Process algebra and conditional composition

Jan A. Bergstra a,b,∗, Alban Ponse a,c

a University of Amsterdam, Programming Research Group, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
b Utrecht University, Department of Philosophy, Heidelberlaan 8, 3584 CS Utrecht, The Netherlands
c CWI, Department of Software Engineering, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Received 21 July 2000; received in revised form 30 December 2000

Abstract

We discern three non-classical truth values, and define a five-valued propositional logic. We combine this logic with process algebra via conditional composition (i.e., if-then-else-). In particular, the choice operation (+) is regarded as a special case of conditional composition. We present an operational semantics in SOS-style and some completeness results. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Process algebra; Conditional composition; Five-valued logic

1. Introduction

Assume P represents some program (or algorithm). Then the initial behavior of the conditional program

\[ \text{if } \phi \text{ then } P \text{ else } P \]

depends on evaluation of the condition \( \phi \): either it yields an immediate error, or it starts with performing \( P \), or it diverges in evaluation of \( \phi \). The following three non-classical truth values for \( \phi \) are sufficient to accommodate these intuitions:

- **Meaningless**, notation M. Typical examples are errors that are detectable during execution such as a type-clash or division by zero.
- **Choice or undetermined**, notation C. This value represents ‘being either true or false’. An example is as above: if \( \phi \) then \( P \) else \( P \) represents the same behavior as \( P \).
- **Divergent or undefined**, notation D. Typically, evaluation of a partial predicate can diverge.

We describe a five-valued propositional logic that incorporates these three non-classical truth values next to true (notation T) and false (notation F).

Furthermore, we define a generalization of process algebra that is based on conditional composition over this logic.

This paper is a successor of [6], in which ACP with five-valued conditions is introduced. In Section 5 we discuss the main differences with [6].

2. Five-valued logic

The five truth values discerned above can be arranged in the partial ordering given in Fig. 1. Let \( x \cup y \) stand for the least upper bound of \( x \) and \( y \). So, \( T \cup F = F \cup T = C \), and \( x \cup y \in \{ x, y \} \) for all other pairs. Furthermore, each truth value can be described with \( \cup \) and the deterministic truth values M, T, F and D.
We first consider a single, ternary operation on these five truth values: conditional composition, notation \( x \triangleleft y \triangleright z \) (this notation stems from [10], modeling \( if \ y \ then \ x \ else \ z \)). Conditional composition is defined as follows:

\[
\begin{align*}
x \triangleleft M & \triangleright y = M, \\
x \triangleleft T & \triangleright y = x \lor y, \\
x \triangleleft F & \triangleright y = y, \\
x \triangleleft D & \triangleright y = D.
\end{align*}
\]

Notice that \( x \triangleleft C \triangleright y \) (as a binary operation) is idempotent, commutative, and associative. Furthermore, we have the following convenient distributivity property:

**Proposition 1.** Conditional composition distributes over \( \lor \); let \( v \) abbreviate \( v_1 \lor v_2 \), then

\[
\begin{align*}
x \triangleleft y \triangleright z & = (x_1 \triangleleft y \triangleright z) \lor (x_2 \triangleleft y \triangleright z) \\
& = (x \triangleleft y_1 \triangleright z) \lor (x \triangleleft y_2 \triangleright z) \\
& = (x \triangleleft y \triangleright z_1) \lor (x \triangleleft y \triangleright z_2).
\end{align*}
\]

As a consequence, conditional composition is monotonic.

Next to conditional composition, we consider the following logical operations (cf. [2,6]): negation, left-sequential conjunction and symmetric (or strict parallel) conjunction. Negation on the newly added non-classical values can be explained from the intuitions provided earlier: \( \neg M = M \) because the negation of an immediate error is one as well. Since \( C \) means “being either true or false”, so does its negation, thus \( \neg C = C \). Furthermore, as \( D \) represents divergence, so does \( \neg D \), hence \( \neg D = D \). With \( \triangleleft \) we denote left-sequential conjunction, i.e., McCarthy’s left to right conjunction [12], adopting the asymmetric notation from [2]. First the left argument is evaluated, and depending on the result of this, possibly the right argument. This yields \( x \triangleleft y = x \) for \( x \in \{ M, F, D \} \), and \( T \triangleleft x = x \). The values of \( C \triangleleft x \) are given below. Finally, symmetric conjunction on the newly added truth values appears to be captured by

\[
x \land y = (x \triangleleft y) \lor (y \triangleleft x).
\]

*Left sequential disjunction, notation \( \triangledown \), and symmetric disjunction (\( \lor \)) are defined as expected:

\[
x \triangledown y = \neg (\neg x \land \neg y), \\
x \lor y = \neg (\neg x \land \neg y).
\]

The complete truth tables for \( \neg, \triangleleft, \) and \( \land \) are the following:

<table>
<thead>
<tr>
<th>( \neg )</th>
<th>( M )</th>
<th>( C )</th>
<th>( T )</th>
<th>( F )</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \neg )</td>
<td>( M )</td>
<td>( C )</td>
<td>( T )</td>
<td>( F )</td>
<td>( D )</td>
</tr>
</tbody>
</table>

These truth tables were also presented in [6], and, when omitting \( C \), coincide with the definitions given in [2]. Note that \( \triangleleft \) and its dual \( \triangledown \) are idempotent and associative.

In the following we establish the relation between conditional composition and the operations just discussed.

**Proposition 2.** The operations \( \neg, \triangleleft \) and \( \land \) are definable from conditional composition:

\[
\begin{align*}
\neg x & = F \triangleleft x \triangleright T, \\
x \triangleleft y & = x \triangleleft x \triangleright F, \\
x \land y & = (x \triangleleft y) \triangleleft C \triangleright (y \triangleleft x).
\end{align*}
\]

Furthermore,

\[
\begin{align*}
x \triangleleft y \triangleright z & = z \triangleleft \neg y \triangleright x, \\
\neg (x \triangleleft y \triangleright z) & = \neg x \triangleleft y \triangleright \neg z.
\end{align*}
\]
Corollary 3. The operations \( \lor \) and \( \lor \) can be defined by:

\[
x \lor \lor y = T \triangleleft x \triangleright y,
\]

\[
x \lor y = (x \lor y) \triangleleft G \triangleright (y \lor x).
\]

Furthermore, \( \neg \), \( \land \), \( \lor \) and \( \lor \) distribute over \( \Delta \)
and all these operations are monotonic.

Conversely, \( x \triangleleft C \triangleright y \) can be defined by \((C \land x) \lor (C \land y) \lor (x \land y)\). This leads to the following result:

Proposition 4. Conditional composition \( x \triangleleft y \triangleright z \) can be defined from \( \neg \), \( \land \), and \( \lor \) by

\[
x \triangleleft y \triangleright z = E \triangleleft C \triangleright F,
\]

where \( x \triangleleft C \triangleright y \) is given above, and

\[
E = (y \lor D) \land (x \lor G),
\]

\[
F = (\neg y \lor D) \land (z \lor H),
\]

\[
G = (y \land x) \lor (\neg y \land z),
\]

\[
H = (\neg y \lor x) \land (y \lor z).
\]

We denote the resulting five-valued logic by

\[
L_5(-, \land, \lor),
\]

or shortly \( L_5 \) whenever we do not care which operations are considered primitive.

Following McCarthy and Hayes [13], let \( f, g, \ldots \) be names for fluents, i.e., objects that in any state (i.e., at each instance of time) may take a deterministic value, thus a value in \([M, T, F, D]\). Let \( p_4 \) be a set of fluents. We write \( L_5(p_4) \) for the extension of \( L_5 \) with the fluents in \( p_4 \). In order to equate propositions in \( L_5(p_4) \) we use substitution of fluents: for \( f, g \in p_4 \),

\[
\phi/f \phi, \quad \phi/g \phi = \phi,
\]

\[
\phi/f \phi c = \phi c \quad \text{for } c \in \{M, C, T, F, D\},
\]

\[
\phi/f \psi_1 < \phi/f \psi_2 \triangleright \psi_3 \Delta \psi_1 < \phi \psi_2 \triangleright \psi_3,
\]

and as a proof rule the excluded fifth rule (cf. [5]):

\[
\phi/f \phi = \phi \psi \quad \text{for } c \in \{M, T, F, D\}.
\]

By Proposition 2 it follows that substitution distributes over the other logical operations in the expected way.

Together with the identities generated by the truth tables this yields a complete evaluation system for equations over \( L_5(p_4) \). We write \( L_5(p_4) \models \phi = \psi \) or shortly \( \models \phi = \psi \), if \( \phi = \psi \) follows from the system defined above and the truth tables for \( L_5(p_4) \).

3. A generalization of BPA with five-valued conditions

Let \( A \) be a set of constants \( a, b, c, \ldots \) denoting atomic actions (atoms), i.e., processes that are not subject to further division, and that execute in finite time. We consider a generalized version of \( BPA_{δ,μ}(A) \), i.e., Basic Process Algebra (see, e.g., \([3,1,8]\)) extended with \( δ \notin A \) (inaction or deadlock) and with \( μ \notin A \). The meaningless process \( μ \) represents the operational contents of \( M \), and is introduced in \([4,5]\). We use the notation \( G_{L_5(p_4)}(BPA_{δ,μ}(A)) \) for a generalization of \( BPA_{δ,μ}(A) \) in which alternative composition is a special case of conditional composition over \( L_5(p_4) \) (various other generalizations are conceivable). The operations of \( G_{L_5(p_4)}(BPA_{δ,μ}(A)) \) are:

**Sequential composition:** \( X \cdot Y \) denotes the process that performs \( X \), and upon completion of \( X \) starts with \( Y \).

**Conditional composition:** \( X +_δ Y \) with \( φ \in L_5(p_4) \) denotes the process that either performs \( X \) or \( Y \), or represents \( δ \) or \( μ \), depending on the value of \( φ \) (which may depend on some valuation).

(Conditional composition \( X +_δ Y \) is often denoted \( X <_φ \triangleright Y \), cf. \([10]\).)

We mostly suppress the \( \cdot \) in process expressions, and brackets according to the rule that \( \cdot \) binds strongest. Accommodating to classical process algebra, we shall often use the *abbreviation* \( +_c \) for \( +_c \) (modeling 'alternative composition' or 'choice'), thus \( X + Y \) is short for \( X +_c Y \).

In Table 1 we give the rule of equivalence (ROE) and the axioms of \( G_{L_5(p_4)}(BPA_{δ,μ}(A)) \).

**Example 5.** In \( G_{L_5(p_4)}(BPA_{δ,μ}(A)) \) one easily derives

\[
X + δ = X,
\]

\[
X + X = X,
\]

\[
X + μ = μ.
\]
A special case of conditional composition:

Example 6. With the axioms CC, GA2, and P semantics for process terms. Given a (non-empty) set called process terms φ.

(Use axiom GA3 with 44 J.A. Bergstra, A. Ponse / Information Processing Letters 80 (2001) 41–49 we can apply the rule of equivalence (ROE).

If expressing a guarded command (called X the intricate identity is X + (φ ⊾ Y) = (X + φ) + Y.

Because w = (φ ⊾ X) ⊾ F = (φ ⊾ X) ⊾ C ⊾ F, we can apply the rule of equivalence (ROE).

Closed terms over \( G_{\text{LT}(\mathbb{P}_d)}(\text{BPA}_\phi(A)) \) will be further called process terms. We provide an operational semantics for process terms. Given a (non-empty) set \( \mathbb{P}_d \) of fluents, let \( w \) range over \( \mathcal{W} \), the valuations (interpretations) of \( \mathbb{P}_d \) in \( \{M, T, F, D\} \). Valuations are extended to propositions in the usual way. In Table 2 we define for each \( w \in \mathcal{W} \) a unary predicate meaningless, notation \( \mu(w, \_ \_) \), over process terms in \( G_{\text{LT}(\mathbb{P}_d)}(\text{BPA}_\phi(A)) \). This predicate defines whether a process term represents the meaningless process \( \mu \) under valuation \( w \).

The axioms and rules for \( \mu(w, \_ \_) \) given in Table 2 are extended by those given in Table 3, which define transitions \( \_ \to \_ \) as a binary relation on process terms, and unary “tick-predicates” or “termination transitions” \( \_ \\triangleleft \_ \) over \( A \). Transitions characterize under which interpretations a process term defines the possibility to execute an atomic action, and what remains to be executed (if anything, otherwise \( \triangleright \) symbolizes successful termination). Note that if a process term \( P \) has a transition \( P \to \_ \_ \_ \_ \_ \_ \to \_ \_ \_ \_ \_ \) then \( \mu(w, P) \).

The axioms and rules in Tables 2 and 3 yield a structured operational semantics (SOS) with negative premises in the style of [9]. Moreover, they satisfy the so called panth-format [15]. Using [9,15], it is easy to establish that the meaningless instances and transitions defined by these rules are uniquely determined, and go with the following notion of bisimulation equivalence:
Definition 7. A binary relation $B$ over process terms is a \textit{bisimulation} if for all $P$, $Q$ with $PBQ$ the following conditions hold for all $w \in \mathcal{W}$ and $a \in A$:

- $\mu(w, P) \leftrightarrow \mu(w, Q)$,
- $P \xrightarrow{w,a} \top \Leftrightarrow Q \xrightarrow{w,a} \top$,
- $\forall P' (P \xrightarrow{w,a} P' \Rightarrow \exists Q'(Q \xrightarrow{w,a} Q' \land P'BQ'))$,
- $\forall Q' (Q \xrightarrow{w,a} Q' \Rightarrow \exists P'(P \xrightarrow{w,a} P' \land P'BQ'))$.

Two processes $P$, $Q$ are \textit{bisimilar}, notation $P \equiv Q$, if there exists a bisimulation containing the pair $(P, Q)$.

By the main result in [15] it follows that bisimilarity is a \textit{congruence} relation for all operations involved. Note that conditional composition constructs are considered binary operations: for each $\phi \in \mathcal{L}_5(\mathcal{P}_4)$ there is an operation $+_\phi$.

We write $G \models P = Q$ whenever $P \equiv Q$ according to the notions just defined, and for $X = X_1, \ldots, X_n$, $G \models t(\vec{X}) = t'(\vec{X})$ if for all $\vec{P} = P_1, \ldots, P_n$ it holds that $G \models t(\vec{P}) = t'(\vec{P})$. It is not difficult to show that in the bisimulation model thus obtained all equations of Table 1 are true. Hence we conclude:

\textbf{Lemma 8} (Soundness). If $G_{\mathcal{L}_5(\mathcal{P}_4)}(\mathcal{BPA}_{\delta,\mu}(A)) \models t(\vec{X}) = t'(\vec{X})$, then $G_{\mathcal{L}_5(\mathcal{P}_4)} \models t(\vec{X}) = t'(\vec{X})$.

Finally, we provide a completeness result for $G_{\mathcal{L}_5(\mathcal{P}_4)}(\mathcal{BPA}_{\delta,\mu}(A))$. Our proof refers to the completeness result in [5], which is based on a representation of process terms for which bisimilarity implies derivability in a straightforward way.

\textbf{Definition 9.} A process term $P$ over $G_{\mathcal{L}_5(\mathcal{P}_4)}(\mathcal{BPA}_{\delta,\mu}(A))$ is a \textit{generalized basic term} if it is of the form

$P ::= \delta \mid a \mid aP \mid P +_\phi P$,

where $a \in A$ and $\phi \in \mathcal{L}_5(\mathcal{P}_4)$.

\textbf{Lemma 10.} Each process term over $G_{\mathcal{L}_5(\mathcal{P}_4)}(\mathcal{BPA}_{\delta,\mu}(A))$ is provably equal to a generalized basic term.

In the following we relate process terms over $G_{\mathcal{L}_5(\mathcal{P}_4)}(\mathcal{BPA}_{\delta,\mu}(A))$ with terms over $\mathcal{BPA}_{\delta,\mu}(A)$ extended with conditional guard constructs, of which the conditions are in

$\mathcal{L}_4(\mathcal{P}_4) \triangleq \mathcal{L}_{\{M,T,F,D\}}(\mathcal{P}_4, \neg, \land, \lor)$,

thus $\mathcal{L}_5(\mathcal{P}_4)$ without $G$. The only operations of $\mathcal{BPA}_{\delta,\mu}(A)$ are sequential composition and the choice operation $+$, i.e., the operation $+_c$. In the following, finite sums $P_1 + P_2 + \cdots + P_n$ are abbreviated by $\sum_{i=1}^{n} P_i$.

Let the symbol $\equiv$ denote syntactic equivalence, and let $\mathcal{L} \subseteq \mathcal{L}_5(\mathcal{P}_4)$.

\textbf{Definition 11.} A process term $P$ over $G_{\mathcal{L}_5(\mathcal{P}_4)}(\mathcal{BPA}_{\delta,\mu}(A))$ is called an $\mathcal{L}$-\textit{basic term} if $P$ \equiv

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
$a \in A$ & $a \xrightarrow{w,a}$ & \\hline
$X \xrightarrow{w,a} \top$, $\neg \mu(w, Y)$ & $X \xrightarrow{w,a} X'$, $\neg \mu(w, Y)$ & \\hline
\end{tabular}
\caption{Transition rules in \textit{panth}-format}
\end{table}
\[ \sum_{i \in I} \phi_i \rightarrow Q_i, \text{ where } I \text{ is a finite, non-empty index set, } \phi_i \in \mathcal{L}, \text{ and } Q_i \in \{ \delta, a, aR | a \in A \}. \]

**Lemma 12.** Each process term over \( G_{\mathcal{L}}(\text{BPA}_{\delta, \mu}(A)) \) is provably equal to an \( \mathcal{L}_4(\mathbb{P}_4) \)-basic term.

**Proof.** By Lemma 10 it is sufficient to consider generalized basic terms. Then, representation easily follows for \( \mathcal{L}_4(\mathbb{P}_4) \)-basic terms by induction (where axiom CC is needed, cf. footnotes 5, 6 in Section 5). It remains to be shown that each \( \mathcal{L}_5(\mathbb{P}_4) \)-basic term is provably equal to one in which \( \mathcal{C} \) does not occur in any conditional guard construct. As \( \mathcal{C} \rightarrow X = X \), this follows easily by induction on the complexity of the guard \( \phi \) in \( \phi \rightarrow X \). \( \square \)

**Theorem 13.** The system \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{BPA}_{\delta, \mu}(A)) \) is complete with respect to bisimulation equivalence.

**Proof.** By Lemmas 12 and 8 it is sufficient to prove that bisimilarity between \( \mathcal{L}_4(\mathbb{P}_4) \)-basic terms implies their provable equality. A detailed (inductive) proof is spelled out in [5], which is also sufficient as all axioms of Basic Process Algebra with four-valued logic are derivable from \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{BPA}_{\delta, \mu}(A)) \) (the less trivial ones were derived in Examples 5 and 6). \( \square \)

### 4. A generalization of ACP with five-valued conditions

We extend \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{BPA}_{\delta, \mu}(A)) \) to a generalized version of \( \text{ACP}(A, |) \) (Algebra of Communicating Processes, see, e.g., [3,1,8]) by including encapsulation and parametrized merge operations \( \phi \circ \psi \). In the latter, \( \phi \) covers the choice between interleaving and synchronization, and \( \psi \) determines the order of interleaving and synchronization:

- **Parametrized merge:** \( X \parallel \phi \parallel \psi Y \) denotes the parallel execution of \( X \) and \( Y \) under conditions \( \phi \) and \( \psi \).

- **Parametrized left merge,** an auxiliary operator: \( X \parallel \psi Y \) denotes \( X \parallel \psi Y \) with the restriction that the first action is a synchronization of both \( X \) and \( Y \).

- **Parametrized communication merge,** an auxiliary operator: \( X \circ | \psi Y \) denotes \( X \circ | \psi Y \) with the restriction that the first action is a synchronization of both \( X \) and \( Y \).

**Encapsulation:** \( \partial_H(X) \) (where \( H \subseteq A \)) renames atoms in \( H \) to \( \delta \).

In \( \text{ACP}(A, |) \), the commutative and associative communication function \( \parallel: A \times A \rightarrow A \cup \{\delta\} \) is given (and extended to process terms). The axioms of our generalization of \( \text{ACP}(A, |) \) are those of \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{BPA}_{\delta, \mu}(A)) \) (including ROE) and those in Table 4. We adopt the convention that \( +_\phi \) binds weakest and \( \cdot \) binds strongest, and denote the resulting system by \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{ACP}(A, |)) \). We note that the \( \parallel \) operation of \( \text{ACP}(A, |) \) equals \( \parallel_\mathcal{C} \). Furthermore, the operation \( \parallel_\mathcal{C} \) restricts \( \parallel \) to interleaving only, while \( \parallel_\mathcal{C} \) for \( \circ \in \{C, T, F\} \) defines “synchronous ACP” and \( \parallel_\mathcal{E} \) represents sequential composition. Some typical \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{ACP}(A, |)) \) identities are:

- \( X \parallel_\mathcal{C} Y = Y \parallel_\mathcal{C} X \),
- \( X \parallel_\mathcal{C} \phi Y = Y \parallel_\mathcal{C} \phi X \),
- \( \mu \parallel_\mathcal{C} \delta = \mu +_\mathcal{C} \delta \),
- \( \mu \parallel_\mathcal{C} a = \mu +_\mathcal{C} \mu \) (\( a \in A \)),
- \( \delta \parallel_\mathcal{C} a = \delta +_\mathcal{C} \delta \) (\( a \in A \)).

In Table 5 we give additional rules for the meaningless predicate defined in Table 2 and the transition rules defined in Table 3. We stick to bisimulation equivalence as defined in Definition 7, and as before it follows that bisimilarity is a congruence for all operations involved. It is not difficult (but tedious) to establish that in the bisimulation model thus obtained all equations of Table 4 are true. Furthermore, each process term over \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{ACP}(A, |)) \) is provably equal to a generalized basic term (see Definition 9). Hence:

**Theorem 14.** The system \( G_{\mathcal{L}}(\mathbb{P}_4)(\text{ACP}(A, |)) \) is complete with respect to bisimulation equivalence.

### 5. Conclusions

In this paper we have shown that process algebra can be viewed from a logical perspective that comprises the truth values *choice* \( \mathcal{C} \) and *divergent* \( \delta \).
D, and the basic operations conditional composition and sequential composition. For instance, the axiom $X +_a Y = \delta$ expresses that $\delta$ is associated with “divergence”. This may seem incompatible with the usual “deadlock” interpretation (modeled by the standard axioms $X + \delta = X$ and $\delta X = \delta$), but can be clarified as follows: in order to support an axiomatic approach to the interleaving hypothesis, 4 the operation $+$ models “optimistic choice” in the sense that $\delta$-alternatives are discarded ($X + \delta = X$). E.g., the derivation $a(b\delta + \delta b) + \delta ab = ab\delta$ shows that $\delta$ has an aspect of divergence: the deadlock in $ab\delta$ is postponed until all concurrent behavior has been executed.

In the following we shortly discuss the main differences between this paper and [6]. Taking four-valued logic over $\{M, T, F, D\}$ [2,14] and its combination with process algebra [5] as a point of departure, the contribution of [6] can be characterized as follows:

- The introduction of conditional composition as a definable operation in $L_5(\mathcal{P}_4)$.
- An axiomatization of $ACP(A, |)$ with conditional guard construct over $L_5(\mathcal{P}_4)$, going with an operational semantics and a completeness result.
- A generalization of $ACP(A, |)$: the $+$ and merge operators can be parameterized with propositions over $L_5(\mathcal{P}_4)$ (or one of its sublogics containing $\mathbb{G}$, $\mathbb{T}$ and $\mathbb{F}$). 4

The present paper records a non-trivial extension of our understanding of $L_5(\mathcal{P}_4)$, and of its combination with process algebra:

- We show that $L_5(\mathcal{P}_4, \lor, \land, \land)$ and $L_5(\mathcal{P}_4, \leadsto)$ are interdefinable (Propositions 2 and 4).
- We provide operational semantics and (ground complete) axiomatizations for our $L_5(\mathcal{P}_4)$-generalizations of $\mathbb{BPA}_5(A)$ and $ACP(A, |)$ (Theorems 13 and 14). 5

Inspired by one of the referees, we end with some considerations about a six-valued logic. By symmetry

\begin{table}
| Additional axioms of $G_L(\mathcal{P}_4)$ ($ACP(A, |)$), $a, b, c \in A$ and $H \subseteq A$ |
|---------------------------------------------------------------|
| (C1) $a \mid b = b \mid a$ | (GD1) $\delta_H(a) = a$ if $a \notin H$ |
| (C2) $(a \mid b) \mid c = a \mid (b \mid c)$ | (GD2) $\delta_H(a) = \delta$ if $a \in H$ |
| (GD3) $\delta_H(X +_a Y) = \delta_H(X) +_a \delta_H(Y)$ | (GD4) $\delta_H(XY) = \delta_H(X)\delta_H(Y)$ |
| (GCM1) $X \parallel Y = (X \upharpoonright \upharpoonright Y +_a Y \upharpoonright \upharpoonright X) +_a X \upharpoonright Y$ |
| (GCM2) $a \parallel \bot X = aX$ |
| (GCM3) $aX \parallel \bot Y = a(X \parallel \bot Y)$ |
| (GCM4) $(X +_a Y) \parallel \bot Z = X \parallel \bot Y +_a Y \parallel \bot X \parallel \bot Z$ |
| (GCMC) $X \parallel \bot Y = X \parallel \bot Y +_a Y \parallel \bot X$ |
| (GCM5) $aX \parallel \bot Y = a \parallel \bot (Y \parallel \bot X)$ |
| (GCM6) $a \parallel \bot b = a \mid b$ |
| (GCM7) $a \parallel \bot bX = (a \mid b)X$ |
| (GCM8) $a \parallel \bot (X +_a Y) = a \parallel \bot X +_a a \parallel \bot Y$ |
| (GCM9) $(X +_a Y) \parallel \bot Z = X \parallel \bot Z +_a Y \parallel \bot Z$ |

1 I.e., concurrency can be analyzed in terms of all possible interleavings.
2 This establishes a second intuition for Kleene’s third truth value (D modeling the first). We note that a complete axiomatization of Kleene’s three-valued logic [11] admits exactly two non-classical truth values, the conjunction of which must equal $F$.
3 These axioms are derivable in $G_L(\mathcal{P}_4)$ ($ACP(A, |)$).
4 For the case $\mathbb{G}$, $\mathbb{T}$, $\mathbb{F}$ we give the axioms. For the proposed generalization to $L_5(\mathcal{P}_4)$ it must be required that $a \neq \delta$ in axiom GCM8.
5 In [6] we have a less general version of axiom GA3, which requires separate axioms $X + \delta = X$ and $X + \mu = \mu$. Moreover, in [6] the axiom CC (see Table 1) is neither present, nor derivable.
Table 5
Additional meaninglessness and transition rules for $\mathcal{G}_{\mathcal{L}(P_A)}(ACP, [])$ in $\text{panm}$-format

<table>
<thead>
<tr>
<th>Rule</th>
<th>Left-hand side</th>
<th>Right-hand side</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi \circ \psi) = M$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\psi) = M$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, \delta_H(X))$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, \delta_H(X))$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, \delta_H(X))$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, X_{\phi \parallel \psi} Y)$ if $w(\phi) = C, T, F$, $\mu(w, Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, X_{\phi \parallel \psi} Y)$, $\mu(w, Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
<tr>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td>$\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$ if $w(\phi) = C, T, F$, $\mu(w, a Y_{\phi \parallel \psi} X)$, $\mu(w, a Y_{\phi \parallel \psi} X)$</td>
<td></td>
</tr>
</tbody>
</table>
one can distinguish a greatest lower bound (notation $\sqcap$) of $T$ and $F$ that majorizes $D$ (see Fig. 2), and extend the definition of conditional composition with $x \triangleleft (T \sqcap F) \triangleright y = x \sqcap y$. This yields a six-valued logic in which the identities $F \triangleleft (T \sqcap F) \triangleright F = F$ and $F \triangleleft D \triangleright F = D$ illustrate the difference between $T \sqcap F$ and $D$. Although this logic is simple and elegant (e.g., conditional composition also distributes over $\sqcap$, cf. Proposition 1), we see no algorithmic motive for distinguishing $D$ and $T \sqcap F$. We can employ process algebraic conditional composition to support this position: by $x \triangleleft (T \sqcap F) \triangleright x = x$ we obtain the associated identity $X +_{T\sqcap F} X = X$, and by $x \triangleleft (T \sqcap F) \triangleright D = D$ we find $X +_{T\sqcap F} \delta = \delta$. This illustrates that the operation $+_{T\sqcap F}$ models a notion of choice, say “pessimistic choice”, for which we have no useful intuition or application. 6

Acknowledgements

We thank the referees for useful and inspiring comments.

6 When combining this six-valued logic with process algebra, the axiom CC (see Table 1) appears to be the only one that should be changed: it allows one to derive undesirable identities, such as $a +_{T\sqcap F} b = (a + X) +_{T\sqcap F} b$. We note that CC is crucial for Lemma 12, and thereby for our completeness results.

References