vee \mathsf{D} :\to x \overset{2.1.3}{=} \phi \mathbin{{}^{\circ}\!\vee} \mathsf{D} :\to x. \blacksquare$$

4 Operational Semantics for $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$

In this section we provide $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$ with an operational semantics. Naturally, interpretations of the propositions occurring at 'top level' in a process expression also determine this semantics. As an example, consider for $p \in \mathbb{P}$, $a \in A_t$ the expression

$$p :\to a.$$

Depending on the interpretation of p, this process either behaves as μ, as a, or as δ.

Given (non-empty) set $\mathbb{P}$ of proposition symbols, let w range over the *valuations* (interpretations) $\mathcal{W}$ of $\mathbb{P}$ in $\mathbb{T}_4$. In the usual way we extend w to $\Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$:

$$\begin{aligned} w(c) &\triangleq c \text{ for } c \in \{\mathsf{M}, \mathsf{T}, \mathsf{F}, \mathsf{D}\}, \\ w(\neg\phi) &\triangleq \neg(w(\phi)), \\ w(\phi \mathbin{\Diamond} \psi) &\triangleq w(\phi) \mathbin{\Diamond} w(\psi) \text{ for } \Diamond \in \{\wedge, \vee, \mathbin{{}_{\circ}\!\wedge}, \mathbin{{}^{\circ}\!\vee}, \wedge_{\circ}, \vee^{\circ}\}. \end{aligned}$$

With the system defined in Section 2, it follows that if

$$\models w(\phi) = w(\psi)$$

for all $w \in \mathcal{W}$, then $\models \phi = \psi$. For each $w \in \mathcal{W}$ and $\phi \in \Sigma_4(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$ we define inductively in Table 2 the unary predicate *meaningless*, notation

$$\mu(w, \_)$$

over process terms in $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$. This predicate defines which process expressions represent the meaningless process μ under a certain valuation w.

The axioms and rules for $\mu(w, \_)$ given in Table 2 are extended by axioms and rules given in Table 3, which define transitions

$$\_ \xrightarrow{w,a} \_ \subseteq \mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P}) \times \mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$$

and unary "tick-predicates" or "termination transitions"

$$\_ \xrightarrow{w,a} \surd \subseteq \mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$$

for all $w \in \mathcal{W}$ and $a \in A_t$. Transitions characterize under which interpretations a process expression defines the possibility to execute an atomic action, and what remains to be executed (if anything, otherwise $\surd$ symbolizes successful termination).

Table 2: Rules for μ in *panth*-format

μ	$\mu(w,\mu)$	
$:\rightarrow$	$\mu(w, \phi :\rightarrow x)$ if $w(\phi) = \mathsf{M}$	$\dfrac{\mu(w,x)}{\mu(w, \phi :\rightarrow x)}$ if $w(\phi) = \mathsf{T}$
$+, \cdot, \parallel, \llfloor, \mid, \partial_H, t_I$	$\dfrac{\mu(w,x)}{\begin{array}{l}\mu(w, x+y)\\ \mu(w, y+x)\\ \mu(w, x \cdot y)\\ \mu(w, x \parallel y)\\ \mu(w, x \llfloor y)\\ \mu(w, x \mid y)\\ \mu(w, \partial_H(x))\\ \mu(w, t_I(x))\end{array}}$	

The following facts follow easily by induction on the structure of the process expression involved, and clarify the relations between (termination) transitions and the meaningless predicate $\mu(w, \_)$:

Lemma 4.1. *If $x \xrightarrow{w,a} x'$ or $x \xrightarrow{w,a} \surd$ for some w and a, then $\neg\mu(w,x)$.*
If $\mu(w,x)$, then $x \not\xrightarrow{w,a} x'$ and $x \not\xrightarrow{w,a} \surd$ for all $a \in A_t$.

Note that the converse implications do not hold (take $x = \delta$). So under a certain valuation w, a process expression either resembles meaningless, or defines outgoing transitions, or represents deadlock (δ).

The axioms and rules in Tables 2 and 3 yield a structured operational semantics (SOS) with negative premises the style of Groote [Gro93]. Moreover, they satisfy the so called *panth-format* defined by Verhoef [Ver95], which also implies the following notion of bisimulation equivalence:

Definition 4.2. *Let $B \subseteq \mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t,\gamma,\mathbb{P}) \times \mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t,\gamma,\mathbb{P})$. Then B is a bisimulation if for all P, Q with PBQ the following conditions hold for all $w \in \mathcal{W}$ and $a \in A_t$:*

- $\mu(w,P) \iff \mu(w,Q)$,
- $\forall P'\ (P \xrightarrow{w,a} P' \implies \exists Q'(Q \xrightarrow{w,a} Q' \wedge P'BQ'))$,
- $\forall Q'\ (Q \xrightarrow{w,a} Q' \implies \exists P'(P \xrightarrow{w,a} P' \wedge P'BQ'))$,
- $P \xrightarrow{w,a} \surd \iff Q \xrightarrow{w,a} \surd$.

Two processes P, Q are bisimilar, notation

$$P \leftrightarrow Q,$$

if there exists a bisimulation containing the pair (P,Q).

Table 3: Transition rules in *panth*-format

$a \in A_t$	$a \xrightarrow{w,a} \surd$	
$\cdot, \lfloor\!\lfloor$	$\dfrac{x \xrightarrow{w,a} \surd}{\begin{array}{c} x \cdot y \xrightarrow{w,a} y \\ x \lfloor\!\lfloor y \xrightarrow{w,a} y \end{array}}$	$\dfrac{x \xrightarrow{w,a} x'}{\begin{array}{c} x \cdot y \xrightarrow{w,a} x'y \\ x \lfloor\!\lfloor y \xrightarrow{w,a} x' \parallel y \end{array}}$
$+, \parallel$	$\dfrac{x \xrightarrow{w,a} \surd \quad \neg\mu(w,y)}{\begin{array}{c} x + y \xrightarrow{w,a} \surd \\ y + x \xrightarrow{w,a} \surd \\ x \parallel y \xrightarrow{w,a} y \\ y \parallel x \xrightarrow{w,a} y \end{array}}$	$\dfrac{x \xrightarrow{w,a} x' \quad \neg\mu(w,y)}{\begin{array}{c} x + y \xrightarrow{w,a} x' \\ y + x \xrightarrow{w,a} x' \\ x \parallel y \xrightarrow{w,a} x' \parallel y \\ y \parallel x \xrightarrow{w,a} y \parallel x' \end{array}}$
$\mid, \parallel$ (communication)	$\dfrac{x \xrightarrow{w,a} \surd \quad y \xrightarrow{w,b} \surd}{\begin{array}{c} x \mid y \xrightarrow{w,c} \surd \\ x \parallel y \xrightarrow{w,c} \surd \end{array}} \quad a \mid b = c$	$\dfrac{x \xrightarrow{w,a} \surd \quad y \xrightarrow{w,b} y'}{\begin{array}{c} x \mid y \xrightarrow{w,c} y' \\ x \parallel y \xrightarrow{w,c} y' \end{array}} \quad a \mid b = c$
	$\dfrac{x \xrightarrow{w,a} x' \quad y \xrightarrow{w,b} \surd}{\begin{array}{c} x \mid y \xrightarrow{w,c} x' \\ x \parallel y \xrightarrow{w,c} x' \end{array}} \quad a \mid b = c$	$\dfrac{x \xrightarrow{w,a} x' \quad y \xrightarrow{w,b} y'}{\begin{array}{c} x \mid y \xrightarrow{w,c} x' \parallel y' \\ x \parallel y \xrightarrow{w,c} x' \parallel y' \end{array}} \quad a \mid b = c$
∂_H	$\dfrac{x \xrightarrow{w,a} \surd}{\partial_H(x) \xrightarrow{w,a} \surd}$ if $a \notin H$	$\dfrac{x \xrightarrow{w,a} x'}{\partial_H(x) \xrightarrow{w,a} \partial_H(x')}$ if $a \notin H$
t_I	$\dfrac{x \xrightarrow{w,a} \surd}{t_I(x) \xrightarrow{w,a} \surd}$ if $a \notin I$	$\dfrac{x \xrightarrow{w,a} x'}{t_I(x) \xrightarrow{w,a} t_I(x')}$ if $a \notin I$
	$\dfrac{x \xrightarrow{w,a} \surd}{t_I(x) \xrightarrow{w,t} \surd}$ if $a \in I$	$\dfrac{x \xrightarrow{w,a} x'}{t_I(x) \xrightarrow{w,t} t_I(x')}$ if $a \in I$
$:\rightarrow$	$\dfrac{x \xrightarrow{w,a} \surd}{\phi :\rightarrow x \xrightarrow{w,a} \surd}$ if $w(\phi) = \mathsf{T}$	$\dfrac{x \xrightarrow{w,a} x'}{\phi :\rightarrow x \xrightarrow{w,a} x'}$ if $w(\phi) = \mathsf{T}$

Furthermore, from [FG96, Ver95] it easily follows that the transitions and meaningless instances defined by these axioms and rules are uniquely determined. This can be established with help of the following *stratification* S:

$$S(\mu(w,x)) = 0,$$
$$S(x \xrightarrow{w,a} x') = S(x \xrightarrow{w,a} \surd) = 1.$$

It follows that we can apply the main result of [Ver95]: bisimilarity is a *congruence* relation for all operations involved. Notice that guarded commands are considered here as *unary* operations: for each $\phi \in \Sigma_4(\mathbb{P}, \neg, \wedge, {}_{\circ}\!\wedge)$ there is an operation $\phi :\rightarrow \_$. Likewise, conditionals are considered as binary operations. It is not difficult, but tedious to establish that in the bisimulation model thus obtained all equations of Table 1 are true. Hence we conclude:

Lemma 4.3. *The system* $\mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4$ *is sound with respect to bisimulation: for all* $P,\ Q \in \mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P})$,

$$\mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4 \vdash P = Q \quad\Longrightarrow\quad P \mathrel{\underline{\leftrightarrow}} Q.$$

5 Completeness

In this section we prove completeness of $\mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4$, i.e.,

$$P \mathrel{\underline{\leftrightarrow}} Q \quad\Longleftrightarrow\quad \mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4 \vdash P = Q.$$

Our proof is based on a representation of process expressions for which bisimilarity implies derivability in a straightforward way.

Definition 5.1. *A process expression* $P \in \mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P})$ *is a basic term if*

$$P \equiv \sum_{i \in I} \phi_i :\rightarrow Q_i$$

where $\equiv$ is used for syntactic equivalence, I is a finite, non-empty index set, $\phi_i \in \Sigma_4(\mathbb{P}, \neg, \wedge, {}_{\circ}\!\wedge)$, and $Q_i \in \{\delta, a, aR \mid a \in A_t,\ R$ a basic term$\}$.

Lemma 5.2. *All process expressions in* $\mathrm{ACP}_{\mathsf{D,M},\mu}(A_t, \gamma, \mathbb{P})$ *can be proved equal to a basic term.*

Proof. Standard induction on term complexity. ∎

For $a \in A_t$ and $\phi \in \Sigma_4(\mathbb{P}, \neg, \wedge, {}_{\circ}\!\wedge)$, the *height* of a basic term is defined by

$$h(\delta) = 0,$$
$$h(a) = 1,$$
$$h(\phi :\rightarrow x) = h(x),$$
$$h(x + y) = max(h(x), h(y)),$$
$$h(a \cdot x) = 1 + h(x).$$

Lemma 5.3. *If P is a basic term, there is a basic term P' with $\vdash P = P'$, $h(P') \leq h(P)$, and P' has either the form*

$$\psi :\rightarrow \delta, \tag{1}$$

or the form

$$\sum_{i\in I} \psi_i :\rightarrow Q_i \text{ with } \begin{array}{rl} (i) & \text{for all } i,j \in I,\ Q_i \not\equiv \delta, \text{ and } Q_i, Q_j \in A_t \Rightarrow Q_i \not\equiv Q_j \text{ if } i \neq j, \\ (ii) & \text{if } \exists i \in I, w \in \mathcal{W} \text{ such that } w(\psi_i) = \mathsf{M}, \text{ then } \forall j \in I,\ w(\psi_j) = \mathsf{M}, \\ (iii) & \text{for each } i \in I \text{ there is } w \in \mathcal{W} \text{ such that } w(\psi_i) = \mathsf{T}, \\ (iv) & \text{for no } i \in I \text{ and valuation } w, w(\psi_i) = \mathsf{F}. \end{array} \tag{2}$$

Proof. Assume

$$P \quad \equiv \quad \textstyle\sum_{i=1}^{n} \phi_i :\rightarrow Q_i$$

for some $n \geq 1$. By Lemma 3.1.1 we may assume that for all i either $Q_i \not\equiv \delta$ or $Q_i \equiv \delta$, which yields with axiom (GC1) form (1). Also with (GC1), each single action occurs at most once. This proves property (i) of form (2). Let

$$\overline{\phi} \quad \equiv \quad (\phi_1 \vee \mathsf{T}) \mathbin{{}_{\circ}\!\wedge} \ldots \mathbin{{}_{\circ}\!\wedge} (\phi_n \vee \mathsf{T}).$$

(Recall that $\mathbin{{}_{\circ}\!\wedge}$ is associative.) Observe that for each $w \in \mathcal{W}$, $w(\overline{\phi}) \in \{\mathsf{T}, \mathsf{M}\}$. Let furthermore

$$\begin{array}{rcl} \overline{\phi_i} & \equiv & \overline{\phi} \mathbin{{}_{\circ}\!\wedge} \phi_i, \\ P'' & \equiv & \sum_{i=1}^{n} \overline{\phi_i} :\rightarrow Q_i. \end{array}$$

Note that if $w(\overline{\phi_i}) = \mathsf{M}$ for some i and w, then $w(\overline{\phi_j}) = \mathsf{M}$ for all $j \in \{1, ..., n\}$. We show that

$$\vdash P = P'' \tag{3}$$

by induction on n.
<u>$n = 1$.</u> This follows immediately from Lemma 2.1.1.
<u>$n = k + 1$.</u> Let $\overline{\overline{\phi}} \equiv (\phi_2 \vee \mathsf{T}) \mathbin{{}_{\circ}\!\wedge} \ldots \mathbin{{}_{\circ}\!\wedge} (\phi_n \vee \mathsf{T})$, and $\overline{\overline{\phi_i}} \equiv \overline{\overline{\phi}} \mathbin{{}_{\circ}\!\wedge} \phi_i$. By induction we have

$$\vdash P = \phi_1 :\rightarrow Q_1 + \textstyle\sum_{i=2}^{n} \overline{\overline{\phi_i}} :\rightarrow Q_i.$$

With k applications of Lemma 3.1.2 and (GCL4), we obtain $\vdash P = \phi_1 :\rightarrow Q_1 + \sum_{i=2}^{n} \overline{\phi_i} :\rightarrow Q_i$. Doing the same once more yields

$$\vdash P = (\overline{\phi_2} \vee \mathsf{T}) \mathbin{{}_{\circ}\!\wedge} \phi_1 :\rightarrow Q_1 + \textstyle\sum_{i=2}^{n} \overline{\phi_i} :\rightarrow Q_i.$$

Now it follows easily that $\models \overline{\phi_1} = (\overline{\phi_2} \vee \mathsf{T}) \mathbin{{}_{\circ}\!\wedge} \phi_1$ (recall that $\overline{\phi_2} \equiv \overline{\phi} \mathbin{{}_{\circ}\!\wedge} \phi_2$ and $w(\overline{\phi}) \in \{\mathsf{T}, \mathsf{M}\}$). This finishes the proof of (3), and proves properties (i) and (ii) of form (2) for P''.

Next we consider all summands from P'' for which no valuation makes the condition true. For each such summand $\overline{\phi_i} :\rightarrow Q_i$ it holds that $\models \overline{\phi_i} = \overline{\phi_i} \mathbin{{}_{\circ}\!\wedge} \mathsf{F}$, and thus

$$\begin{array}{rcl} \vdash \overline{\phi_i} :\rightarrow Q_i & = & \overline{\phi_i} \mathbin{{}_{\circ}\!\wedge} \mathsf{F} :\rightarrow Q_i \\ & = & \overline{\phi_i} :\rightarrow (\mathsf{F} :\rightarrow Q_i) \\ & = & \overline{\phi_i} :\rightarrow \delta. \end{array}$$

In case all summands can be proved equal to $\overline{\phi_j} :\to \delta$ in this way, we are done using (3):

$$\vdash P = \overline{\phi_1} \vee ... \vee \overline{\phi_n} :\to \delta.$$

In the other case, $w(\overline{\phi_i}) = \mathsf{T}$ for certain w, i. If $\vdash \overline{\phi_j} :\to Q_j = \overline{\phi_j} :\to \delta$ for some j, then by Lemma 3.1.2, $\vdash \overline{\phi_j} :\to \delta + \overline{\phi_i} :\to Q_i = (\overline{\phi_j} \vee \mathsf{T}) \mathbin{{}_{\circ}\!\wedge} \overline{\phi_i} :\to Q_i$. Now $\models \overline{\phi_i} = (\overline{\phi_j} \vee \mathsf{T}) \mathbin{{}_{\circ}\!\wedge} \overline{\phi_i}$ as was already used in the proof of (3). Hence we obtain

$$\vdash P'' = \sum_{i=1}^{k} \overline{\phi_i} :\to Q_i$$

with $k \leq n$ (and possibly some rearrangement of indices), and for each $i \in \{1, ..., k\}$ there is a valuation w with $w(\overline{\phi_i}) = \mathsf{T}$. This proves property (iii) of form (2), and preserves properties (i) and (ii) for P''. Finally we define

$$\begin{array}{lcl} \psi_i & \equiv & \overline{\phi_i} \mathbin{{}^{\circ}\!\vee} \mathsf{D} \\ P' & \equiv & \sum_{i=1}^{k} \psi_i :\to Q_i. \end{array}$$

By Lemma 3.1.3 and identity (3) we obtain

$$\vdash P = P'.$$

Moreover, by definition of ψ_i it follows that $w(\psi_i) \neq \mathsf{F}$ for all w, i, which proves property (iv). Application of this lemma of course preserves properties (i)–(iv). ∎

Lemma 5.4. *Let P_1, P_2 be basic terms. Then*

$$P_1 \mathrel{\underline{\leftrightarrow}} P_2 \quad \Longrightarrow \quad \vdash P_1 = P_2.$$

Proof. By the previous Lemma 5.3, we may assume that both P_1 and P_2 satisfy either form (1) or form (2) given there. We proceed by induction on $h = max(h(P_1), h(P_2))$.

Let $h = 0$, then $P_n \equiv \phi_n :\to \delta$ for $n = 1, 2$. So $\vdash \phi_1 :\to \delta \mathrel{\underline{\leftrightarrow}} \phi_2 :\to \delta$ and $w(\phi_1) = \mathsf{M} \Leftrightarrow w(\phi_2) = \mathsf{M}$.

Now $\phi_n :\to \delta = \phi_n :\to (\mathsf{D} :\to \delta) = \phi_n \mathbin{{}_{\circ}\!\wedge} \mathsf{D} :\to \delta = (\phi_n \mathbin{{}_{\circ}\!\wedge} \mathsf{D}) \mathbin{{}^{\circ}\!\vee} \mathsf{D} :\to \delta$. Notice that $\models (\phi_1 \mathbin{{}_{\circ}\!\wedge} \mathsf{D}) \mathbin{{}^{\circ}\!\vee} \mathsf{D} = (\phi_2 \mathbin{{}_{\circ}\!\wedge} \mathsf{D}) \mathbin{{}^{\circ}\!\vee} \mathsf{D}$, as for each valuation w both propositions evaluate to either D or M. In particular

$$\begin{array}{lclcl} w((\phi_1 \mathbin{{}_{\circ}\!\wedge} \mathsf{D}) \mathbin{{}^{\circ}\!\vee} \mathsf{D}) = \mathsf{M} & \Leftrightarrow & w(\phi_1) = \mathsf{M} & \Leftrightarrow & \mu(w, \phi_1 :\to \delta) \\ & \Leftrightarrow & \mu(w, \phi_2 :\to \delta) & \Leftrightarrow & w((\phi_2 \mathbin{{}_{\circ}\!\wedge} \mathsf{D}) \mathbin{{}^{\circ}\!\vee} \mathsf{D}) = \mathsf{M}. \end{array}$$

Consequently, $\vdash P_1 = P_2$.

Let $h > 0$ and $P_n \equiv \sum_{i \in I_n} \psi_{n,i} :\to Q_{n,i}$ for $n = 1, 2$. By the previous Lemma 5.3, we may assume that P_n satisfies form (2) given there. Furthermore, we may assume that for all $i \in I_n$, $Q_{n,i} \mathrel{\not\!\underline{\leftrightarrow}} Q_{n,j}$ for $j \in I_n \setminus \{i\}$. For the case $Q_{n,i} \equiv aR_{n,i}$ and $Q_{n,j} \equiv aR_{n,j}$ this follows by induction: $R_{n,i} \mathrel{\underline{\leftrightarrow}} R_{n,j}$ implies $\vdash R_{n,i} = R_{n,j}$, so $\vdash aR_{n,i} = aR_{n,j}$, and thus (GC1) could have been applied.

Now each summand of P_n can be proved equal to one in P_{3-n}, and by Lemma 5.3, each summand yields a transition for a certain $w \in \mathcal{W}$.

Assume that $P_n \xrightarrow{w,a} \surd$ for some w, a. Thus $w(\psi_{n,i}) = \mathsf{T}$ for some unique $i \in I_n$. By $P_1 \mathrel{\underline{\leftrightarrow}} P_2$, there is a unique $j \in I_{3-n}$ for which $P_{3-n} \xrightarrow{w,a} \surd$ and $\models \psi_{n,i} = \psi_{3-n,j}$ (the latter derivability follows from the representation as defined in Lemma 5.3 and the non-bisimilarity of different summands). Thus

$$\vdash \psi_{n,i} :\to a = \psi_{3-n,j} :\to a.$$

Assume that $P_n \xrightarrow{w,a} R_{n,i}$ for some w, a and unique $i \in I_n$. Thus $w(\psi_{n,i}) = \mathsf{T}$. By $P_1 \leftrightarroweq P_2$, there must be some unique $j \in I_{3-n}$ for which $P_{3-n} \xrightarrow{w,a} R_{3-n,j}$ and $R_{n,i} \leftrightarroweq R_{3-n,j}$, and for which $\models \psi_{n,i} = \psi_{3-n,j}$ follows from Lemma 5.3. By induction we find $\vdash R_{n,i} = R_{3-n,j}$, and therefore $\vdash aR_{n,i} = aR_{3-n,j}$ and hence

$$\vdash \psi_{n,i} :\to aR_{n,i} = \psi_{3-n,j} :\to aR_{3-n,j}.$$

By the derivabilities above and symmetry, it follows that $\vdash P_1 = P_2$. ∎

With Lemmas 5.2, 5.3, 5.4, and soundness (Lemma 4.3) we obtain:

Theorem 5.5. *The system* $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4$ *is complete with respect to bisimulation: for all* $P,\ Q \in \mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$,

$$\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P}) + \mathrm{ROE}_4 \vdash P = Q \quad\Longleftrightarrow\quad P \leftrightarroweq Q.$$

6 Examples

In this section we describe some examples in process algebra with four-valued logic:

6.1. Natural numbers extended with $\{\mathsf{M}, \mathsf{D}\}$,
6.2. Minimal History Logic,
6.3. Action History Logic,
6.4. Floyd-Hoare Logic.

Example 6.1 ($\omega_{\mathsf{M},\mathsf{D}}$, *natural numbers with* $\{\mathsf{M}, \mathsf{D}\}$)**.** Consider the natural numbers

$$\omega = \{0, S(0), S(S(0)), ...\},$$

and let $k, l, m, ...$ range over ω. We write $S(0) = 1$, $S(1) = 2$ etc. Using sequential connectives, one easily defines the predicates Z (zero predicate) and T (number predicate):

$$\begin{aligned} Z(0) &= \mathsf{T}, \\ Z(S(x)) &= \neg T(x), \\ T(x) &= Z(x) \mathbin{{}^{\circ}\!\!\vee} \mathsf{T}. \end{aligned}$$

It follows that $Z(0) = \mathsf{T}$, $Z(k+1) = \mathsf{F}$, and $T(k) = \mathsf{T}$.

We extend ω to $\omega_{\mathsf{M}} = \omega \cup \{\mathsf{M}\}$, as to give the following definition of the predecessor function P

$$\begin{aligned} P(0) &= \mathsf{M}, \\ P(S(x)) &= x. \end{aligned}$$

We judge this a prototypical occurrence of M.

We extend the functions and predicates defined above to ω_{M} by setting $S(\mathsf{M}) = Z(\mathsf{M}) = T(\mathsf{M}) = P(\mathsf{M}) = \mathsf{M}$, which makes predicates Z and T more significant, and which allows us to define left sequential equality, notation ${}_{\circ}\!\!\equiv$, in a recursive way:

$$\begin{aligned} x \mathrel{{}_{\circ}\!\!\equiv} y = {} & (Z(x) \mathbin{{}_{\circ}\!\!\wedge} Z(y)) \mathbin{{}^{\circ}\!\!\vee} \\ & (\neg Z(x) \mathbin{{}_{\circ}\!\!\wedge} \neg Z(y) \mathbin{{}_{\circ}\!\!\wedge} P(x) \mathrel{{}_{\circ}\!\!\equiv} P(y)). \end{aligned}$$

It follows that ${}_{\circ}\!\!\equiv$ is symmetric in ω_{M}, in particular $\mathsf{M} {}_{\circ}\!\!\equiv x = x {}_{\circ}\!\!\equiv \mathsf{M} = \mathsf{M}$.

In the following we describe a prototypical, generic occurrence of D caused by partiality. Let $f : \omega \to \{\mathsf{T}, \mathsf{F}\}$ be some arbitrary function. We define *semi-effective infinitary disjunction*, notation ${}^{\circ}\!\!\bigvee$, by

$$ {}^{\circ}\!\!\bigvee f = f(0) \mathbin{{}^{\circ}\!\!\vee} {}^{\circ}\!\!\bigvee (f \circ S). $$

The recursive definition of ${}^{\circ}\!\!\bigvee f$ implies computation of $f(0), f(1), f(2), \ldots$ till $f(n) = \mathsf{T}$ for some value n. In the particular case that for all $n \in \omega$, $f(n) = \mathsf{F}$, it makes sense to define

$$ {}^{\circ}\!\!\bigvee f = \mathsf{D}. $$

As an example, we consider the following partial definition of a subtraction function *subp* (this example is taken from [BCJ84]). The idea is that

$$ subp(x, y) \sim \textbf{if } x {}_{\circ}\!\!\equiv y \textbf{ then } 0 \textbf{ else } S(subp(S(x), y)). $$

(Thus $subp(x, y) = y - x$ if $x \leq y$.) We first define *subp* in our set-up with help of auxiliary function g

$$ \begin{aligned} subp(x, y) &= g(x, y, 0), \\ g(x, y, z) &= \begin{cases} z & \text{if } x {}_{\circ}\!\!\equiv y, \\ g(S(x), y, S(z)) & \text{otherwise.} \end{cases} \end{aligned} $$

We analyze computation of $g(x, y, z)$ with help of an auxiliary predicate Aux

$$ g(x, y, z) = v \Leftrightarrow Aux(x, y, z, v). $$

The predicate Aux can be recursively defined by

$$ \begin{aligned} Aux(x, y, z, v) \quad = \quad & (x {}_{\circ}\!\!\equiv y \mathbin{{}_{\circ}\!\!\wedge} z {}_{\circ}\!\!\equiv v) \mathbin{{}^{\circ}\!\!\vee} \\ & (\neg(x {}_{\circ}\!\!\equiv y) \mathbin{{}_{\circ}\!\!\wedge} Aux(S(x), y, S(z), v)). \end{aligned} $$

Furthermore, it follows that $subp(\mathsf{M}, x) = subp(x, \mathsf{M}) = \mathsf{M}$. Observe that $k + 1 {}_{\circ}\!\!\equiv 0 = \mathsf{F}$. Hence we can infer

$$ \begin{aligned} Aux(S(0), 0, 0, v) \quad = \quad & (S(0) {}_{\circ}\!\!\equiv 0 \mathbin{{}_{\circ}\!\!\wedge} 0 {}_{\circ}\!\!\equiv v) \\ & {}^{\circ}\!\!\vee \\ & (S(S(0)) {}_{\circ}\!\!\equiv 0 \mathbin{{}_{\circ}\!\!\wedge} S(0) {}_{\circ}\!\!\equiv v) \\ & {}^{\circ}\!\!\vee \\ & (S(S(S(0) {}_{\circ}\!\!\equiv 0 \mathbin{{}_{\circ}\!\!\wedge} S(S(0)) {}_{\circ}\!\!\equiv v) \\ & {}^{\circ}\!\!\vee \\ & \ldots \end{aligned} $$

Now take $f = \lambda x. S(x) {}_{\circ}\!\!\equiv 0 \mathbin{{}_{\circ}\!\!\wedge} x {}_{\circ}\!\!\equiv v$, thus $Aux(S(0), 0, 0, v) = {}^{\circ}\!\!\bigvee f$. By $S^{k+1}(0) {}_{\circ}\!\!\equiv 0 = \mathsf{F}$ it follows that $f(n) = \mathsf{F}$ for all $n \in \omega$, and hence $Aux(S(0), 0, 0, v) = \mathsf{D}$, irrespective of v. So we obtain

$$ subp(S(0), 0) = \mathsf{D}. $$

We extend the functions and predicates defined above to $\omega_{\mathsf{M},\mathsf{D}} = \omega_{\mathsf{M}} \cup \{\mathsf{D}\}$ by setting $S(\mathsf{D}) = Z(\mathsf{D}) = P(\mathsf{D}) = \mathsf{D}$. Observe that $\mathsf{D} {}_{\circ}\!\!\equiv x = \mathsf{D}$ and $\mathsf{M} {}_{\circ}\!\!\equiv x = \mathsf{M}$ in $\omega_{\mathsf{M},\mathsf{D}}$.

Example 6.2 (MHL, *Minimal History Logic*)**.** Let I be the assertion which is true of the initial state of a process and false thereafter. Furthermore, let $\mathsf{P}_4(\phi)$ be the assertion that ϕ is valid in the previous state (i.e., the state before the last action); if there is no such state, $\mathsf{P}_4(\phi) = \mathsf{M}$. We use the subscript 4 to distinguish P_4 from the usual P (weak past, [Bur84]), which we can now define as

$$\mathsf{P}(\phi) = \neg\mathsf{I} \mathbin{{}_{\circ}\!\!\wedge} \mathsf{P}_4(\phi),$$

so that $\mathsf{P}(\phi) = \mathsf{F}$ in the initial state. Though P_4 is a modality, we have

$$\begin{aligned}
\mathsf{P}_4(\mathsf{T}) &= \neg\mathsf{I} \mathbin{{}^{\circ}\!\!\vee} \mathsf{M},\\
\mathsf{P}_4(\mathsf{F}) &= \mathsf{I} \mathbin{{}_{\circ}\!\!\wedge} \mathsf{M},\\
\mathsf{P}_4(\neg\phi) &= \neg\mathsf{P}_4(\phi),\\
\mathsf{P}_4(\phi \wedge \psi) &= \mathsf{P}_4(\phi) \wedge \mathsf{P}_4(\psi),\\
\mathsf{P}_4(\phi \mathbin{{}_{\circ}\!\!\wedge} \psi) &= \mathsf{P}_4(\phi) \mathbin{{}_{\circ}\!\!\wedge} \mathsf{P}_4(\psi),
\end{aligned}$$

and one can set $\mathsf{P}_4(\mathsf{M}) = \mathsf{M}$ and $\mathsf{P}_4(\mathsf{D}) = (\mathsf{I} \mathbin{{}_{\circ}\!\!\wedge} \mathsf{M}) \mathbin{{}^{\circ}\!\!\vee} (\neg\mathsf{I} \mathbin{{}_{\circ}\!\!\wedge} \mathsf{D})$. It then follows that P_4 can be removed from finite expressions except for atoms of the form $\mathsf{P}_4^n(\mathsf{I})$ for $n \in \omega$.

To keep track of the history of a process we will use the minimal history operator H, defined by:

$$\begin{aligned}
H(x) &= H_0(x),\\
H_n(c) &= c \text{ for } c \in A_{t\delta} \cup \{\mu\},\\
H_n(ax) &= a \cdot H_{n+1}(x) \text{ for } a \in A_t,\\
H_n(x+y) &= H_n(x) + H_n(y),\\
H_n(\phi :\rightarrow x) &= H_n(\phi) :\rightarrow H_n(x),\\
H_n(c) &= c \text{ for } c \in \{\mathsf{M}, \mathsf{T}, \mathsf{F}, \mathsf{D}\},\\
H_n(\mathsf{I}) &= n \mathbin{{}_{\circ}\!\!\equiv} 0,\\
H_0(\mathsf{P}_4(\phi)) &= \mathsf{M},\\
H_{n+1}(\mathsf{P}_4(\phi)) &= H_n(\phi),\\
H_n(\neg\phi) &= \neg H_n(\phi),\\
H_n(\phi \wedge \psi) &= H_n(\phi) \wedge H_n(\psi),\\
H_n(\phi \mathbin{{}_{\circ}\!\!\wedge} \psi) &= H_n(\phi) \mathbin{{}_{\circ}\!\!\wedge} H_n(\psi).
\end{aligned}$$

The history operator H keeps track of the number of actions that a process has performed since initialization. As an example, consider

$$\begin{aligned}
\Phi = \;& \mathsf{I}\\
& {}^{\circ}\!\!\vee\\
& (\neg\mathsf{P}_4(\mathsf{I}) \mathbin{{}_{\circ}\!\!\wedge} \mathsf{P}_4^2(\mathsf{I}))\\
& {}^{\circ}\!\!\vee\\
& (\neg\mathsf{P}_4(\mathsf{I}) \mathbin{{}_{\circ}\!\!\wedge} \neg\mathsf{P}_4^2(\mathsf{I}) \mathbin{{}_{\circ}\!\!\wedge} \neg\mathsf{P}_4^3(\mathsf{I}) \mathbin{{}_{\circ}\!\!\wedge} \mathsf{P}_4^4(\mathsf{I})).
\end{aligned}$$

We assume that all communications are δ. Now consider

$$\begin{aligned}
X &= H(X_1 \parallel X_2),\\
X_1 &= (\Phi :\rightarrow a)^2(\Phi :\rightarrow b),\\
X_2 &= (\neg\Phi :\rightarrow c)(\neg\Phi :\rightarrow d),
\end{aligned}$$

where *bounded repetition* $x^n y$ is defined by $x^0 y = y$, and $x^{n+1} y = x \cdot (x^n y)$ [BP97c]. We find that

$$X = acadb.$$

The history operator in cooperation with Φ schedules $X_1 \parallel X_2$ as an alternation of steps, beginning with X_1.

Consider potentially nonterminating processes, which we specify with *, the binary Kleene star [Kle56], and that is defined by

$$x^* y = x(x^* y) + y.$$

(See also [BBP94].) In particular, $x^*\delta$ repeatedly performs x, as follows easily from the axioms (A6) and (A7), and can also be defined by $x^\omega = x(x^\omega)$ (see [Fok97]). An obvious question is how to provide scheduling guards for potentially nonterminating processes. This leads us to infinitary propositions. We give some examples first.

$$\Phi_{even} = \mathsf{I} \mathbin{{}^{\circ}\!\vee} \neg\mathsf{P}_4(\Phi_{even}).$$

Thus Φ_{even} will hold of even step numbers.

$$\Phi_{tr} = \mathsf{I} \mathbin{{}^{\circ}\!\vee} (\neg\mathsf{P}_4(\mathsf{I}) \mathbin{{}_{\circ}\!\wedge} \neg\mathsf{P}_4^2(\mathsf{I}) \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4^3(\Phi_{tr})).$$

So Φ_{tr} will hold if the action history length is a multiple of 3. We notice

$$\begin{aligned} \Phi_{even} &= \mathsf{I} \mathbin{{}^{\circ}\!\vee} \neg\mathsf{P}_4(\mathsf{I} \mathbin{{}^{\circ}\!\vee} \neg\mathsf{P}_4(\mathsf{I} \mathbin{{}^{\circ}\!\vee} \ldots)) \\ &= \mathsf{I} \mathbin{{}^{\circ}\!\vee} (\neg\mathsf{P}_4(\mathsf{I}) \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4^2(\mathsf{I} \mathbin{{}^{\circ}\!\vee} ..)) \\ &= \mathsf{I} \mathbin{{}^{\circ}\!\vee} (\neg\mathsf{P}_4(\mathsf{I}) \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4^2(\mathsf{I}) \mathbin{{}^{\circ}\!\vee} ..)). \end{aligned}$$

It easily follows that

$$H((\Phi_{even} :\rightarrow a)^*\delta \parallel (\neg\Phi_{even} :\rightarrow b)^*\delta) = (ab)^*\delta,$$

and

$$H(((\Phi_{tr} :\rightarrow a)(\Phi_{tr} :\rightarrow b)(\Phi_{tr} :\rightarrow c))^*\delta \parallel ((\neg\Phi_{tr} :\rightarrow d)(\neg\Phi_{tr} :\rightarrow e))^*\delta) = (adebdecde)^*\delta.$$

Due to the form of the recursion equations for Φ_{even} and Φ_{tr}, we find for all $n \in \omega$ that $H_n(\Phi_{even})$ and $H_n(\Phi_{tr})$ have their values in $\{\mathsf{T}, \mathsf{F}\}$.

It is worth looking in general to such recursive equations. First of all we allow only $\mathsf{T}, \mathsf{F}, \mathsf{I}, \neg, \mathbin{{}_{\circ}\!\wedge}, \mathbin{{}^{\circ}\!\vee}, \mathsf{P}_4$ to occur, thus giving these conditions a clean algorithmic content. Consider

$$\Phi = \mathsf{F} \mathbin{{}^{\circ}\!\vee} \Phi.$$

This is just ${}^{\circ}\!\bigvee \lambda n.\mathsf{F}$, so the plausible value for Φ is D (cf. the previous Example 6.1). For simplicity we consider conditions defined by a single recursion equation

$$\Phi = f(\Phi)$$

only, and we assume that $f(\Phi) \not\equiv \Phi$. The interpretation of Φ is

$$\lim_{n\to\infty} f^n(\mathsf{D}),$$

where $f^n(\mathsf{D})$ is for each n a finite proposition equivalent to one built from atoms $\mathsf{P}_4^k(\mathsf{I})$. This interpretation can be motivated by the fact that all operations are monotone in the partial ordering where D is smallest. For each history, i.e., valuation w on the $\mathsf{P}_4^k(\mathsf{I})$ we have $w(f^n(\mathsf{D})) \in \{\mathsf{M}, \mathsf{T}, \mathsf{F}, \mathsf{D}\}$. Furthermore, if $w(f^n(\mathsf{D})) \neq \mathsf{D}$, then $w(f^k(\mathsf{D})) = w(f^n(\mathsf{D}))$ for $k > n$, which proves that the limit always exists. For this to hold with a number of simultaneous defining equations the restriction that only sequential connectives occur is needed:

$$\begin{array}{lclcl} \Phi_1 & = & \mathsf{T} \vee \Phi_2 & = & f_1(\Phi_1, \Phi_2), \\ \Phi_2 & = & \Phi_1 \mathbin{{}_{\circ}\!\wedge} \mathsf{M} & = & f_2(\Phi_1, \Phi_2). \end{array}$$

We find $\vec{f}(\mathsf{D}, \mathsf{D}) = (\mathsf{T}, \mathsf{D})$, $\vec{f}(\mathsf{T}, \mathsf{D}) = (\mathsf{T}, \mathsf{M})$, and finally $\vec{f}(\mathsf{T}, \mathsf{M}) = (\mathsf{M}, \mathsf{M})$, still a limit, but with less algorithmic content since Φ_1 loses the value T.

Let $\Phi = \mathsf{I} \mathbin{{}^{\circ}\!\vee} \mathsf{P}_4(\Phi)$. We notice that within the scope of H, Φ will always evaluate to T as there must be a finite history. Similarly, $\Phi' = \neg\mathsf{I} \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4(\Phi')$ will always evaluate to F, and $\Phi'' = \mathsf{F} \mathbin{{}^{\circ}\!\vee} \mathsf{P}_4(\Phi'')$ will evaluate to M in all histories. The occurrence of D can be avoided if care is taken that all occurrence of recursive calls are in the scope of P_4.

We conclude that MHL (generated using constant I and modality P_4) naturally leads to a four-valued logic. Together with the history operator it can be used to obtain conditions which drastically reduce the number of interleavings introduced by parallel composition.

Example 6.3 (AHL, *Action History Logic*)**.** Action history logic AHL extends MHL with a predicate L on A_t. The atomic condition $\mathsf{L}(a)$ expresses that the last action was a. In case the state is initial, $\mathsf{L}(a)$ evaluates to F. Again, L can be seen as a two-valued variant of L_4 that yields $\mathsf{L}_4(a) = \mathsf{M}$ in the initial state. Then

$$\mathsf{L}(a) = \neg\mathsf{I} \mathbin{{}_{\circ}\!\wedge} \mathsf{L}_4(a).$$

The history operator H will now memorize the last trace (action history) of a process. So for σ ranging over A_t^* and ϵ denoting the empty string,

$$\begin{array}{rcl} H(x) & = & H_\epsilon(x), \\ H_\sigma(c) & = & c \text{ for } c \in A_{t\delta} \cup \{\mu\}, \\ H_\sigma(ax) & = & a \cdot H_{\sigma a}(x) \text{ for } a \in A_t, \\ H_\sigma(x+y) & = & H_\sigma(x) + H_\sigma(y), \\ H_\sigma(\phi :\rightarrow x) & = & H_\sigma(\phi) :\rightarrow H_\sigma(x), \\ H_\sigma(c) & = & c \text{ for } c \in \{\mathsf{M}, \mathsf{T}, \mathsf{F}, \mathsf{D}\}, \\ H_\epsilon(\mathsf{I}) & = & \mathsf{T}, \\ H_{\sigma a}(\mathsf{I}) & = & \mathsf{F}, \\ H_\epsilon(\mathsf{P}_4(\phi)) & = & \mathsf{M}, \\ H_{\sigma a}(\mathsf{P}_4(\phi)) & = & H_\sigma(\phi), \\ H_\sigma(\neg\phi) & = & \neg H_\sigma(\phi), \\ H_\sigma(\phi \wedge \psi) & = & H_\sigma(\phi) \wedge H_\sigma(\psi), \\ H_\sigma(\phi \mathbin{{}_{\circ}\!\wedge} \psi) & = & H_\sigma(\phi) \mathbin{{}_{\circ}\!\wedge} H_\sigma(\psi), \\ H_\epsilon(\mathsf{L}_4(a)) & = & \mathsf{M}, \\ H_{\sigma a}(\mathsf{L}_4(b)) & = & a \equiv b. \end{array}$$

We notice that for all $\sigma \in A_t^*$, $H_\sigma(I) = H_\sigma\left(\bigwedge_{a \in A_t} \neg \mathsf{L}_4(a)\right)$, and if $a \not\equiv b$, then $H_\sigma\left(\mathsf{L}_4(a) \wedge \mathsf{L}_4(b)\right) = \mathsf{F}$. AHL with recursively defined conditions and its semantics is developed just as in the case of MHL. As an example, consider

$$\begin{array}{rcl}
\Phi & = & \mathsf{I} \mathbin{\overset{\circ}{\vee}} \mathsf{L}_4(a), \\
X_1 & = & ((\Phi :\rightarrow c_1)((\Phi :\rightarrow c_2) + (\Phi :\rightarrow c_3)(\Phi :\rightarrow c_4)))^*\, \delta, \\
X_2 & = & ((\neg\Phi :\rightarrow g_1)((\neg\Phi :\rightarrow a) + (\neg\Phi :\rightarrow g_2)(\neg\Phi :\rightarrow g_3)(\neg\Phi :\rightarrow b)))^*\, \delta, \\
X & = & H(X_1 \parallel X_2).
\end{array}$$

We obtain

$$\begin{array}{rcl}
X & = & (c_1 Y (c_2 Y + c_3 Y c_4 Y))^* \delta, \\
Y & = & g_1(a + g_2 g_3 b Y).
\end{array}$$

A condition Φ can be called *static* if it satisfies

$$\Phi :\rightarrow xy = (\Phi :\rightarrow x)(\Phi :\rightarrow y).$$

For dynamic conditions as considered here this equation need not be valid. For instance,

$$(\mathsf{I} :\rightarrow a)(\mathsf{I} :\rightarrow b) = (\mathsf{I} :\rightarrow a) \cdot \delta = \mathsf{I} :\rightarrow a\delta,$$

but $\mathsf{I} :\rightarrow ab = (\mathsf{I} :\rightarrow a)b$. Finally, it is tempting to assert

$$a \cdot X = a \cdot (\mathsf{L}_4(a) :\rightarrow x),$$

but also that does not hold: $H(ab \parallel c) \neq H(a \cdot (\mathsf{L}_4(a) :\rightarrow b) \parallel c)$.

Example 6.4 (FHL, *Floyd-Hoare Logic*)**.** We connect the previous example to Floyd-Hoare logic. The point of this example is to provide a simple set of sound rules for partial correctness assertions, defined with help of left sequential implication. We will not analyze the meta-theory of the proof system however. A partial correctness assertion has syntax

$$\{\phi\}P\{\psi\}$$

where ϕ, ψ are *assertions*, and P is a process expression. An overview of correctness assertions and Floyd-Hoare logic is given in [Apt81]. See [Pon91], and [GP94] for a process algebraic approach.

Let for $\sigma \in A_t^*$ the expression $x \xrightarrow{\sigma} \surd$ denote that x can terminate successfully by performing the actions in σ (under appropriate valuations). The interpretation of correctness assertions is defined as follows:

$$\models \{\phi\}P\{\psi\} \;\text{ if }\; \forall \sigma \in A_t^*,\; H_\sigma(\phi) \;\&\; H_\sigma(x) \xrightarrow{\sigma'} \surd \Longrightarrow H_{\sigma\sigma'}(\psi).$$

We first argue that there is no uniform definition of a *strongest postcondition* of ϕ and P, where a postcondition ψ is *stronger* than postcondition ξ if $\neg(\psi = \xi)$ and for all valuations w,

$$\begin{array}{rcl}
w(\psi) = \mathsf{T} & \Longrightarrow & w(\xi) = \mathsf{T}, \\
w(\xi) = \mathsf{F} & \Longrightarrow & w(\psi) = \mathsf{F}.
\end{array}$$

Consider the correctness assertion

$$\{\phi\}a\{\neg\mathsf{I} \mathbin{{}_{\circ}\!\wedge} \mathsf{L}(a) \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4(\phi)\}.$$

Though it may seem that $\neg\mathsf{I} \mathbin{{}_{\circ}\!\wedge} \mathsf{L}(a) \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4(\phi)$ is here a strongest postcondition of ϕ and a, this is not the case if $\phi = \mathsf{M}$ as the postcondition above equals M, whereas a "strongest postcondition" of M and a is of course F.

In Table 4 we give a simple proof system for deriving partial correctness assertions, based on AHL. Here we use *left sequential implication*, notation $\circ\!\!\rightarrow$, and a three-valued predicate T defined by

$$\begin{aligned} x \circ\!\!\rightarrow y &= \neg x \mathbin{{}^{\circ}\!\vee} y, \\ T(\phi) &= \phi \mathbin{{}^{\circ}\!\vee} \neg\phi, \end{aligned}$$

in order to formulate a weak variant of the rule of consequence $(viii)$. The soundness of the axioms and rules in Table 4 follows straightforward. Finally, note that if $\models \{\mathsf{T}\}x\{\phi\}$, then $H(x(\phi :\rightarrow y)) = H(xy)$.

Table 4: Axioms and rules for Floyd-Hoare Logic.

(i)	$\{\phi\}a\{\neg\mathsf{I} \mathbin{{}_{\circ}\!\wedge} \mathsf{L}(a) \mathbin{{}_{\circ}\!\wedge} \mathsf{P}_4(\phi)\}$	(ii)	$\{\phi\}\delta\{\mathsf{F}\}$
(iii)	$\{\phi\}\mu\{\mathsf{F}\}$	(iv)	$\dfrac{\{\phi\}x\{\psi\} \quad \{\phi\}y\{\psi\}}{\{\phi\}x+y\{\psi\}}$
(v)	$\dfrac{\{\phi\}x\{\psi\} \quad \{\psi\}y\{\xi\}}{\{\phi\}x\cdot y\{\xi\}}$	(vi)	$\dfrac{\{\phi\}x\{\phi\} \quad \{\phi\}y\{\psi\}}{\{\phi\}x^*y\{\psi\}}$
(vii)	$\dfrac{\{\phi \mathbin{{}_{\circ}\!\wedge} \xi\}x\{\psi\}}{\{\phi\}\xi :\rightarrow x\{\psi\}}$	$(viii)$	$\dfrac{\phi \circ\!\!\rightarrow \phi' = T(\phi) \quad \{\phi'\}x\{\psi'\} \quad \psi' \circ\!\!\rightarrow \psi = T(\psi')}{\{\phi\}x\{\psi\}}$

7 Digression

Parameterized Actions and Non-Classical Values. When dealing with actions $a(x)$ parameterized by x over some data type, it makes sense to consider the case in which data can also take values D and M (cf. [BS96]). If so, one faces the question how to interpret $a(\mathsf{D})$ and $a(\mathsf{M})$. Given the preceding interpretation of conditions, a natural choice is to take

$$\begin{aligned} a(\mathsf{M}) &= \mu, \\ a(\mathsf{D}) &= \delta. \end{aligned}$$

(So $a(x)$ is an atomic action for $\mathsf{D} \neq x \neq \mathsf{M}$.)

If ϕ in $x \lhd \phi \rhd y$ or $\phi :\rightarrow x$ can take value M, our process specification features a modeling error. It is vital for the operational meaning that the condition ϕ can be evaluated whenever needed. The value D is an acceptable consequence of evaluation, M however is an outright error. That jeopardizes the operational understanding of the conditional construct and guarded command in process algebra, whence $x \lhd \mathsf{M} \rhd y = \mathsf{M} :\to x = \mu$.

The only reasonable way in which $a(\mathsf{M})$ can occur is in $\mathsf{F} :\to a(\mathsf{M})$ or in $\mathsf{D} :\to a(\mathsf{M})$. In these contexts the guard prevents the effect of M to become "visible" in a process. So the appropriate style of dealing with M is using expressions $\phi(\vec{x}) :\to a(f(\vec{x}))$ with $\phi(\vec{x}) \in \{\mathsf{F}, \mathsf{D}\}$ whenever $f(\vec{x}) = \mathsf{M}$. Occurrence of $a(\mathsf{D})$ is of course less problematic.

Process Algebra with Three-Valued Logics. As far as we can see, at least two reasonable proposals for the restriction to three-valued logic can be made:

$$\mathbb{K}_3(\mathbb{P}) = \Sigma_3(\mathbb{P}, \neg, \wedge)$$

which has its values in $\mathbb{T}_3^D = \mathbb{T}_4 \setminus \{\mathsf{M}\}$. This is Kleene's logic [Kle38], for which a complete axiomatization follows from [Kal58] (see also [BBR95]). This logic can be combined in a straightforward way with ACP, and has its own right of existence, as is argued in [BP97b]. Since $\mathbb{K}_3(\mathbb{P})$ has a complete axiomatization, we do not need the rule of equivalence, and consider $\_ :\to \_$ as a binary operation. Typically, the principle of the excluded middle—*tertium non datur*—is not a tautology anymore, but otherwise very little changes. The adapted version of $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$ is $\mathrm{ACP}_{\mathsf{D}}(A_t, \gamma, \mathbb{P})$, which is obtained by leaving out all axioms that make reference to M and μ, and exchanging (GCL4), i.e.,

$$\phi :\to (\psi :\to x) = \phi \mathbin{{}_{\circ}\!\wedge} \psi :\to x,$$

with its symmetric counterpart (GC4), i.e.,

$$\phi :\to (\psi :\to x) = \phi \wedge \psi :\to x.$$

The system $\mathrm{ACP}_{\mathsf{D}}(A_t, \gamma, \mathbb{P})$ is suited to handle algorithms/programs that may contain divergencies. The associated operational semantics and completeness proof are obtained by leaving out μ and the meaningless predicate $\mu(w, \_)$. For a completeness proof, note that all transformations on basic terms underlying the completeness proof of Theorem 5.5 only use the constants D and T. A detailed completeness proof is given in [BP97b]. An extension of this setting is obtained by involving μ. Also in this case completeness is preserved[3]. Notice that both $\mathrm{ACP}_{\mathsf{D}}(A_t, \gamma, \mathbb{P})$ and its extension with μ can be characterized by the (derivable) identity

$$x = x + x \lhd \phi \rhd x,$$

which does not hold in $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$ (set $\phi = \mathsf{M}$).

A second 'reasonable' three-valued logic emerges from a combination of Bochvar's strict three-valued logic [Boc39] and McCarthy's sequential three-valued logic[4] [McC63]:

$$\mathbb{BM} = \Sigma_3(\mathbb{P}, \neg, \wedge, \mathbin{{}_{\circ}\!\wedge})$$

which has its values in $\mathbb{T}_3^M = \mathbb{T}_4 \setminus \{\mathsf{D}\}$. In [BP97a] we argue that the combination of this logic and process algebra is suited to analyze concurrent process expressions in which meaningless can occur. The adapted version of the rule of equivalence is

$$(\mathrm{ROE}_3^M) \quad \frac{\mathbb{BM}(\mathbb{P}) \models \phi = \psi}{\mathrm{ACP}_{\mathsf{M},\mu}(A_t, \gamma, \mathbb{P}) \vdash \phi :\to x = \psi :\to x}$$

[3] In this case, the definition of a basic term should be adapted: also $\phi :\to \mu$ is one. Furthermore, height $h(\mu) = 0$. Finally, in the proof of the adaptation of Lemma 5.4, the "Case $h = 0$" should also cover $P_n \equiv \phi_n :\to \mu$.

[4] In fact we combine both variants of McCarthy's logic, i.e., left sequential and right sequential, which are distinguished by the use of asymmetric symbols for the connectives.

where $\mathrm{ACP}_{\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$ is obtained by leaving out D from $\mathrm{ACP}_{\mathsf{D},\mathsf{M},\mu}(A_t, \gamma, \mathbb{P})$. Also in this case, the completeness result is preserved: we can restrict Lemma 5.3 to properties $(i) - (iii)$. Furthermore, in Lemma 5.4, the "$h = 0$"-case can be proved by the following observation: $w(\phi_1) = \mathsf{M} \Leftrightarrow w(\phi_2) = \mathsf{M}$ implies that $\mathbb{BM}(\mathbb{P}) \models \phi_1 \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F} = \phi_2 \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F}$.

References

[Apt81] K.R. Apt. Ten years of Hoare's logic, a survey, part I. *ACM Transactions on Programming Languages and Systems*, 3(4):431–483, 1981.

[BB90] J.C.M. Baeten and J.A. Bergstra. Process algebra with signals and conditions. In M. Broy, editor, Programming and Mathematical Method, *Proceedings Summer School Marktoberdorf*, 1990 NATO ASI Series F, pages 273-323, Springer-Verlag, 1992.

[BBK86] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, IX(2):127-168, 1986.

[BBP94] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. *Computer Journal*, 37(4):243–258, 1994.

[BBR95] J.A. Bergstra, I. Bethke, and P.H. Rodenburg. A propositional logic with 4 values: true, false, divergent and meaningless. *Journal of Applied Non-Classical Logics*, 5:199-217, 1995.

[BCJ84] H. Barringer, J.H. Cheng, and C.B. Jones. A logic covering undefinedness in program proofs. *Acta Informatica*, 21:251–269, 1984.

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560-599, 1984.

[BK84a] J.A. Bergstra and J.W. Klop. The algebra of recursively defined processes and the algebra of regular processes. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, *Algebra of Communicating Processes, Utrecht 1994*, Workshops in Computing, pages 1-25. Springer-Verlag, 1995. An extended abstract appeared in J. Paredaens, editor, *Proceedings* 11^{th} *ICALP*, Antwerp, volume 172 of *Lecture Notes in Computer Science*, pages 82-95. Springer-Verlag, 1984.

[BK84b] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Computation*, 60(1/3):109-137, 1984.

[Boc39] D.A. Bochvar. On a 3-valued logical calculus and its application to the analysis of contradictions (in Russian). *Matématičeskij sbornik*, 4:287–308, 1939.

[BP96] J.A. Bergstra and J.C. van de Pol. A calculus for sequential logic with 4 values. Technical Report 160, Logic Group Preprint Series, Utrecht University, 1996.

[BP97a] J.A. Bergstra and A. Ponse. Bochvar-McCarthy logic and process algebra. Technical Report P9722, Programming Research Group, University of Amsterdam, 1997.

[BP97b] J.A. Bergstra and A. Ponse. Kleene's three-valued logic and process algebra. Technical Report P9723, Programming Research Group, University of Amsterdam, 1997.

[BP97c] J.A. Bergstra and A. Ponse. Process algebra primitives for file transfer. Liber Amicorum dedicated to Paul Klint, pages 33–42, CWI, 1997.

[Bur84] J.P. Burgess. Basic tense logic. In D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic, Vol. II*, pages 89–133, Reidel, 1984.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra*. Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.

[BS96] J.A. Bergstra and M.P.A. Sellink. Sequential data algebra primitives. Technical Report P9602b, Programming Research Group, University of Amsterdam, 1996.

[Dij76] E.W. Dijkstra. *A Discipline of Programming*. Prentice Hall International, Englewood Cliff, 1976.

[FG96] W.J. Fokkink and R.J. van Glabbeek. Ntyft/ntyxt rules reduce to ntree rules. *Information and Computation*, 126:1–10, 1996.

[Fok97] W.J. Fokkink. Axiomatizations for the perpetual loop in process algebra. In P. Degano, R. Gorrieri and A. Marchetti-Spaccamela (eds.), *Proc. 24th Colloquium on Automata, Languages and Programming - ICALP'97*, Lecture Notes in Computer Science 1256, pages 571–581, Springer-Verlag, 1997.

[Gro93] J.F. Groote. Transition system specifications with negative premises. *Theoretical Computer Science*, 118(2):263–299, 1993.

[GP94] J.F. Groote and A. Ponse. Process algebra with guards: combining Hoare logic and process algebra. *Formal Aspects of Computing*, 6:115–164, 1994.

[HHJ+87] C.A.R. Hoare, I.J. Hayes, He Jifeng, C.C. Morgan, A.W. Roscoe, J.W. Sanders, I.H. Sorensen, J.M. Spivey, and B.A. Sufrin. Laws of programming. *Communications of the ACM*, 30(8):672-686, August 1987.

[JM94] C.B. Jones, C.A. Middelburg. A typed logic of partial functions reconstructed classically. *Acta Informatica*, 31(5):399-430, 1994.

[Jon90] C.B. Jones. *Systematic Software Development using VDM* (2nd edition). Prentice-Hall International, Englewood Cliffs, 1990.

[Kal58] J. Kalman. Lattices with involution. *Trans. Am. Math. Soc.*, 87:485–491, 1958.

[Kle38] S.C. Kleene. On a notation for ordinal numbers. *Journal of Symbolic Logic*, 3:150-155, 1938.

[Kle56] S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, Princeton NJ, 1956.

[McC63] J. McCarthy. A basis for a mathematical theory of computation. In P. Braffort and D. Hirshberg (eds.), *Computer Programming and Formal Systems*, pages 33-70, North-Holland, Amsterdam, 1963.

[Pon91] A. Ponse. Process expressions and Hoare's logic. *Information and Computation* 95(2):192–217, 1991.

[Rod96] P.H. Rodenburg. A complete system of four-valued logic. Technical Report P9616, Programming Research Group, University of Amsterdam, 1996.

[Ver95] C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. *Nordic Journal of Computing*, 2(2):274-302, 1995.