Sequential propositional logic deviates from conventional propositional logic by taking into account that during the sequential evaluation of a propositional statement, atomic propositions may yield different Boolean values at repeated occurrences. We introduce ‘free valuations’ to capture this dynamics of a propositional statement’s environment. The resulting logic is phrased as an equationally specified algebra rather than in the form of proof rules, and is named ‘proposition algebra’. It is strictly more general than Boolean algebra to the extent that the classical connectives fail to be expressively complete in the sequential case. The four axioms for free valuation congruence are then combined with other axioms in order define a few more valuation congruences that gradually identify more propositional statements, up to static valuation congruence (which is the setting of conventional propositional logic).

Proposition algebra is developed in a fashion similar to the process algebra ACP and the program algebra PGA, via an algebraic specification which has a meaningful initial algebra for which a range of coarser congruences are considered important as well. In addition infinite objects (that is propositional statements, processes and programs respectively) are dealt with by means of an inverse limit construction which allows the transfer of knowledge concerning finite objects to facts about infinite ones while reducing all facts about infinite objects to an infinity of facts about finite ones in return.

Categories and Subject Descriptors: F.3.2 [Logics and Meanings of Programs]: Semantics of Programming Languages—Algebraic approaches to semantics

General Terms: Theory; Verification

Additional Key Words and Phrases: Conditional composition, propositional statement, reactive valuation, satisfiability, sequential connective, short-circuit evaluation, side effect

1. INTRODUCTION

A propositional statement is a composition of atomic propositions made by means of one or more (proposition) composition mechanisms, usually called connectives. Atomic propositions are considered to represent facts about an environment (execution environment, execution architecture, operating context) that are used by the
logical mechanism contained in the propositional statement which aggregates these facts for presentation to the propositional statement’s user. Different occurrences of the same atomic propositions represent different queries (measurements, issued information requests) at different moments in time.

A valuation that may return different Boolean values for the same atomic proposition during the sequential evaluation of a single propositional statement is called free, or in the case the evaluation result of an atomic proposition can have effect on subsequent evaluation, it is called reactive. This is in contrast to a “static” valuation, which always returns the same value for the same atomic proposition. Free valuations are thus semantically at the opposite end of static valuations, and are observation based in the sense that they capture the identity of a propositional statement as a pattern of queries followed by a Boolean value.

Many classes of valuations can be distinguished. Given a class $K$ of valuations, two propositional statements are $K$-equivalent if they evaluate to the same Boolean value for each valuation in $K$. Given a family of proposition connectives, $K$-equivalence need not be a congruence, and $K$-congruence is the largest congruence that is contained in $K$-equivalence. It is obvious that with larger $K$ more propositional statements can be distinguished and the one we consider most distinguishing is named free valuation congruence. It is this congruence that plays the role of an initial algebra for the proposition algebras developed in this paper. The axioms of proposition algebra specify free valuation congruence in terms of the single ternary connective conditional composition (in computer science terminology: if-then-else) and constants for truth and falsity, and their soundness and completeness (for closed equations) is easily shown. Additional axioms are given for static valuation congruence, and for some reactive valuation congruences in between.

Sequential versions of the well-known binary connectives of propositional logic and negation can be expressed in terms of conditional composition. We prove that these connectives have insufficient expressive power at this level of generality and that a ternary connective is needed (in fact this holds for any collection of binary connectives definable by conditional composition.)

Repeated use of the same atomic proposition is meaningful in free or reactive valuation semantics, and as a consequence infinite propositions are meaningful and may be more expressive than finite ones. Infinite propositions are defined by means of an inverse limit construction which allows the transfer of knowledge concerning finite objects to facts about infinite ones while reducing all facts about infinite objects to an infinity of facts about finite ones in return. This construction was applied in giving standard semantics for the process algebra ACP by Bergstra and Klop [1984] (see Baeten and Weijland [1990] for a more recent overview). In doing so the design of proposition algebra is very similar to the thread algebra of Bergstra and Middelburg [2007] which is based on a similar ternary connective but which features constants for termination and deadlock rather than for truth and falsity. Whereas thread algebra focuses on multi-threading and concurrency, proposition algebra has a focus on sequential mechanisms.

The paper is structured as follows: In the next section we discuss some motivation for proposition algebra. In Section 3 we define the signature and equations of proposition algebra, and in Section 4 we formally define valuation algebras. In Sec-
tion 5 we consider some observation based equivalences and congruences generated by valuations, and in Sections 6-9 we provide complete axiomatisations of these congruences. Definable (binary) connectives are formally introduced in Section 10. In Section 11 we briefly consider some complexity issues concerning satisfiability. The expressiveness (functional incompleteness) of binary connectives is discussed in Section 12. In Section 13 we introduce projection and projective limits for defining potentially infinite propositions, and in Section 14 we discuss recursive specifications of infinite propositions. The paper is ended with some conclusions in Section 15.

2. MOTIVATION FOR PROPOSITION ALGEBRA

Proposition algebra is proposed as a preferred way of viewing the data type of propositional statements, at least in a context of sequential systems. Here are some arguments in favor of that thesis:

In a sequential program a test, which is a conjunction of \( P \) and \( Q \) will be evaluated in a sequential fashion, beginning with \( P \) and not evaluating \( Q \) unless the evaluation of \( P \) led to a positive outcome. The sequential form of evaluation takes precedence over the axioms or rules of conventional propositional logic or Boolean algebra. For instance, neither conjunction nor disjunction are commutative when evaluated sequentially in the presence of side-effects, errors or exceptions. The absence of these latter features is never claimed for imperative programming and thus some extension or modification of ordinary two-valued logic is necessary to understand the basics of propositional logic as it occurs in the context of imperative programs. Three-, four- or more sophisticated many-valued logics may be used to explain the logic in this case (see, e.g., [Bergstra et al. 1995; Bergstra and Ponse 1998b; Hähnle 2005]). The non-commutative, sequential reading of conjunction mentioned above can be traced back to the seminal work on computation theory by McCarthy [1963], in which a specific value for undefinedness (e.g., a divergent computation) is considered that in conjunction with falsity results in the value that was evaluated first. In many explanations of the semantics of Boolean operators in programming languages, this form of sequential evaluation is called short-circuit evaluation (and in some, minimal evaluation or McCarthy evaluation).

Importing non-commutative conjunction to two-valued propositional logic means that the sequential order of events is significant, and that is what proposition algebra is meant to specify and analyze in the first place. As a simple example, consider the propositional statement that a pedestrian evaluates just before crossing a road with two-way traffic driving on the right:

\[
\text{look-left-and-check} \land \text{look-right-and-check} \land \text{look-left-and-check}. \tag{1}
\]

Here \( \land \) is left-sequential conjunction, which is similar to conjunction but the left argument is evaluated first and upon \( F \) ("false"), evaluation finishes with result \( F \). A valuation associated with this example is (should be) a free valuation: also in the case that the leftmost occurrence of \( \text{look-left-and-check} \) evaluates to \( T \) ("true"), its second evaluation might very well evaluate to \( F \). However, the order of events (or their amount) needs not to be significant in all circumstances and one may still wish or require that in particular cases conjunction is idempotent or even commutative.
A most simple example is perhaps

\[ a \land a = a \]

with \( a \) an atomic proposition, which is not valid in free valuation semantics (and neither is the falsity of \( a \land \neg a \)). For this reason we distinguish a number of restricted forms of reactive valuation equivalences and congruences that validate this example or variations thereof, but still refine static valuation congruence. It is evident that many more such refinements can be distinguished.

We take the point of departure that the very purpose of any action taken by a program under execution is to change the state of a system. If no change of state results with certainty the action can just as well be skipped. This holds for tests as well as for any action mainly performed because of its intended side-effects. The common intuition that the state is an external matter not influenced by the evaluation of a test, justifies ignoring side-effects of tests and for that reason it justifies an exclusive focus on static valuations to a large extent, thereby rendering the issue of reactivity pointless as well. But there are some interesting cases where this intuition is not necessarily convincing. We mention three such issues, all of which also support the general idea of considering propositional statements under sequential evaluation:

1. It is common to accept that in a mathematical text an expression \( x/x \) is admissible only after a test \( x \neq 0 \) has been performed. One might conceive this test as an action changing the state of mind of the reader thus influencing the evaluation of further assertions such as \( x/x = 1 \).

2. A well-known dogma on computer viruses introduced by Cohen in 1984 (journal publication in [1987]) states that a computer cannot decide whether or not a program that it is running is itself a virus. The proof involves a test which is hypothetically enabled by a decision mechanism which is supposed to have been implemented on a modified instance of the machine under consideration. It seems fair to say that the property of a program being viral is not obviously independent of the state of the program. So here is a case where performing the test might (in principle at least) result in a different state from which the same test would lead to a different outcome. This matter has been analyzed in detail in Bergstra and Ponse [2005] and Bergstra et al. [2007] with the conclusion that the reactive nature of valuations gives room for criticism of Cohen’s original argument. In the didactic literature on computer security Cohen’s viewpoint is often repeated and it can be found on many websites and in the introduction of many texts. But there is a remarkable lack of secondary literature on the matter; an exception is the discussion in Cohen [2001] and the papers cited therein.

3. The on-line halting problem is about execution environments which allow a running program to acquire information about its future halting or divergence. This information is supposed to be provided by means of a forecasting service. In Bergstra and Ponse [2007] that feature is analyzed in detail in a setting of thread algebra and the impossibility of sound and complete forecasting of halting is established. In particular, calling a forecasting service may have side-effects which leads to different replies in future calls (see, e.g., Ponse and

Table I. The set CP of axioms for proposition algebra

van der Zwaag [2008]).

Our account of proposition algebra is based on the ternary operator conditional composition (or if-then-else). This operator has a sequential form of evaluation as its natural semantics, and thus combines naturally with free and reactive valuation semantics. Furthermore, proposition algebra constitutes a simple setting for constructing infinite propositions by means of an inverse limit construction. The resulting projective limit model can be judged as one that didactically precedes (prepares for) technically more involved versions for process algebra and thread algebra, and as such provides by itself a motivation for proposition algebra.

3. PROPOSITION ALGEBRA

In this section we introduce the signature and equational axioms of proposition algebra. Let $A$ be a countable set of atomic propositions $a, b, c, \ldots$. The elements of $A$ serve as atomic (i.e., non-divisible) queries that will produce a Boolean reply value.

We assume that $|A| > 1$. The case that $|A| = 1$ is described in detail in Regenboog [2010]. We come back to this point in Section 15.

The signature of proposition algebra consists of the constants $T$ and $F$ (representing true and false), a constant $a$ for each $a \in A$, and, following Hoare [1985b], the ternary operator conditional composition

$$\diamond \triangleleft \vdash \triangleleft$$

We write $\Sigma_{\text{CP}}(A)$ for the signature introduced here. Terms are subject to the equational axioms in Table I. We further write CP for this set of axioms (where CP abbreviates conditional propositions).

An alternative name for the conditional composition

$$y \diamond x \triangleright z$$

is if $x$ then $y$ else $z$: the axioms CP1 and CP2 model that its central condition $x$ is evaluated first, and depending on the reply either its leftmost or rightmost argument is evaluated. Axiom CP3 establishes that a term can be extended to a larger conditional composition by adding $T$ as a leftmost argument and $F$ as a rightmost one, and CP4 models the way a non-atomic central condition distributes over the outer arguments. We note that the expression

$$F \diamond x \triangleright T$$

can be seen as defining the negation of $x$:

$$\text{CP} \vdash z \diamond (F \diamond x \triangleright T) \triangleright y = (z \diamond F \triangleright y) \diamond x \triangleright (z \triangleright T \triangleright y) = y \diamond x \triangleright z, \quad (2)$$
which illustrates that “if \( \neg x \) then \( z \) else \( y \)” and “if \( x \) then \( y \) else \( z \)” are considered equal.

We introduce the abbreviation
\[
x \circ y \quad \text{for} \quad y \triangleleft x \triangleright y,
\]
and we name this expression \( x \) and then \( y \). It follows easily that \( \circ \) is associative:
\[
(x \circ y) \circ z = z \triangleleft (y \triangleleft x \triangleright y) \triangleright z = (z \triangleleft y \triangleright z) \triangleleft x \triangleright (z \triangleleft y \triangleright z) = x \circ (y \circ z).
\]
We take the and-then operator \( \circ \) to bind stronger than conditional composition. At a later stage we will formally add negation, the “and then” connective \( \circ \), and some other binary connectives to proposition algebra (i.e., add their function symbols to \( \Sigma_{\text{CP}}(A) \) and their defining equations to CP).

Closed terms over \( \Sigma_{\text{CP}}(A) \) are called \textit{propositional statements}, with typical elements \( P, Q, R, \ldots \).

**Definition 3.1.** A propositional statement \( P \) is a \textbf{basic form} if
\[
P ::= T \mid F \mid P_1 \triangleleft a \triangleright P_2
\]
with \( a \in A \), and \( P_1 \) and \( P_2 \) basic forms.

So, basic forms can be seen as binary trees of which the leaves are labeled with either \( T \) or \( F \), and the internal nodes with atomic propositions. Following Baeten and Weijland [1990] we use the name \textit{basic form} instead of \textit{normal form} because we associate the latter with a term rewriting setting.

**Lemma 3.2.** Each propositional statement can be proved equal to one in basic form using the axioms in Table I.

**Proof.** We first show that if \( P, Q, R \) are basic forms, then \( P \triangleleft Q \triangleright R \) can be proved equal to a basic form by structural induction on \( Q \). If \( Q = T \) or \( Q = F \), this follows immediately, and if \( Q = Q_1 \triangleleft a \triangleright Q_2 \) then
\[
\text{CP} \vdash P \triangleleft Q \triangleright R = P \triangleleft (Q_1 \triangleleft a \triangleright Q_2) \triangleright R
= (P \triangleleft Q_1 \triangleright R) \triangleleft a \triangleright (P \triangleleft Q_2 \triangleright R)
\]
and by induction there are basic forms \( P_i \) for \( i = 1, 2 \) such that \( \text{CP} \vdash P_i = P \triangleleft Q_i \triangleright R \), hence \( \text{CP} \vdash P \triangleleft Q \triangleright R = P_1 \triangleleft a \triangleright P_2 \) and \( P_1 \triangleleft a \triangleright P_2 \) is a basic form.

Next we prove the lemma’s statement by structural induction on the form that propositional statement \( P \) may take. If \( P = T \) or \( P = F \) then \( P \) is a basic form, and if \( P = a \), then \( \text{CP} \vdash P = T \triangleleft a \triangleright F \). For the case \( P = P_1 \triangleleft P_2 \triangleright P_3 \) it follows by induction that there are basic forms \( Q_1, Q_2, Q_3 \) with \( \text{CP} \vdash P_i = Q_i \), so \( \text{CP} \vdash P = Q_1 \triangleleft Q_2 \triangleright Q_3 \). Now apply the first result. \( \square \)

We write
\[
P \equiv Q
\]
to denote that propositional statements \( P \) and \( Q \) are syntactically equivalent. In Section 6 we prove that basic forms constitute a convenient representation:

**Proposition 3.3.** If \( \text{CP} \vdash P = Q \) for basic forms \( P \) and \( Q \), then \( P \equiv Q \).
4. VALUATION ALGEBRAS

In this section we define valuation algebras. Let $B$ be the sort of the Booleans with constants $T$ and $F$. The signature $\Sigma_{\text{Val}}(A)$ of valuation algebras contains the sort $B$ and a sort $\text{Val}$ of valuations. The sort $\text{Val}$ has two constants $T_{\text{Val}}$ and $F_{\text{Val}}$, which represent the valuations that assign to each atomic proposition the value $T$ respectively $F$, and for each $a \in A$ a function $y_a : \text{Val} \rightarrow B$ called the yield of $a$, and a function $a \bullet : \text{Val} \rightarrow \text{Val}$ called the $a$-derivative. Given a valuation $H$ we write $a \bullet H$ (instead of $a \bullet (H)$) to denote the transformation of $H$ after $a$ has been evaluated.

DEFINITION 4.1. A $\Sigma_{\text{Val}}(A)$-algebra $\mathcal{A}$ is a valuation algebra (VA) if for all $a \in A$ it satisfies the axioms

\[
y_a(T_{\text{Val}}) = T, \\
y_a(F_{\text{Val}}) = F, \\
a \bullet T_{\text{Val}} = T_{\text{Val}}, \\
a \bullet F_{\text{Val}} = F_{\text{Val}}.
\]

Given a valuation algebra $\mathcal{A}$, a valuation $H$ (of sort $\text{Val}$) in $\mathcal{A}$, and a propositional statement $P$ we now define both the evaluation of $P$ over $H$, notation $P/H$, and the $P$-derivative of $H$, a generalized notion of the $a$-derivative of $H$, notation $P \bullet H$.

DEFINITION 4.2. Let $\mathcal{A}$ be a valuation algebra and $H$ a valuation in $\mathcal{A}$, and let $P, Q, R$ be propositional statements. Evaluation and (valuation) derivatives are defined by the following case distinctions:

\[
T/H = T, \\
F/H = F, \\
a/H = y_a(H), \\
(P \circ Q \triangleright R)/H = \begin{cases} 
  P/(Q \bullet H) & \text{if } Q/H = T, \\
  R/(Q \bullet H) & \text{if } Q/H = F,
\end{cases}
\]

and

\[
T \bullet H = H, \\
F \bullet H = H, \\
(P \circ Q \triangleright R) \bullet H = \begin{cases} 
  P \bullet (Q \bullet H) & \text{if } Q/H = T, \\
  R \bullet (Q \bullet H) & \text{if } Q/H = F.
\end{cases}
\]
Some explanation: whenever in a conditional composition the central condition is an atomic proposition, say \( c \), then a valuation \( H \) distributes over the outer arguments as \( c \cdot H \), thus

\[
(P \triangleleft c \triangleright Q)/H = \begin{cases} 
  P/(c \cdot H) & \text{if } y_c(H) = T, \\
  Q/(c \cdot H) & \text{if } y_c(H) = F.
\end{cases}
\] (3)

If in a conditional composition the central condition is not atomic, valuation decomposes further according to the equations above, e.g.,

\[
(a \triangleleft (b \triangleleft c \triangleright d) \triangleright e)/H = \begin{cases} 
  a/((b \triangleleft c \triangleright d) \cdot H) & \text{if } (b \triangleleft c \triangleright d)/H = T, \\
  e/((b \triangleleft c \triangleright d) \cdot H) & \text{if } (b \triangleleft c \triangleright d)/H = F,
\end{cases}
\] \[
(a \triangleleft (b \cdot (c \cdot H)) \text{ if } y_c(H) = T \text{ and } y_b(c \cdot H) = T, \\
(a \cdot (d \cdot (c \cdot H)) \text{ if } y_c(H) = F \text{ and } y_d(c \cdot H) = T, \\
(c/((d \cdot (c \cdot H)) \text{ if } y_c(H) = T \text{ and } y_b(c \cdot H) = F, \\
(d/((d \cdot (c \cdot H)) \text{ if } y_c(H) = F \text{ and } y_d(c \cdot H) = F.
\] (4)

We compare the last example with

\[
((a \triangleleft b \triangleright e) \triangleleft c \triangleright (a \triangleleft d \triangleright e))/H,
\] (5)

which is a particular instance of (3) above. For the case \( y_c(H) = T \) we find from (3) that

\[
(5) = (a \triangleleft b \triangleright e)/(c \cdot H) = \begin{cases} 
  a/(b \cdot (c \cdot H)) & \text{if } y_b(c \cdot H) = T, \\
  e/(b \cdot (c \cdot H)) & \text{if } y_b(c \cdot H) = F,
\end{cases}
\]

and for the case \( y_c(H) = F \) we find the other two right-hand sides of (4). In a similar way it follows that

\[
((a \triangleleft b \triangleright e) \triangleleft c \triangleright (a \triangleleft d \triangleright e)) \cdot H = (a \triangleleft (b \triangleleft c \triangleright d) \triangleright e) \cdot H,
\]

thus providing a prototypical example of the soundness of axiom CP4 of CP.

**Theorem 4.3 (Soundness).** If for propositional statements \( P \) and \( Q \), CP \( \vdash P = Q \), then for each VA \( \mathcal{A} \) and each valuation \( H \in \mathcal{A} \),

\[
P/H = Q/H \text{ and } P \cdot H = Q \cdot H.
\]

**Proof.** Let \( \mathcal{A} \) be some VA and \( H \in \mathcal{A} \). It is an easy exercise to show that an arbitrary instance \( P = Q \) of one of the axioms in CP satisfies \( P/H = Q/H \) and \( P \cdot H = Q \cdot H \).

Assume that CP \( \vdash P = Q \) follows from the context rule, thus \( P \equiv P_1 \triangleleft P_2 \triangleright P_3, \) \( Q \equiv Q_1 \triangleleft Q_2 \triangleright Q_3, \) and CP \( \vdash P_1 \triangleleft Q_1 \). Then for all \( H \in \mathcal{A} \), \( P_1/H = Q_1/H \) and \( P_2 \cdot H = Q_2 \cdot H \), and we find the desired result by case distinction: let \( H \in \mathcal{A} \) and \( P_2/H = Q_2/H \text{ and } P_2/H = T \), then

\[
(P_1 \triangleleft P_2 \triangleright P_3)/H = P_1/(P_2 \cdot H) = Q_1/(Q_2 \cdot H) = (Q_1 \triangleleft Q_2 \triangleright Q_3)/H,
\]

\[
(P_1 \triangleleft P_2 \triangleright P_3) \cdot H = P_1 \cdot (P_2 \cdot H) = Q \cdot (Q_2 \cdot H) = (Q_1 \triangleleft Q_2 \triangleright Q_3) \cdot H.
\]

For the case \( P_2/H = Q_2/H \text{ and } P_2/H = F \) a similar argument applies. \( \square \)
5. VALUATION VARIETIES AND VALUATION CONGRUENCES

We introduce some specific equivalences and congruences generated by classes of valuations. The class of VAs that satisfy a certain collection of equations over \( \Sigma_{Val}(A) \) is called a valuation variety. We distinguish the following varieties, where each next one is subvariety of the one defined:

1. The variety of VAs with free valuations is defined by the equations in Definition 4.1.
2. The variety of VAs with repetition-proof valuations: all VAs that satisfy for all \( a \in A \),
   \[
   y_a(x) = y_a(a \cdot x).
   \]
   So the reply to a series of consecutive atoms \( a \) is determined by the first reply.
   Typical example: \( (P \triangleleft a \sqcup Q) \triangleleft a \sqcup (R \triangleleft a \sqcup S) = (a \cdot P) \triangleleft a \sqcup (a \cdot S) \).
3. The variety of VAs with contractive valuations: all repetition-proof VAs that satisfy for all \( a \in A \),
   \[
   a \cdot (a \cdot x) = a \cdot x.
   \]
   Each successive atom \( a \) is contracted by using the same evaluation result.
   Typical example: \( (P \triangleleft a \triangleright Q) \triangleright a \triangleright (R \triangleleft a \triangleright S) = P \triangleleft a \triangleright S \).
4. The variety of VAs with weakly memorizing valuations consists of all contractive VAs that satisfy for all \( a,b \in A \),
   \[
   y_a(a \cdot x) = y_a(x) \to (a \cdot (b \cdot (a \cdot x))) = b \cdot (a \cdot x) \land y_a(b \cdot (a \cdot x)) = y_a(x).
   \]
   Here the evaluation result of an atom \( a \) is memorized in a subsequent evaluation of \( a \) if the evaluation of intermediate atoms yields the same result, and this subsequent \( a \) can be contracted.
   Two typical examples are
   \[
   ((P \triangleleft a \triangleright Q) \triangleright b \triangleright R) \triangleright a \triangleright S = (P \triangleleft b \triangleright R) \triangleright a \triangleright S,
   P \triangleleft a \triangleright (Q \triangleleft b \triangleright (R \triangleleft c \triangleright (S \triangleleft a \triangleright V))) = P \triangleleft a \triangleright (Q \triangleleft b \triangleright (R \triangleleft c \triangleright V)).
   \]
5. The variety of VAs with memorizing valuations: all contractive VAs that satisfy for all \( a,b \in A \),
   \[
   a \cdot (b \cdot (a \cdot x)) = b \cdot (a \cdot x) \land y_a(b \cdot (a \cdot x)) = y_a(x).
   \]
   Here the evaluation result of an atom \( a \) is memorized in all subsequent evaluations of \( a \) and all subsequent \( a \)'s can be contracted.
   Typical axiom (right-oriented version):
   \[
   x \triangleright y \triangleright (z \triangleright u \triangleright (v \triangleright y \triangleright w)) = x \triangleright y \triangleright (z \triangleright u \triangleright w).
   \]
   Typical counter-example: \( a \triangleright b \triangleright F \neq b \triangleleft a \triangleright F \) (thus \( b \land a \neq a \land b \)).
6. The variety of VAs with static valuations: all VAs that satisfy for all \( a,b \in A \),
   \[
   y_a(b \cdot x) = y_a(x)
   \]
   This is the setting of conventional propositional logic.
   Typical identities: \( a = b \triangleright a \) and \( a \triangleright b \triangleright F = b \triangleleft a \triangleright F \) (thus \( b \land a = a \land b \)).
Definition 5.1. Let \( K \) be a variety of valuation algebras over \( A \). Then propositional statements \( P \) and \( Q \) are \( K \)-equivalent, notation \( P \equiv_K Q \), if \( P/H = Q/H \) for all \( \mathcal{A} \in K \) and \( H \in \mathcal{A} \). Let \( =_K \) be the largest congruence contained in \( \equiv_K \). Propositional statements \( P \) and \( Q \) are \( K \)-congruent if \( P =_K Q \).

From this definition it follows that \( P =_K Q \) if \( P \equiv_K Q \) and for all propositional statements \( R \) and \( S \), the following three cases are true:

\[
\begin{align*}
P & \equiv_K Q, & (6) \\
R & \equiv_K R \implies \equiv_K R, & (7) \\
R & \equiv_K S \implies \equiv_K S. & (8)
\end{align*}
\]

Case (6) follows immediately from \( P \equiv_K Q \); for all \( \mathcal{A} \in K \) and \( H \in \mathcal{A} \),

\[
(P \equiv R \equiv S)/H = \begin{cases} 
P/(R \cdot H) = Q/(R \cdot H) = (Q \equiv R \equiv S)/H & \text{if } R/H = T, \\
S/(R \cdot H) = (Q \equiv R \equiv S)/H & \text{if } R/H = F,
\end{cases}
\]

and case (8) follows in a similar way. So in order to prove \( P =_K Q \) it suffices to prove that case (7) is true (note that with \( R = T \) and \( S = F \), case (7) implies \( P =_K Q \)). We will often use the contraposition of this conditional property:

\[
P \neq_K Q \implies \exists \mathcal{A} \in K, H \in \mathcal{A}, R, S \left( (R \equiv P \equiv S)/H \neq (R \equiv Q \equiv S)/H \right). & (9)
\]

By the varieties defined thus far we distinguish six types of \( K \)-equivalence and \( K \)-congruence: free, repetition-proof, contractive, weakly memorizing, memorizing and static. We use the following abbreviations for these:

\[
\begin{align*}
K & = fr, \, rp, \, cr, \, wm, \, mem, \, st,
\end{align*}
\]

respectively.

Proposition 5.2. The inclusions \( \equiv_{fr} \subseteq \equiv_{rp} \subseteq \equiv_{cr} \subseteq \equiv_{wm} \subseteq \equiv_{mem} \subseteq \equiv_{st} \), and \( =_K \subseteq \equiv_K \) for \( K \in \{fr, \, rp, \, cr, \, wm, \, mem\} \) are all proper.

Proof. In this proof we assume that all VAs we use satisfy \( T \neq F \). We first consider the differences between the mentioned equivalences:

1. \( a \equiv_{rp} a \circ a \circ F \), but \( \equiv_{fr} \) does not hold in this case as is witnessed by a VA with valuation \( H \) that satisfies \( y_a(H) = T \) and \( y_a(a \circ H) = F \) (yielding \( a/H = T \) and \( (a \circ a \circ F)/H = F \)).

2. \( b \circ a \circ F \equiv_{cr} b \circ (a \circ a \circ F) \circ F \), but \( \equiv_{rp} \) does not hold in the VA with element \( H \) with \( y_a(H) = y_b(a \circ H) = T \) and \( y_b(a \circ (a \circ H)) = F \).

3. \( (a \circ b \circ F) \circ a \circ F \equiv_{wm} b \circ a \circ F \), but \( \equiv_{cr} \) does not hold in the VA with element \( H \) with \( y_a(H) = y_b(a \circ H) = T \) and \( y_b(b \circ (a \circ H)) = F \).

4. \( (T \circ b \circ (F \circ a \circ T)) \circ a \circ F \equiv_{mem} b \circ a \circ F \), but \( \equiv_{wm} \) does not hold in the VA with element \( H \) with \( y_a(H) = T \) and \( y_b(a \circ H) = y_a(b \circ (a \circ H)) = F \).
(5) \( a \equiv a \triangleleft b \triangleright a \) (distinguish all possible cases), but \( \equiv_{\text{mem}} \) does not hold as is witnessed by the VA with element \( H \) with \( y_a(H) = y_b(H) = T \) and \( y_a(b \bullet H) = F \) (yielding \( a/H = T \) and \( (a \triangleleft b \triangleright a)/H = F \)).

Finally, observe that for \( K \in \{ \text{fr}, \text{rp}, \text{cr}, \text{wm}, \text{mem} \} \) it holds that \( T \equiv K T \triangleleft a \triangleright T \), but \( b \triangleleft T \triangleright T \not\equiv K b \triangleleft (T \triangleleft a \triangleright T) \triangleright T \) as is witnessed by the VA with element \( H \) with \( y_a(H) = y_b(H) = T \) and \( y_a(b \bullet H) = F \).

The following proposition stems from Regenboog [2010] and can be used to deal with the difference between \( K \)-congruence and \( K \)-equivalence.

**Proposition 5.3.** If \( P \equiv K Q \) and for all \( A \in K \) and \( H \in A \), \( P \bullet H = Q \bullet H \), then \( P =_K Q \).

**Proof.** Assume \( P \equiv K Q \) and the further requirement in the proposition is satisfied. As argued before (see (7)), \( P =_K Q \) if for all propositional statements \( R \) and \( S \),

\[
R \triangleleft P \triangleright S \equiv_K R \triangleleft Q \triangleright S.
\]

This follows easily: let \( A \in K \) and \( H \in A \) and \( P \bullet H = Q \bullet H \), then

\[
(R \triangleleft P \triangleright S)/H = \begin{cases} 
R/(P \bullet H) & \text{if } P/H = T, \\
S/(P \bullet H) & \text{if } P/H = F,
\end{cases}
\]

\[
= \begin{cases} 
R/(Q \bullet H) & \text{if } Q/H = T, \\
S/(Q \bullet H) & \text{if } Q/H = F,
\end{cases}
\]

\[
= (R \triangleleft Q \triangleright S)/H.
\]

As a consequence, the soundness of CP with respect to all valuation varieties \( K \) introduced can be phrased as follows: for all propositional statements \( P \) and \( Q \),

\[
\text{CP} \vdash P = Q \implies P =_K Q.
\]

In particular this holds for free valuation congruence, the most distinguishing congruence we consider, thus \(\text{CP} \vdash P = Q \implies P =_{\text{fr}} Q\).

6. COMPLETENESS FOR THE VARIETIES \( \text{fr}, \text{rp} \) AND \( \text{cr} \)

In this section we provide complete axiomatizations of free valuation congruence, and repetition-proof and contractive valuation congruence. We start with a basic result on free valuation congruence of basic forms.

**Lemma 6.1.** For all basic forms \( P \) and \( Q \), \( P =_{\text{fr}} Q \) implies \( P \equiv Q \).

**Proof.** The implication of the lemma follows by contraposition and structural induction on \( P \), where in each case we apply structural induction on \( Q \).

If \( P = T \) then if \( Q = F \), \( P \not=_{\text{fr}} Q \) is witnessed by each VA in which \( T \not= F \). If \( Q = Q_1 \triangleleft a \triangleright Q_2 \), consider a VA with valuation \( H \) that satisfies \( y_a(H) = y_b(H) = T \) and for all propositional statements \( R \), \( y_b(R \bullet (a \bullet H)) = F \). Then \( (P \triangleleft b)/H = T \) while \( (Q \triangleleft b)/H = y_b(Q_1 \bullet (a \bullet H)) = F \), so \( P \not=_{\text{fr}} Q \).

If \( P = F \) a similar argument applies.
Lemma 6.1.

If \( P \equiv P_1 \triangleleft a \triangleright P_2 \) then the cases \( Q \equiv T \) and \( Q \equiv F \) can be dealt with as above. If \( Q \equiv Q_1 \triangleleft a \triangleright Q_2 \) then assume \( P \not\equiv Q \) because \( P_1 \not\equiv Q_1 \). By induction, \( P_1 \not\equiv_{fr} Q_1 \). By implication (9) there exists \( \Lambda \in A \) with valuation \( H \) and propositional statements \( R \) and \( S \) such that \( (R \triangleleft P_1 \triangleright S)/H \not\equiv (R \triangleleft Q_1 \triangleright S)/H \). But then there is \( \Lambda' \supseteq \Lambda \) with valuation \( H' \) such that \( y_a(H') = T \) and \( a \circ H' = H \) by which \( P \not\equiv_{fr} Q \) because \( R \triangleleft P \triangleright Q \) and \( R \triangleleft Q \triangleright S \) yield different results under \( H' \):

\[
(R \triangleleft P \triangleright S)/H' = (R \triangleleft P_1 \triangleright S)/H \not\equiv (R \triangleleft Q_1 \triangleright S)/H = (R \triangleleft Q \triangleright S)/H'.
\]

Finally, if \( Q \equiv Q_1 \triangleleft b \triangleright Q_2 \) with \( b \not\equiv a \) then \( P \not\equiv_{fr} Q \) follows by considering \( \Lambda \) with \( T \not\equiv F \) and with valuation \( H \) that satisfies \( y_a(H) = y_b(H) = T \) and for all \( R \), \( y_a(R \circ (a \bullet H)) = T \) and \( y_a(R \circ (b \bullet H)) = F \). Clearly, \( (P \circ a)/H \not\equiv (Q \circ a)/H \).

As a corollary we find a proof of Proposition 3.3, i.e., for basic forms, provable equality in \( CP \) and syntactic equivalence coincide:

**Proof of Proposition 3.3.** By the soundness of \( CP \) it is sufficient to prove that for all basic forms \( P \) and \( Q \), \( P \equiv_{fr} Q \) implies \( P \equiv Q \), and this is proved in Lemma 6.1.

It easily follows that \( CP \) axiomatizes free valuation congruence:

**Theorem 6.2 (Completeness).** If \( P \equiv_{fr} Q \) for propositional statements \( P \) and \( Q \), then \( CP \vdash P = Q \).

**Proof.** Assume \( P \equiv_{fr} Q \). By Lemma 3.2 there are basic forms \( P' \) and \( Q' \) with \( CP \vdash P = P' \) and \( CP \vdash Q = Q' \). By soundness, \( P' \equiv_{fr} Q' \) and by Proposition 3.3 \( P' \equiv Q' \). Hence, \( CP \vdash P = Q \).

We proceed by discussing completeness results for the valuation varieties \( rp \) and \( cr \) introduced in the previous section.

Write \( CP_{rp} \) for the axioms in \( CP \) and these axiom schemes \((a \in A)\):

\[
(CPrp1) \quad (x \triangleleft a \triangleright y) \triangleleft a \triangleright z = (x \triangleleft a \triangleright x) \triangleleft a \triangleright z,
\]

\[
(CPrp2) \quad x \triangleleft a \triangleright (y \triangleleft a \triangleright z) = x \triangleleft a \triangleright (z \triangleleft a \triangleright z).
\]

**Theorem 6.3.** Repetition-proof valuation congruence \( \equiv_{rp} \) is axiomatized by \( CP_{rp} \).

**Proof.** Let \( \Lambda \) be a VA in the variety \( rp \) of repetition-proof valuation algebras, thus for all \( a \in A \),

\[ y_a(x) = y_a(a \bullet x). \]

Concerning soundness, we only check axiom scheme \( CPrp2 \) (a proof for \( CPrp1 \) is very similar): let \( H \in \Lambda \), then

\[
(P \triangleleft a \triangleright (Q \triangleleft a \triangleright R))/H = \begin{cases} P/(a \bullet H) & \text{if } y_a(H) = T, \\ (Q \triangleleft a \triangleright R)/(a \bullet H) & \text{if } y_a(H) = F, \\ R/(a \bullet (a \bullet H)) & \text{if } y_a(H) = F = y_a(a \bullet H), \end{cases}
\]

Published in ACM Transactions on Computational Logic, Vol. , No. , 20.
These schemes contract for each and all $a$.

P \iff y_a(H) = T,$

\begin{align*}
(P \triangleleft a \triangleright (Q \triangleleft a \triangleright R)) \bullet H &= \begin{cases} 
  P \bullet (a \bullet H) & \text{if } y_a(H) = T, \\
  (Q \triangleleft a \triangleright R) \bullet (a \bullet H) & \text{if } y_a(H) = F,
\end{cases} \\
&= \begin{cases} 
  P \bullet (a \bullet H) & \text{if } y_a(H) = T, \\
  R \bullet (a \bullet (a \bullet H)) & \text{if } y_a(H) = F = y_a(a \bullet H),
\end{cases} \\
&= (P \triangleleft a \triangleright (R \triangleleft a \triangleright R)) \bullet H.
\end{align*}

In order to prove completeness we use a variant of basic forms, which we call $rp$-basic forms, that ‘minimizes’ on repetition-proof valuation congruence:

$\neg T$ and $F$ are $rp$-basic forms, and

$\neg P \triangleleft a \triangleright P$ is an $rp$-basic form if $P$ and $P$ are $rp$-basic forms, and if $P$ is not equal to $T$ or $F$, then either the central condition in $P$ is different from $a$, or $P$ is of the form $a \circ P'$ with $P'$ an $rp$-basic form.

Each propositional statement can in $CP_{rp}$ be proved equal to an $rp$-basic form by structural induction. For $P$ and $Q$ $rp$-basic forms, $P \equiv_{rp} Q$ implies $P \equiv Q$.

This follows in a similar way as in the proof of Lemma 6.1 because all valuations used in that proof can be seen as elements from VAs in $rp$:

(1) For the inductive case $P \equiv P \triangleleft a \triangleright P$ and $Q \equiv Q \triangleleft a \triangleright Q$, while $P \neq Q$ and for some valuation $H$ and propositional statements $R$ and $S$, $(R \triangleleft P \triangleright S)/H \neq (R \triangleleft Q \triangleright S)/H$, it is now required that for $a \bullet H' = H$ and $y_a(H') = T$,

also $y_a(a \bullet H') = T$. However, since $P$ and $Q$ are $rp$-basic forms, a possible initial $a$-occurrence in $P$ implies $P' \equiv a \circ P'$ and similar for $Q$, so this new requirement on $H'$ is not relevant and $(R \triangleleft P \triangleright S)/H' \neq (R \triangleleft Q \triangleright S)/H'$.

(2) For the last case $P \equiv P \triangleleft a \triangleright P$ and $Q \equiv Q \triangleleft b \triangleright Q$, a valuation $H$ is used that satisfies $y_a(H) = y_a(R \bullet (a \bullet H))$ for all propositional statements $R$. VAs with such a valuation exist in the variety $rp$.

Assume $P \equiv Q$, so there exist $rp$-basic forms $P'$ and $Q'$ with $CP_{rp} \vdash P \equiv P'$ and $CP_{rp} \vdash Q \equiv Q'$. By soundness, $P' \equiv Q'$ and as argued above, $P' \equiv Q'$. Hence, $CP_{rp} \vdash P \equiv Q$.

Write $CP_{cr}$ for the axioms in $CP$ and these axiom schemes ($a \in A$):

\begin{align*}
(CP_{cr1}) & \quad (x \triangleleft a \triangleright y) \triangleleft a \triangleright z = x \triangleleft a \triangleright z, \\
(CP_{cr2}) & \quad x \triangleleft a \triangleright (y \triangleleft a \triangleright z) = x \triangleleft a \triangleright z.
\end{align*}

These schemes contract for each $a \in A$ respectively the $T$-case and the $F$-case, and immediately imply CPrp1 and CPrp2.

**Theorem 6.4.** Contractive valuation congruence $=_{cr}$ is axiomatized by $CP_{cr}$.

**Proof.** Let $A$ be a VA in the variety $cr$ of contractive valuation algebras, thus for all $a \in A$,

\begin{align*}
a \bullet (a \bullet x) = a \bullet x \quad \text{and} \quad y_a(x) = y_a(a \bullet x).
\end{align*}
Concerning soundness we only check axiom scheme CPcr1: let $H \in \mathcal{A}$, then

$$((P \triangleleft a \triangleright Q) \triangleleft a \triangleright R)/H = \begin{cases} 
(P \triangleleft a \triangleright Q)/(a \bullet H) & \text{if } y_a(H) = T, \\
R/(a \bullet H) & \text{if } y_a(H) = F,
\end{cases}$$

and

$$((P \triangleleft a \triangleright Q) \triangleleft a \triangleright R) \bullet H = \begin{cases} 
(P \triangleleft a \triangleright Q) \bullet (a \bullet H) & \text{if } y_a(H) = T, \\
R \bullet (a \bullet H) & \text{if } y_a(H) = F,
\end{cases}$$

In order to prove completeness we again use a variant of basic forms, which we call $cr$-basic forms, that ‘minimizes’ on contractive valuation congruence:

$-T$ and $F$ are $cr$-basic forms, and

$-P_1 \triangleleft a \triangleright P_2$ is a $cr$-basic form if $P_1$ and $P_2$ are $cr$-basic forms, and if $P_1$ is not equal to $T$ or $F$, the central condition in $P_1$ is different from $a$.

Each propositional statement can in CP$_{cr}$ be proved equal to a $cr$-basic form by structural induction. For $P$ and $Q$ $cr$-basic forms, $P =_{cr} Q$ implies $P \equiv Q$. Again this follows in a similar way as in the proof of Lemma 6.1 because all valuations used in that proof can be seen as elements from VAs that are in $cr$:

1. For the inductive case $P \equiv P_1 \triangleleft a \triangleright P_2$ and $Q \equiv Q_1 \triangleleft a \triangleright Q_2$, while $P_1 \not= Q_1$ and for some valuation $H$ and propositional statements $R$ and $S$, $(R \triangleleft P_1 \triangleright S)/H \not= (R \triangleleft Q_1 \triangleright S)/H$, it is now required that for $a \bullet H' = H$ and $y_a(H') = T$, also $y_a(a \bullet H') = T$. However, since $P$ and $Q$ are $cr$-basic forms, $P_1$ and $Q_1$ cannot have $a$ as a central condition, so this new requirement on $H'$ is not relevant and $(R \triangleleft P \triangleright S)/H' \not= (R \triangleleft Q \triangleright S)/H'$.

2. For the last case $P \equiv P_1 \triangleleft a \triangleright P_2$ and $Q \equiv Q_1 \triangleleft b \triangleright Q_2$, a valuation $H$ is used that satisfies $y_a(H) = y_a(R \bullet (a \bullet H))$ for all propositional statements $R$. VAs with such a valuation exist in the variety $cr$.

Assume $P =_{cr} Q$, so there exist $cr$-basic forms $P'$ and $Q'$ with CP$_{cr} \vdash P = P'$ and CP$_{cr} \vdash Q = Q'$. By soundness, $P' =_{cr} Q'$ and as argued above, $P' \equiv Q'$. Hence, CP$_{cr} \vdash P = Q$.  \(\square\)

7. **Completeness for the Variety** \(wm\)

In this section we provide a complete axiomatization of weakly memorizing valuation congruence.
Write CP\textsubscript{wm} for the axioms in CP\textsubscript{cr} and these axiom schemes \((a, b \in A)\):

\begin{align*}
{(CP\textsubscript{wm}1)} & \quad (x \triangleleft a \triangleright y) \triangleleft b \triangleright z \triangleleft a \triangleright v = (x \triangleleft b \triangleright z) \triangleleft a \triangleright v, \\
{(CP\textsubscript{wm}2)} & \quad x \triangleleft a \triangleright (y \triangleleft b \triangleright (z \triangleleft a \triangleright v)) = x \triangleleft a \triangleright (y \triangleleft b \triangleright v).
\end{align*}

**Theorem 7.1.** Weakly memorizing valuation congruence \(\equiv_{wm}\) is axiomatized by CP\textsubscript{wm}.

Before proving this theorem, we define a special type of basic forms and formulate two auxiliary lemmas.

Let \(P\) be a basic form. Define \(\text{pos}(P)\) as the set of atoms that occur at the central position or at left-hand (positive) positions in \(P\):

\[\text{pos}(T) = \text{pos}(F) = \emptyset\text{ and }\text{pos}(P \triangleleft a \triangleright Q) = \{a\} \cup \text{pos}(P)\]

and define \(\text{neg}(P)\) as the set of atoms that occur at the central position or at right-hand (negative) positions in \(P\):

\[\text{neg}(T) = \text{neg}(F) = \emptyset\text{ and }\text{neg}(P \triangleleft a \triangleright Q) = \{a\} \cup \text{neg}(Q)\].

Now \(wm\)-basic forms are defined as follows:

\(-T\) and \(F\) are \(wm\)-basic forms, and

\(-P \triangleleft a \triangleright Q\) is a \(wm\)-basic form if \(P\) and \(Q\) are \(wm\)-basic forms and \(a \notin \text{pos}(P) \cup \text{neg}(Q)\).

The idea is that in a \(wm\)-basic form, as long as the evaluation of consecutive atoms keeps yielding the same reply, no atom is evaluated twice. Clearly, each \(wm\)-basic form also is a \(cr\)-basic form, but not vice versa, e.g., \(T \triangleleft a \triangleright (T \triangleleft b \triangleright (T \triangleleft a \triangleright F))\) is not a \(wm\)-basic form because \(a \in \text{neg}(T \triangleleft b \triangleright (T \triangleleft a \triangleright F))\).

**Lemma 7.2.** For each propositional statement \(P\) there is a \(wm\)-basic form \(P'\) with \(CP\textsubscript{wm} \vdash P' = P\).

**Proof.** By Lemma 3.2 we may assume that \(P\) is a basic form and we proceed by structural induction on \(P\). If \(P \equiv T\) or \(P \equiv F\) there is nothing to prove. If \(P \equiv P_1 \triangleleft a \triangleright P_2\) we may assume that \(P_i\) are \(wm\)-basic forms (if not, they can proved equal to \(wm\)-basic forms). We first consider the positive side of \(P\). If \(a \notin \text{pos}(P_1)\) we are done, otherwise we saturate \(P_1\) by replacing each atom \(b \notin a\) that occurs in a positive position with \((a \triangleleft b \triangleright F)\) using axiom \(CP\textsubscript{wm}1\). In this way we can retract each \(a\) that is in \(\text{pos}(P_1)\) (also using axiom \(CP\textsubscript{cr}1\)) and end up with \(P_1\) that does not contain \(a\) on positive positions. For example,

\begin{align*}
(((T \triangleleft a \triangleright R) \triangleleft b \triangleright S) \triangleleft c \triangleright V) & \triangleleft a \triangleright P_2 \\
= (((((T \triangleleft a \triangleright R) \triangleleft (a \triangleleft b \triangleright F) \triangleright S) \triangleleft (a \triangleleft c \triangleright F) \triangleright V) \triangleleft a \triangleright P_2 \\
= ((((((T \triangleleft a \triangleright R) \triangleleft a \triangleright S) \triangleleft b \triangleright S) \triangleleft a \triangleright V) \triangleleft c \triangleright V) \triangleleft a \triangleright P_2 \\
= (((T \triangleleft b \triangleright S) \triangleleft a \triangleright V) \triangleleft c \triangleright V) \triangleleft a \triangleright P_2 \\
= ((T \triangleleft b \triangleright S) \triangleleft c \triangleright V) \triangleleft a \triangleright P_2.
\end{align*}
Following the same procedure for the negative side of \( P \) (saturation with \((T \triangleleft a \triangleright a)\) for all \( b \neq a \) etc.) yields a \( w\text{m} \)-basic form \( P_1' \triangleleft a \triangleright P_2' \) with \( CP_{\text{wm}} \vdash P = P_1' \triangleleft a \triangleright P_2' \). \( \square \)

Recall that \( \Lambda \) is a VA in the variety \( w\text{m} \) if for all \( a, b \in A \),

\[

a \bullet (a \bullet x) = a \bullet x, \quad y_a(x) = y_a(a \bullet x),
\]

\[
y_b(a \bullet x) = y_a(x) \rightarrow (a \bullet (b \bullet (a \bullet x))) = b \bullet (a \bullet x) \land y_a(b \bullet (a \bullet x)) = y_a(x)).
\]

**Lemma 7.3.** For all \( w\text{m} \)-basic forms \( P \) and \( Q \), \( P =_{w\text{m}} Q \) implies \( P \equiv Q \).

**Proof.** The implication of the lemma follows by contraposition and nested induction on the complexity of \( w\text{m} \)-basic forms.

If \( P \equiv T \), then if \( Q \equiv F \), \( P \neq_{w\text{m}} Q \) is witnessed by each VA in which \( T \neq F \). If \( Q \equiv Q_1 \triangleleft a \triangleright Q_2 \) then consider \( \Lambda \in w\text{m} \) with \( T \neq F \) and with a valuation \( H \) that satisfies \( y_a(H) = y_b(H) = T \) and \( y_a(R \bullet (a \bullet H)) = F \) for all propositional statements \( R \). Then \( (P \circ a)/H = T \) while \( (Q \circ b)/H = F \), so \( P \neq_{w\text{m}} Q \).

If \( P \equiv F \) a similar argument applies.

If \( P \equiv P_1 \triangleleft a \triangleright P_2 \) then the cases \( Q \equiv T \) and \( Q \equiv F \) can be dealt with as above. If \( Q \equiv Q_1 \triangleleft a \triangleright Q_2 \) then assume \( P \neq Q \) because \( P_1 \neq Q_1 \). By induction, \( P_1 \neq_{w\text{m}} Q_1 \). By implication (9) there exists \( \Lambda \in w\text{m} \) with valuation \( H \) and propositional statements \( R \) and \( S \) such that \( (R \triangleleft P_1 \triangleright S)/H \neq (R \triangleleft Q_1 \triangleright S)/H \). Observe that \( a \) does not occur in \( \text{pos}(P_1) \) and \( \text{pos}(Q_1) \), and that we may assume that if \( a \) occurs in \( R \) and/or \( S \), its substitution by \( T \) preserves this inequality (otherwise replace \( a \) by \( F \triangleleft a \triangleright T \)).

Now extend \( \Lambda \) to \( \Lambda' \) with a valuation \( H' \) with \( y_a(H') = T \) and \( a \bullet H' = H \), then \( R \triangleleft P \triangleright S \) and \( R \triangleleft Q \triangleright S \) yield different results under \( H' \), hence \( P \neq_{w\text{m}} Q \). It is clear that also \( \Lambda' \in w\text{m} \). If \( P_1 \equiv Q_1 \) then \( P_2 \neq Q_2 \) and a similar argument applies.

Finally, if \( Q \equiv Q_1 \triangleleft a \triangleright Q_2 \) with \( b \neq a \) then \( P \neq_{w\text{m}} Q \) follows by considering \( \Lambda \) with \( T \neq F \) and with valuation \( H \) that satisfies for all \( R \), \( y_a(H) = y_a(R \bullet (a \bullet H)) = T \) and \( y_b(H) = y_a(R \bullet (b \bullet H)) = F \), so \( (P \circ a)/H = T \) while \( (Q \circ a)/H = F \). Clearly, \( \Lambda \in w\text{m} \). \( \square \)

**Proof of Theorem 7.1.** Let \( \Lambda \) be a VA in the variety \( w\text{m} \). The soundness of axiom \( CP_{\text{wm}1} \) and \( CP_{\text{wm}2} \) follows immediately and we only show this for the latter one: let \( H \in \Lambda \), then

\[
(P \triangleleft a \triangleright (Q \triangleleft b \triangleright (Z \triangleleft a \triangleright V))))/H
=
\begin{cases}
    P/(a \bullet H) & \text{if } y_a(H) = T, \\
    Q/(b \bullet (a \bullet H)) & \text{if } y_a(H) = F \text{ and } y_b(a \bullet H) = T, \\
    (Z \triangleleft a \triangleright V)/(b \bullet (a \bullet H)) & \text{if } y_a(H) = F \text{ and } y_b(a \bullet H) = F,
\end{cases}
\]

\[
(P/(a \bullet H))
=
\begin{cases}
    Q/(b \bullet (a \bullet H)) & \text{if } y_a(H) = F \text{ and } y_b(a \bullet H) = T, \\
    V/(b \bullet (a \bullet H)) & \text{if } y_a(H) = F \text{ and } y_b(a \bullet H) = F,
\end{cases}
\]

\[
= (P \triangleleft a \triangleright (Q \triangleleft b \triangleright V))/H,
\]
and

\[(P \triangleleft a \triangleright (Q \triangleleft b \triangleright (Z \triangleleft a \triangleright V))) \triangleleft H =
\begin{cases}
  P \triangleleft (a \triangleleft H) & \text{if } y_a(H) = T, \\
  Q \triangleleft (b \triangleleft (a \triangleleft H)) & \text{if } y_a(H) = F \text{ and } y_b(a \triangleleft H) = T, \\
  (Z \triangleleft a \triangleright V) \triangleleft (b \triangleleft (a \triangleleft H)) & \text{if } y_a(H) = F \text{ and } y_b(a \triangleleft H) = F,
\end{cases}
\]

\[= \begin{cases}
  P \triangleleft (a \triangleleft H) & \text{if } y_a(H) = T, \\
  Q \triangleleft (b \triangleleft (a \triangleleft H)) & \text{if } y_a(H) = F \text{ and } y_b(a \triangleleft H) = T, \\
  V \triangleleft (b \triangleleft (a \triangleleft H)) & \text{if } y_a(H) = F \text{ and } y_b(a \triangleleft H) = F,
\end{cases}
\]

\[= (P \triangleleft a \triangleright (Q \triangleleft b \triangleright V)) \triangleleft H.
\]

In order to prove completeness assume \(P =_{wm} Q\). By Lemma 7.2 there are \(wm\)-basic forms \(P'\) and \(Q'\) with \(CP_{wm} \vdash P = P'\) and \(CP_{wm} \vdash Q = Q'\). By soundness \(P' =_{wm} Q'\), thus by Lemma 7.3, \(P' \equiv Q'\), and thus \(CP_{wm} \vdash P = Q\). \(\square\)

8. COMPLETENESS FOR THE VARIETY mem

In this section we provide a complete axiomatization of memorizing valuation congruence.

Write \(CP_{mem}\) for the axioms in CP and this axiom:

\[(CP_{mem}) \quad x \triangleleft y \triangleright (z \triangleleft u \triangleright (v \triangleleft y \triangleright w)) = x \triangleleft y \triangleright (z \triangleleft u \triangleright w).
\]

**Theorem 8.1.** Memorizing valuation congruence \(\equiv_{mem}\) is axiomatized by \(CP_{mem}\).

Before proving this theorem, we discuss some characteristics of \(CP_{mem}\). Axiom \(CP_{mem}\) defines how the central condition \(y\) may recur in an expression. This axiom yields in combination with CP some interesting consequences. First, \(CP_{mem}\) has three symmetric variants, which all follow easily with \(x \triangleleft y \triangleright z = z \triangleleft (F \triangleleft y \triangleright T) \triangleright x\) (\(= z \triangleleft \neg y \triangleright x\)):

\[x \triangleleft y \triangleright ((z \triangleleft y \triangleright u) \triangleleft v \triangleright w) = x \triangleleft y \triangleright (u \triangleleft v \triangleright w),\]  
(10)
\[x \triangleleft y \triangleright (z \triangleleft u \triangleright v) \triangleleft u \triangleright w = (x \triangleleft y \triangleright z) \triangleleft u \triangleright w,\]  
(11)
\[(x \triangleleft y \triangleright z) \triangleleft u \triangleright v \triangleleft y \triangleright w = (x \triangleleft u \triangleright v) \triangleleft y \triangleright w.\]  
(12)

The axioms of \(CP_{mem}\) imply various laws for contraction:

\[x \triangleleft y \triangleright (v \triangleleft y \triangleright w) = x \triangleleft y \triangleright w\]  
(take \(u = F\) in \(CP_{mem}\)),
(13)
\[x \triangleleft y \triangleright (T \triangleleft u \triangleright y) = x \triangleleft y \triangleright u\]  
(take \(z = v = T\) and \(w = F\) in \(CP_{mem}\)),
(14)
\[(x \triangleleft y \triangleright z) \triangleleft y \triangleright u = x \triangleleft y \triangleright u,\]
(15)
\[(x \triangleleft y \triangleright F) \triangleleft x \triangleright z = y \triangleleft x \triangleright z,\]
(16)

and thus (take \(v = T\) and \(w = F\) in (13), respectively \(x = T\) and \(z = F\) in (15)),

\[x \triangleleft y \triangleright y = x \triangleleft y \triangleright F \quad \text{and} \quad y \triangleleft y \triangleright u = T \triangleleft y \triangleright u.
\]

The latter two equations immediately imply the following very simple contraction laws:

\[x \triangleleft x \triangleright x = x \triangleleft x \triangleright F = T \triangleleft x \triangleright x = T \triangleleft x \triangleright F = x.
\]
Let $A'$ be a subset of $A$. We employ a special type of basic forms based on $A'$: mem-basic forms over $A'$ are defined by

$-T$ and $F$ are mem-basic forms over $A'$, and

$-P \triangleleft a \triangleright Q$ is a mem-basic form over $A'$ if $a \in A'$ and $P$ and $Q$ are mem-basic forms over $A' \setminus \{a\}$.

E.g., for $A' = \{a\}$ the set of all mem-basic forms is $\{bv, bv \triangleleft a \triangleright bv' \mid bv, bv' \in \{T, F\}\}$, and for $A' = \{a, b\}$ it is

$\{bv, t_1 \triangleleft a \triangleright t_2, t_3 \triangleleft b \triangleright t_4 \mid bv \in \{T, F\}, t_1, t_2 \text{ mem-basic forms over } \{b\}, t_3, t_4 \text{ mem-basic forms over } \{a\}\}$.

**Lemma 8.2.** For each propositional statement $P$ there is a mem-basic form $P'$ with $\text{CP}_{\text{mem}} \vdash P = P'$.

**Proof.** By Lemma 3.2 we may assume that $P$ is a basic form and we proceed by structural induction on $P$. If $P \equiv T$ or $P \equiv F$ there is nothing to prove.

Assume $P \equiv P_1 \triangleleft a \triangleright P_2$. We write $[T/a]P_1$ for the term which results when $T$ is substituted for $a$ in $P_1$. We first show that $\text{CP}_{\text{mem}} \vdash P_1 \triangleleft a \triangleright P_2 = [T/a]P_1 \triangleleft a \triangleright P_2$ by induction on $P_1$: if $P_1 \equiv T$ or $P_1 \equiv F$ this is clear. If $P_1 \equiv Q \triangleleft a \triangleright R$ then $\text{CP} \vdash [T/a]P_1 = [T/a]Q$ and we derive

$$P_1 \triangleleft a \triangleright P_2 = (Q \triangleleft a \triangleright R) \triangleleft a \triangleright P_2 \equiv ([T/a]Q \triangleleft a \triangleright R) \triangleleft a \triangleright P_2 \tag{15}$$

and if $P_1 \equiv Q \triangleleft b \triangleright R$ with $b \neq a$ then $\text{CP} \vdash [T/a]P_1 = [T/a]Q \triangleleft b \triangleright [T/a]R$ and we derive

$$P_1 \triangleleft a \triangleright P_2 = (Q \triangleleft b \triangleright R) \triangleleft a \triangleright P_2 \equiv ([T/a]Q \triangleleft a \triangleright T) \triangleleft b \triangleright ([T/a]R \triangleleft a \triangleright T) \triangleleft a \triangleright P_2 \tag{16}$$

In a similar way, but now using (13), axiom CPmem and (10) instead, we find $\text{CP}_{\text{mem}} \vdash P_1 \triangleleft a \triangleright P_2 = P_1 \triangleleft a \triangleright [F/a]P_2$, and thus

$$\text{CP}_{\text{mem}} \vdash P_1 \triangleleft a \triangleright P_2 = [T/a]P_1 \triangleleft a \triangleright [F/a]P_2.$$

With axioms CP1 and CP2 we find basic forms $Q_1$ in which $a$ does not occur with $\text{CP}_{\text{mem}} \vdash Q_1 = [T/a]P_1$ and $\text{CP}_{\text{mem}} \vdash Q_2 = [F/a]P_2$.

By induction it follows that there are mem-basic forms $R_1$ and $R_2$ with $\text{CP}_{\text{mem}} \vdash R_1 = Q_1$, and hence $\text{CP}_{\text{mem}} \vdash P = R_1 \triangleleft a \triangleright R_2$ and $R_1 \triangleleft a \triangleright R_2$ is a mem-basic form. □
Next we formulate two more auxiliary lemmas. Recall that \( \mathbb{A} \) is a VA in the variety \( \text{mem} \) if for all \( a, b \in A \),

\[
\begin{align*}
  a \cdot (a \cdot x) &= a \cdot x, \\
  a \cdot (b \cdot (a \cdot x)) &= b \cdot (a \cdot x),
\end{align*}
\]

\[
\begin{align*}
  y_a(x) &= y_a(a \cdot x), \\
  y_a(b \cdot (a \cdot x)) &= y_a(x).
\end{align*}
\]

**Lemma 8.3.** For each valuation algebra \( \mathbb{A} \in \text{mem} \), and for all propositional statements \( P, Q \) and valuations \( H \in \mathbb{A} \),

\[
Q \cdot (P \cdot (Q \cdot H)) = P \cdot (Q \cdot H) \quad \text{and} \quad Q/(P \cdot (Q \cdot H)) = Q/H. \quad (17)
\]

**Proof.** By structural induction on \( Q \). The cases \( Q \equiv T \) and \( Q \equiv F \) are trivial. If \( Q \equiv a \), then apply structural induction on \( P \): the cases \( P \equiv T \) and \( P \equiv F \) are trivial, and if \( P \in A \) then \((17)\) follows by definition of \( \text{mem} \). If \( P \equiv P_1 \circ P_2 \circ P_3 \) then if \( P_2/(a \cdot H) = T \),

\[
\begin{align*}
  a \cdot ((P_1 \circ P_2 \circ P_3) \cdot (a \cdot H)) &= a \cdot (P_1 \cdot (P_2 \cdot (a \cdot H))) \\
  &= a \cdot (P_1 \cdot (a \cdot (P_2 \cdot (a \cdot H)))) \\
  &= P_1 \cdot (a \cdot (P_2 \cdot (a \cdot H))) \\
  &= P_1 \cdot (P_2 \cdot (a \cdot H)) \\
  &= (P_1 \circ P_2 \circ P_3) \cdot (a \cdot H),
\end{align*}
\]

and

\[
\begin{align*}
  a/(P_1 \circ P_2 \circ P_3) \cdot (a \cdot H)) &= a/(P_1 \cdot (P_2 \cdot (a \cdot H))) \\
  &= a/(P_1 \cdot (a \cdot (P_2 \cdot (a \cdot H)))) \\
  &= a/(P_2 \cdot (a \cdot H)) \\
  &= a/H,
\end{align*}
\]

and if \( P_2/(a \cdot H) = F \) a similar argument applies.

If \( Q \equiv Q_1 \circ Q_2 \circ Q_3 \) then first assume \( Q_2/H = T \), so \( Q \cdot H = Q_1 \cdot (Q_2 \cdot H) \). Observe that

\[
(Q_1 \circ P) \cdot (Q_2 \cdot H) = P \cdot (Q_1 \cdot (Q_2 \cdot H)),
\]

so by induction \( Q_2/(P \cdot (Q_1 \cdot (Q_2 \cdot H))) = Q_2/((Q_1 \circ P) \cdot (Q_2 \cdot H)) = Q_2/H = T \).

Using induction we further derive

\[
\begin{align*}
  Q \cdot (P \cdot (Q \cdot H)) &= Q_1 \cdot ((P \circ Q_2) \cdot (Q_1 \cdot (Q_2 \cdot H))) \\
  &= (P \circ Q_2) \cdot (Q_1 \cdot (Q_2 \cdot H)) \\
  &= Q_2 \cdot ((Q_1 \circ P) \cdot (Q_2 \cdot H)) \\
  &= P \cdot (Q_1 \cdot (Q_2 \cdot H)) \\
  &= P \cdot (Q \cdot H),
\end{align*}
\]

and

\[
\begin{align*}
  (Q_1 \circ Q_2 \circ Q_3)/(P \cdot (Q \cdot H)) &= Q_1/((P \circ Q_2) \cdot (Q_1 \cdot (Q_2 \cdot H))) \\
  &= Q_1/(Q_2 \cdot H) \\
  &= Q/H.
\end{align*}
\]

Finally, if \( Q_2/H = F \) a similar argument applies. \( \square \)
Lemma 8.4. For all mem-basic forms $P$ and $Q$, $P \equiv_{\text{mem}} Q$ implies $P \equiv Q$.

Proof. The implication of the lemma follows by contraposition and nested induction on the complexity of mem-basic forms.

If $P \equiv T$ then if $Q \equiv F$, $P \not\equiv_{\text{mem}} Q$ is witnessed by each VA in which $T \neq F$. If $Q \equiv Q_1 \triangleleft a \triangleright Q_2$ then consider $\mathbb{A} \in \text{mem}$ with $T \neq F$ and with a valuation $H$ that satisfies $y_a(H) = y_b(H) = T$ and $y_b(R \bullet (a \bullet H)) = F$ for all propositional statements $R$. Then $(P \circ b)/H = T$ while $(Q \circ b)/H = F$, so $P \not\equiv_{\text{mem}} Q$.

If $P \equiv F$ a similar argument applies.

If $P \equiv P_1 \triangleleft a \triangleright P_2$ then the cases $Q \equiv T$ and $Q \equiv F$ can be dealt with as above. If $Q \equiv Q_1 \triangleleft a \triangleright Q_2$ then assume $P \not\equiv Q$ because $P_1 \not\equiv Q_1$. By induction, $P_1 \not\equiv_{\text{mem}} Q_1$. By implication (9) there exists $\mathbb{A} \in \text{mem}$ with valuation $H$ and propositional statements $R$ and $S$ such that $(R \triangleleft P_1 \triangleright S)/H \neq (R \triangleleft Q_1 \triangleright S)/H$. Observe that $a$ does not occur in $P_1$ and $Q_1$, and that we may assume that if $a$ occurs in $R$ and/or $S$, its substitution by $T$ preserves this inequality (otherwise replace $a$ by $F \triangleleft a \triangleright T$). Now extend $\mathbb{A}$ to $\mathbb{A}'$ with a valuation $H'$ with $y_a(H') = T$ and $a \bullet H' = H$, then $R \triangleleft P \triangleright S$ and $R \triangleleft Q \triangleright S$ yield different results under $H'$, hence $P \not\equiv_{\text{mem}} Q$. Clearly, also $\mathbb{A}' \in \text{mem}$. If $P_1 \equiv Q_1$ then $P_2 \not\equiv Q_2$ and a similar argument applies.

Finally, if $Q \equiv Q_1 \triangleleft b \triangleright Q_2$ with $b \neq a$ then $P \not\equiv_{\text{mem}} Q$ follows by considering $\mathbb{A} \in \text{mem}$ with $T \neq F$ and with valuation $H$ that satisfies $y_a(H) = T$, $y_b(H) = F$, and for all propositional statements $R$, $y_a(R \bullet (b \bullet H)) = F$. Then by Lemma 8.3, $y_a(P \bullet H) = y_a(P_1 \bullet (a \bullet H)) = y_a(H)$, so $(P \circ a)/H = T$ while $(Q \circ a)/H = F$.

Proof of Theorem 8.1. Let $\mathbb{A}$ be a VA in the variety mem. In order to prove soundness we have to show that axiom CPmem holds in $\mathbb{A}$. Consider propositional statement

$$P \triangleleft Q \triangleright (R \triangleleft S \triangleright (V \triangleleft Q \triangleright W)).$$

By Lemma 8.3 we find

$$Q/H = F \implies \begin{cases} (V \triangleleft Q \triangleright W)/(S \bullet (Q \bullet H)) = W/(S \bullet (Q \bullet H)), \\ (V \triangleleft Q \triangleright W) \bullet (S \bullet (Q \bullet H)) = W \bullet (S \bullet (Q \bullet H)). \end{cases}$$

Now the soundness of axiom CPmem follows immediately:

$$(P \triangleleft Q \triangleright (R \triangleleft S \triangleright (V \triangleleft Q \triangleright W))/H)$$

$$= \begin{cases} P/(Q \bullet H) & \text{if } Q/H = T, \\ R/(S \bullet (Q \bullet H)) & \text{if } Q/H = F \text{ and } S/(Q \bullet H) = T, \\ (V \triangleleft Q \triangleright W)/(S \bullet (Q \bullet H)) & \text{if } Q/H = F \text{ and } S/(Q \bullet H) = F, \\ P/(Q \bullet H) & \text{if } Q/H = T, \\ R/(S \bullet (Q \bullet H)) & \text{if } Q/H = F \text{ and } S/(Q \bullet H) = T, \\ W/(S \bullet (Q \bullet H)) & \text{if } Q/H = F \text{ and } S/(Q \bullet H) = F, \end{cases}$$

and

\[(P \triangleleft Q \triangleright (R \triangleleft S \triangleright (V \triangleleft Q \triangleright W))) \cdot H\]

\[
\begin{cases}
  P \cdot (Q \cdot H) & \text{if } Q/H = T, \\
  R \cdot (S \cdot (Q \cdot H)) & \text{if } Q/H = F \text{ and } S/(Q \cdot H) = T, \\
  (V \triangleleft Q \triangleright W) \cdot (S \cdot (Q \cdot H)) & \text{if } Q/H = F \text{ and } S/(Q \cdot H) = F, \\
  P \cdot (Q \cdot H) & \text{if } Q/H = T, \\
  R \cdot (S \cdot (Q \cdot H)) & \text{if } Q/H = F \text{ and } S/(Q \cdot H) = T, \\
  W \cdot (S \cdot (Q \cdot H)) & \text{if } Q/H = F \text{ and } S/(Q \cdot H) = F, \\
\end{cases}
\]

= \((P \triangleleft Q \triangleright (R \triangleleft S \triangleright W)) \cdot H.\]

In order to prove completeness assume \(P =_{\text{mem}} Q\). By Lemma 8.2 there are \(\text{mem}\)-basic forms \(P'\) and \(Q'\) with \(\text{CP}_{\text{mem}} \vdash P = P'\) and \(\text{CP}_{\text{mem}} \vdash Q = Q'\). By soundness, \(P' =_{\text{mem}} Q'\), thus by Lemma 8.4, \(P' \equiv Q'\), and thus \(\text{CP}_{\text{mem}} \vdash P = Q\).

**9. COMPLETENESS FOR THE VARIETY \(st\)**

In this section we provide a complete axiomatization of static valuation congruence.

**Theorem 9.1 [Hoare 1985b].** Static valuation congruence \(=_{st}\) is axiomatized by the axioms in \(\text{CP}\) (see Table I) and these axioms:

\[(\text{CPstat}) \quad (x \triangleleft y \triangleright z) \triangleleft u \triangleright v = (x \triangleleft u \triangleright v) \triangleleft y \triangleright (z \triangleleft u \triangleright v),\]

\[(\text{CPcontr}) \quad (x \triangleleft y \triangleright z) \triangleleft y \triangleright u = x \triangleleft y \triangleright u.\]

We write \(\text{CP}_{st}\) for this set of axioms.

Observe that axiom \(\text{CPcontr}\) equals the derivable identity (15) which holds in \(\text{CP}_{\text{mem}}\). Also note that the symmetric variants of the axioms \(\text{CPstat}\) and \(\text{CPcontr}\), say

\[(\text{CPstat'}) \quad x \triangleleft y \triangleright (z \triangleleft u \triangleright v) = (x \triangleleft y \triangleright z) \triangleleft u \triangleright (x \triangleleft y \triangleright v),\]

\[(\text{CPcontr'}) \quad x \triangleleft y \triangleright (z \triangleleft y \triangleright u) = x \triangleleft y \triangleright u,\]

easily follow with identity (2), i.e., \(y \triangleleft x \triangleright z = z \triangleleft (F \triangleleft x \triangleright T) \triangleright y\), which is even valid in free valuation congruence, and that \(\text{CPcontr'} = (13)\). Thus, the axiomatization of static valuation congruence is obtained from \(\text{CP}\) by adding the axiom \(\text{CPstat}\) that prescribes for a nested conditional composition how the order of the first and a second central condition can be changed, and a generalization of the axioms \(\text{CPcr1}\) and \(\text{CPcr2}\) that prescribes contraction for terms (instead of atoms). Moreover, in \(\text{CP}_{st}\) it can be derived that

\[
x = (x \triangleleft y \triangleright z) \triangleright F \triangleright x
= (x \triangleleft F \triangleright x) \triangleright y \triangleright (z \triangleleft F \triangleright x)
= x \triangleleft y \triangleright x
= y \circ x,
\]

thus any ‘and-then’ prefix can be added to (or left out from) a propositional statement while preserving static valuation congruence, in particular \((x \triangleleft y \triangleright x = x \circ x = x)\).
\[ \neg T = F \quad (19) \]
\[ \neg F = T \quad (20) \]
\[ \neg \neg x = x \quad (21) \]
\[ \neg(x \diamond y \updownarrow z) = \neg x \diamond y \updownarrow \neg z \quad (22) \]
\[ x \diamond \neg y \updownarrow z = z \diamond y \updownarrow x \quad (23) \]

Table II. Some immediate consequences of the set of axioms CP and equation (18)

Proof of Theorem 9.1. Soundness follows from the definition of static valuations: let \( A \) be a VA that satisfies for all \( a, b \in A \),
\[ y_a(b \bullet x) = y_a(x). \]
These equations imply that for all \( P, Q \) and \( H \in A \),
\[ P/H = P/(Q \bullet H). \]
As a consequence, the validity of axioms CPstat and CPcontr follows from simple case distinctions. Furthermore, Hoare showed in [1985b] that CPstat is complete for static valuation congruence.

For an idea of a direct proof, assume \( P =_st Q \) and assume that the atoms occurring in \( P \) and \( Q \) are ordered as \( a_1, \ldots, a_n \). Then under static valuation congruence each propositional statement containing no other atoms than \( a_1, \ldots, a_n \) can be rewritten into the following special type of basic form: consider the full binary tree with at level \( i \) only occurrences of atom \( a_i \) (there are \( 2^{i-1} \) such occurrences), and at level \( n + 1 \) only leaves that are either \( T \) or \( F \) (there are \( 2^n \) such leaves). Then each series of leaves represents one of the possible propositional statements in which these atoms may occur, and the axioms in CPstat are sufficient to rewrite both \( P \) and \( Q \) into exactly one such basic form. For these basic forms, static valuation congruence implies syntactic equivalence. Hence, completeness follows.

As an aside, we note that the axioms CPcontr and CPcontr‘ immediately imply CPcr1 and CPcr2, and conversely, that each instance of these axioms is derivable from CP + CPstat + CPcr1 + CPcr2 (by induction on basic forms on \( y \)'s position), which proves completeness of this particular group of axioms.

10. ADDING NEGATION AND DEFINABLE CONNECTIVES

In this section we formally add negation and various definable connectives to CP. As stated earlier (see identity (2)), negation \( \neg x \) can be defined as follows:
\[ \neg x = F \diamond x \updownarrow T \quad (18) \]

The derivable identities in Table II play a role in the derivable connectives that we discuss below. They can be derived as follows:
\[ (19) \text{ follows from } \neg T = F \diamond T \updownarrow T = F, \]
\[ (20) \text{ follows in a similar way,} \]
\[ (21) \text{ follows from } \neg \neg x = F \diamond (F \diamond x \updownarrow T) \updownarrow T = (F \diamond F \updownarrow T) \diamond x \updownarrow (F \diamond T \updownarrow T) = T \diamond x \updownarrow F = x, \]
\[ (22) \text{ follows in a similar way,} \]

(23) follows from $x \triangleleft y \triangleright z = (z \triangleleft F \triangleright x) \triangleleft y \triangleright (z \triangleleft T \triangleright x) = z \triangleleft (F \triangleleft y \triangleright T) \triangleright x = z \triangleleft y \triangleright x$.

A definable (binary) connective already introduced is the and then operator $\circ$ with defining equation $x \circ y = y \triangleleft x \triangleright y$. Furthermore, following Bergstra et al. [1995] we write

$\circ$ for left-sequential conjunction, i.e., a conjunction that first evaluates its left argument and only after that is found $T$ carries on with evaluating its second argument (the small circle indicates which argument is evaluated first). Similar notations are used for other sequential connectives. We provide defining equations for a number of derived connectives in Table III.

The operators $\land$ and left-sequential disjunction $\lor$ are associative and the dual of each other, and so are their right-sequential counterparts. For $\circ$ a proof of this is as follows:

$$(x \circ y) \circ z = z \triangleleft (y \triangleleft x \triangleright F) \triangleright F$$

$$= (z \triangleleft y \triangleright F) \triangleleft x \triangleright (z \triangleleft F \triangleright F)$$

$$= (y \circ z) \triangleleft x \triangleright F$$

$$= x \circ (y \circ z),$$

and (a sequential version of De Morgan’s laws)

$$\neg (x \circ y) = F \triangleleft (y \triangleleft x \triangleright F) \triangleright T$$

$$= (F \triangleleft y \triangleright T) \triangleleft x \triangleright (F \triangleleft F \triangleright T)$$

$$= \neg y \triangleleft x \triangleright T$$

$$= T \triangleleft \neg x \triangleright \neg y$$

$$= \neg x \lor \neg y.$$
common definition of implication:

\[ x \rightarrow y = \neg x \lor y \quad \text{and} \quad x \rightarrow y = \neg x \lor y. \]

From now on we extend \( \Sigma_{\text{CP}}(A) \) with the “and then” operator \( \circ \), negation and all derived connectives introduced in this section, and we adopt their defining equations. Of course, it remains the case that each propositional statement has a unique basic form (cf. Lemma 3.2).

Concerning the example of the propositional statement sketched in Example (1) in Section 2:

\[ \text{look-left-and-check} \land \text{look-right-and-check} \land \text{look-left-and-check} \]

indeed precisely models part of the processing of a pedestrian planning to cross a road with two-way traffic driving on the right.

We end this section with a brief comment on these connectives in the setting of other valuation congruences. In memorizing valuation congruence, the sequential connective \( \land \) has the following properties:

1. The associativity of \( \land \) is valid,
2. The identity \( x \land y \land x = x \land y \) is valid (by equation (16)),
3. The connective \( \land \) is not commutative.

In static valuation congruence, all of \( \land, \lor, \land, \lor \) are commutative and idempotent. For example, we derive with axiom \( \text{CPstat} \) that

\[ x \land y = y \land x \land F = (T \land y \land F) \land x \land F = (T \land (y \land F) \land x \land F) = x \land y \land F = y \land x, \]

and with axiom \( \text{CPcontr} \) and its symmetric counterpart \( \text{CPcontr}' \),

\[ x \land x = x \land x \land F = (T \land x \land F) \land x \land F = T \land (x \land F) \land x \land F = x, \]
\[ x \lor x = T \land x \land x = T \land x \land (T \land x \land F) = T \land x \land F = x. \]

As a consequence, the sequential notation of these connectives is not meaningful in this case, and distributivity and absorption hold in static valuation congruence.

11. SATISFIABILITY AND COMPLEXITY

In this section we briefly consider some complexity issues. Given a variety \( K \) of valuation algebras, a propositional statement is satisfiable with respect to \( K \) if for some non-trivial \( A \in K \) (\( T_{\text{val}} \) and \( F_{\text{val}} \) are different) there exists a valuation \( H \in A \) such that

\[ P/H = T. \]

We write

\[ \text{SAT}_{K}(P) \]

if \( P \) is satisfiable. We say that \( P \) is falsifiable with respect to \( K \), notation

\[ \text{FAL}_{K}(P), \]

if and only if a valuation \( H \in A \in K \) exists with \( P/H = F \). This is the case if and only if \( \text{SAT}_{K}(F \circ P \circ T) \).
It is a well-known fact that $\text{SAT}_{st}$ is an NP-complete problem, and it is easily seen that for all propositional statements $P$, $\text{SAT}_{mem}(P) = \text{SAT}_{st}(P)$. We now argue that $\text{SAT}_{fr}$ is in P. This is the case because in the variety $fr$, both $\text{SAT}_{fr}$ and $\text{FAL}_{fr}$ can be simultaneously defined in an inductive manner: let $a \in A$ and write $\neg \text{SAT}_{fr}(P)$ to express that $\text{SAT}_{fr}(P)$ does not hold, and similar for $\text{FAL}_{fr}$, then

$$
\text{SAT}_{fr}(T), \quad \neg \text{FAL}_{fr}(T),
$$

$$
\neg \text{SAT}_{fr}(F), \quad \text{FAL}_{fr}(F),
$$

$$
\text{SAT}_{fr}(a), \quad \text{FAL}_{fr}(a),
$$

and

$$
\text{SAT}_{fr}(P \triangleleft Q \triangleright R) \text{ if } \begin{cases} 
\text{SAT}_{fr}(Q) \text{ and } \text{SAT}_{fr}(P), \\
\text{or} \\
\text{FAL}_{fr}(Q) \text{ and } \text{SAT}_{fr}(R), 
\end{cases}
$$

$$
\text{FAL}_{fr}(P \triangleleft Q \triangleright R) \text{ if } \begin{cases} 
\text{SAT}_{fr}(Q) \text{ and } \text{FAL}_{fr}(P), \\
\text{or} \\
\text{FAL}_{fr}(Q) \text{ and } \text{FAL}_{fr}(R). 
\end{cases}
$$

Hence, with respect to free valuation congruence both $\text{SAT}_{fr}(P)$ and $\text{FAL}_{fr}(P)$ are computable in polynomial time. In a similar way one can show that both $\text{SAT}_{cr}$ and $\text{SAT}_{cr}$ are in P.

Of course, many more models of CP exist than those discussed in the previous sections. For example, call a valuation \textit{positively memorizing} ($P_{mem}$) if the reply $T$ to an atom is preserved after all subsequent replies:

$$
x \triangleleft a \triangleright y = [T/a]x \triangleleft a \triangleright y
$$

for all atomic propositions $a$. In a similar way one can define \textit{negatively memorizing} valuations ($N_{mem}$):

$$
x \triangleleft a \triangleright y = x \triangleleft a \triangleright [F/a]y.
$$

Contractive or weakly memorizing valuations that satisfy $P_{mem}$ ($N_{mem}$) give rise to new models in which more propositional statements are identified.

\textbf{Theorem 11.1.} For

\[K \in \{P_{mem}, cr+P_{mem}, wm+P_{mem}, N_{mem}, cr+N_{mem}, wm+N_{mem}, mem\}\]

it holds that $\text{SAT}_{K}$ is NP-complete.

\textbf{Proof.} We only consider the case for $P_{mem}$. Then

$$
\text{SAT}_{st}(P) = \text{SAT}_{mem}(P) = \text{SAT}_{p_{mem}}(P \triangleleft d \triangleright \cdots \triangleleft d \triangleright P),
$$

where $P$ is repeated $n + 1$ times with $n$ the number of atoms occurring in $P$. Each time a $P$ evaluates to $T$ while it would not do so in $mem$, this is due to some
atom that changes the reply. So, this must be a change from $F$ to $T$, because $T$
remains $T$ in $P_{mem}$. Per atom this can happen at most once, and if each $P$ yields
$T$, then at least once without any atom taking different values. But then $P$ is also
satisfiable in $mem$. Thus the NP-complete problem $\text{SAT}_{st}(P)$ can be polynomially
reduced to $\text{SAT}_{P_{mem}}(P)$, hence $\text{SAT}_{P_{mem}}$ is NP-complete.

For $K \in \{cr+P_{mem}, \text{wm}+P_{mem}, cr+N_{mem}, \text{wm}+N_{mem}\}$, each closed term
can be written with $T, F, \neg$ and $\land$ only. For example in $cr+P_{mem}$:

$$x \triangleleft a \triangledown y = (a \land x)^{\lor} (\neg a \land y)$$

because after a positive reply to $a$ and whatever happens in $x$, the next $a$ is again
positive, so $y$ is not evaluated, and after a negative reply to $a$, the subsequent $a$
gets a negative reply because of $cr$, so then $y$ is tested. So here we see models
that identify less than $=_{mem}$ and in which each closed term can be written without
conditional composition. At first sight, this cannot be done in a uniform way (using
variables only), and it also yields a combinatoric explosion because first a rewriting
to basic form is needed. For these models $K$, $\text{SAT}_{K}$ is known to be NP-complete.

12. EXPRESSIVENESS

In this section we first show that the ternary conditional operator cannot be replaced
by $\neg$ and $\land$ and one of $T$ and $F$ (which together define $^{\lor}$) modulo free valuation
congruence. Then we show that this is the case for any collection of unary and
binary operators each of which is definable in $\Sigma_{CP}(A)$ with free valuation congruence,
and in a next theorem we lift this result to contractive valuation congruence.

Finally we observe that the conditional operator is definable with $\neg$ and $\land$ and one
of $T$ and $F$ modulo memorizing valuation congruence. We were unable to decide
whether this is also the case in weakly memorizing valuation congruence.

An occurrence of an atom $a$ in a propositional statement over $A$, $\neg$, $\land$ and $^{\lor}$
is redundant or inaccessible if it is not used along any of the possible arbitrary-valuations,
as in for example $F \land a$.

The opposite of redundancy is accessibility, which is defined thus ($acc(\phi) \subseteq A$):

$$acc(a) = \{a\},$$

$$acc(T) = \emptyset,$$

$$acc(\neg x) = acc(x),$$

$$acc(x \land y) = \begin{cases} acc(x) & \text{if } \neg \text{SAT}_{fr}(x), \\ acc(x) \cup acc(y) & \text{if } \text{SAT}_{fr}(x), \end{cases}$$

$$acc(x \lor y) = \begin{cases} acc(x) & \text{if } \neg \text{SAT}_{fr}(\neg x), \\ acc(x) \cup acc(y) & \text{if } \text{SAT}_{fr}(\neg x). \end{cases}$$

**Proposition 12.1.** Let $a \in A$, then the propositional statement $a \triangleleft a \triangledown \neg a$
cannot be expressed in $\Sigma_{CP}(A)$ with free valuation congruence using $\land$, $^{\lor}$, $\neg$,
$T$ and $F$ only.

**Proof.** Let $\psi$ be a minimal expression of $a \triangleleft a \triangledown \neg a$.
Assume $\psi \equiv \psi_0^{\lor} \psi_1$. We notice:
(1) Both \( \psi_0 \) and \( \psi_1 \) must contain \( a \): if \( \psi_0 \) contains no \( a \), it is either \( T \) and then \( \psi \) always yields \( T \) which is wrong, or \( F \) and then \( \psi \) can be simplified; if \( \psi_1 \) contains no \( a \) it is either \( T \) and then \( \psi \) always yields \( T \) which is wrong, or \( F \) and then \( \psi \) can be simplified so \( \psi \) was not minimal.

(2) \( \psi_0 \) can yield \( F \) otherwise \( \psi \) is not minimal. It will do so after using exactly one test \( a \) (yielding \( F \) without a use of \( a \) simply means that \( a \not\in acc(\psi_0) \)), yielding \( F \) after two uses of \( a \) implies that evaluation of \( \psi \) has at least three uses of \( a \) (which is wrong).

(3) If \( \psi_0 \) yields \( T \), this completes the evaluation of \( \psi \), so then the evaluation of \( \psi_0 \) involves two uses of \( a \). If \( \psi_0 \) yields \( F \), it must contain at least one use of \( a \) (but no more because otherwise evaluation of \( \psi_1 \) yields at least a third use of \( a \)).

Thus, \( \psi_0 = F \land a \land (\pi \lor T) \) or \( \psi_0 = F \land \neg a \land (\pi \lor T) \), where the \( \pi \) in the right-hand sides equals either \( a \) or \( \neg a \), and these sides take their particular form by minimality (other forms are \( T \lor \pi = T \), etc.). But both are impossible as both imply that after a first use of \( a \) the final value of \( \psi \) can be independent of the second value returned for \( a \) which is not true for \( a \land a \lor \neg a \).

For the case \( \psi \equiv \psi_0 \land \psi_1 \) a similar type of reasoning applies. \( \Box \)

Below we will prove two more general results. We first introduce some auxiliary definitions and notations because we have to be precise about definability by unary and binary operators. For \( X \) a countable set of variables, we define the following sets:

\[
T_C(X) : \text{the set of terms over } X, T, F, \land, \lor, \neg.
\]

\[
T_{\text{TND}}(X) : \text{the set of terms over } X, T, \neg, \land, \lor.
\]

\[
T^{1,2}_{C}(X) : \text{the smallest set of terms } V \text{ such that}
\]

\[
\begin{cases}
T, F \in V, \\
\text{if } x \in X \text{ and } t \in T_C(\{x\}) \text{ then } t \in V, \\
\text{if } x, y \in X \text{ and } t \in T_C(\{x, y\}) \text{ then } t \in V,
\end{cases}
\]

\( V \) is closed under substitution.

Thus \( T^{1,2}_{C}(X) \) contains the terms that can be made from unary and binary operators definable in \( T_C(X) \). For \( t \in T^{1,2}_{C}(\{x\}) \) we sometimes write \( t(x) \) instead of \( t \), and if \( s \in T^{1,2}_{C}(X) \) we write \( t(s) \) for the term obtained by substituting \( s \) for all \( x \) in \( t \). Similarly, if \( u \in T^{1,2}_{C}(\{x, y\}) \) we may write \( u(x, y) \) for \( u \) and if \( s, s' \in T^{1,2}_{C}(X) \) we write \( u(s, s') \) for the term obtained by substituting \( s \) for \( x \) and \( s' \) for \( y \) in \( u \). Finally, we define \( \#_{2p}(t) \) as the number of 2-place terms used in the definition of \( t \), i.e.,

\[
\#_{2p}(x) = \#_{2p}(T) = \#_{2p}(F) = 0,
\]

\[
\#_{2p}(t(s)) = \#_{2p}(t) + \#_{2p}(s),
\]

\[
\#_{2p}(u(s, s')) = \#_{2p}(u) + \#_{2p}(s) + \#_{2p}(s').
\]

Notice \( T_{\text{TND}}(X) \subseteq_K T^{1,2}_{C}(X) \subseteq_K T_C(X) \), where \( M \subseteq_K N \) if for each term \( t \in M \) there is a term \( r \in N \) with \( r =_K t \) for \( K \in \{fr, rp, cr, wm, mem, st\} \). We write \( \in_K \) for the membership relation associated with \( \subseteq_K \).
The sets $T_{\text{TND}}(A)$, $T_{C}^{1,2}(A)$ and $T_{C}(A)$ contain the closed substitution
instances of the respective term sets when constants from $A$ are substituted for the variables.

The set $T_{C}^{1,2}(A, X)$ contains the terms constructed from $T_{C}^{1,2}(A)$ and $T_{C}^{1,2}(X)$. For
given terms $r(x) \in T_{C}^{1,2}(A, \{x\})$ and $t \in T_{C}^{1,2}(A)$ we write $r(t)$ for the term obtained
by substituting $t$ for all $x$ in $r$ (thus $r(t) \equiv [t/x] r(x)$). We extend the definition of
$\#_{2p}(t)$ to $T_{C}^{1,2}(A, X)$ in the expected way by defining $\#_{2p}(a) = 0$ for all $a \in A$.

Clearly for all $K$,

$$T_{\text{TND}}(A) \subseteq_{K} T_{C}^{1,2}(A) \subseteq_{K} T_{C}(A).$$

From Proposition 12.1 we find that $a \not\triangleleft b \not\equiv c \not\in T_{\text{TND}}(A)$, thus

$$T_{\text{TND}}(A) \not\subseteq_{fr} T_{C}^{1,2}(A).$$

Theorem 12.2 below establishes that $T_{C}^{1,2}(A) \not\subseteq_{fr} T_{C}(A)$ as $a \not\triangleleft b \not\equiv c \not\in T_{C}^{1,2}(A)$. This
result transfers to $r p$-congruence without modification. However, in $wm$-congruence we find

$$a \not\triangleleft b \not\equiv c = \text{wm} (T \not\triangleleft b \not\equiv c) \not\triangleleft (a \not\triangleleft b \not\equiv T) \triangleright F = (-b \lor a) \land (b \lor c),$$

thus $a \not\triangleleft b \not\equiv c \in \text{wm} T_{C}^{1,2}(A)$.

**Theorem 12.2.** If $|A| > 2$ then the conditional operator cannot be expressed
modulo free valuation congruence in $T_{C}^{1,2}(X)$.

**Proof.** It is sufficient to prove that $a \not\triangleleft b \not\equiv c \not\in T_{C}^{1,2}(A)$.

Towards a contradiction, assume $t \in T_{C}^{1,2}(A)$ is a term such that $t \equiv_{fr} a \not\triangleleft b \not\equiv c$ and
$\#_{2p}(t)$ is minimal (i.e., if $u \in T_{C}^{1,2}(A)$ and $u \equiv_{fr} t$ then $\#_{2p}(u) \leq \#_{2p}(t)$).

We first argue that $t \not= f(b, t')$ for some binary function $f$ and term $t'$. Suppose
otherwise, then $b$ must be the central condition in $f(b, t')$, so $f(b, t') \equiv_{fr} g(b, t') \not\triangleleft b \not\equiv h(b, t')$ for certain binary functions $g$ and $h$ in $T_{C}^{1,2}(X)$. Because it is neither
the case that $b$ can occur as a central condition in both $g(b, t')$ and $h(b, t')$, nor that
each of these can be modulo $fr$ in $\{T, F\}$, we find

$$t \equiv_{fr} (P \not\triangleleft t' \not\equiv Q) \not\triangleleft b \not\equiv (P' \not\triangleleft t' \not\equiv Q')$$

for certain $P, P', Q, Q'$. The only possibilities left are that the central atom of $t'$ is
either $a$ or $c$, and both choices contradict $f(b, t') \equiv_{fr} a \not\triangleleft b \not\equiv c$.

So it must be the case that

$$t \equiv r(f(b, t'))$$

for some term $r(x) \in T_{C}^{1,2}(\{x\})$ such that $b$ is central in $f(b, t')$ and $x$ is central
in $r(x)$. If no such term $r(x)$ exists, then $t \equiv f'(a')$ with $f'(x)$ a unary operator
definable in $T_{C}^{1,2}(\{x\})$ and $a' \in A$, which cannot hold because $t$ needs to contain $a$, $b$ and $c$.

Also there cannot be a unary function $f' \in T_{C}^{1,2}(\{x\})$ with $r(f'(b)) \equiv_{fr} r(f(b, t'))$,
otherwise $r(f'(b)) \in T_{C}^{1,2}(A)$ while

$$\#_{2p}(r) = \#_{2p}(r(f'(b))) < \#_{2p}(r(f(b, t'))) = \#_{2p}(r) + \#_{2p}(t') + 1,$$

which contradicts the minimality of $\#_{2p}(t)$.
As \( x \) is central in \( f(x, y) \) we may write

\[
f(x, y) =_{fr} g(x, y) \triangleleft x \triangleright h(x, y)
\]

for certain binary functions \( g \) and \( h \) in \( T^{1,2}_C(X) \). Because \( b \) is central in \( t \) we find

\[
t =_{fr} r(g(b, t') \triangleleft b \triangleright h(b, t')).
\]

We proceed with a case distinction on the form that \( g(b, t') \) and \( h(b, t') \) may take. At least one of these is modulo \( fr \) not equal to \( T \) or \( F \) (otherwise \( f(b, t') \) could be replaced by \( f'(b) \) for some unary function \( f' \) and this was excluded above).

1. Suppose \( g(b, t') \not\in_{fr} \{T, F\} \) and \( h(b, t') \not\in_{fr} \{T, F\} \). First notice that \( b \) cannot occur as a central condition in both \( g(b, t') \) and in \( h(b, t') \). So, both \( g(x, y) \) and \( h(x, y) \) can be written as a conditional composition with \( y \) as the central variable, and we find

\[
t =_{fr} r((P \triangleleft a' \triangleright Q) \triangleleft b \triangleright (P' \triangleleft a' \triangleright Q'))
\]

for certain closed terms \( P, Q, P', Q' \). By supposition \( t' \not\in_{fr} \{T, F\} \), and the only possibilities left are that its central atom equals both \( a \) and \( c \), which clearly is impossible.

2. We are left with four cases: either \( a \) is central in \( g(b, t') \) and \( h(b, t') \in_{fr} \{T, F\} \), or \( c \) is central in \( h(b, t') \) and \( g(b, t') \in_{fr} \{T, F\} \). These cases are symmetric and it suffices to consider only the first one, the others can be dealt with similarly. So assume \( a \) is central in \( g(b, t') \) and \( h(b, t') =_{fr} T \), hence

\[
g(b, t') =_{fr} P \triangleleft a \triangleright Q \quad \text{for some } P, Q \in \{T, F\}.
\]

We find

\[
t =_{fr} r((P \triangleleft a \triangleright Q) \triangleleft b \triangleright T),
\]

and we distinguish two cases:

i. \( P \equiv T \) or \( Q \equiv T \). Now a central \( c \) can be reached after a negative reply to \( b \). But this central \( c \) can also be reached after a positive reply to \( b \) and the appropriate reply to \( a \), which contradicts free congruence with \( a \triangleleft b \triangleright c \).

ii. \( P \equiv Q \equiv F \). Then the reply to \( a \) in \( r((F \triangleleft a \triangleright F) \triangleleft b \triangleright T) \) is not used, which also contradicts free congruence with \( a \triangleleft b \triangleright c \).

This concludes our proof. \( \square \)

We will now argue that \( a \triangleleft b \triangleright c \not\in_{cr} T^{1,2}_C(A) \). We will make use of additional operators \( T_a \) and \( F_a \) for each atom \( a \in A \), defined for all \( b \in A \) and terms \( t, r \in T^{1,2}_C(A) \) by

\[
\begin{align*}
T_a(T) &= T, & F_a(T) &= F, \\
T_a(F) &= T, & F_a(F) &= F, \\
T_a(t \triangleleft b \triangleright r) &= t \triangleleft b \triangleright r \quad \text{if } a \not\in b, & F_a(t \triangleleft b \triangleright r) &= t \triangleleft b \triangleright r \quad \text{if } a \not\in b, \\
T_a(t \triangleleft a \triangleright r) &= T_a(t), & F_a(t \triangleleft a \triangleright r) &= F_a(r).
\end{align*}
\]

Observe that \( T_a \) (\( F_a \)) simplifies a term \( t \) as if it is a subterm of \( a \triangleleft t \) with the additional knowledge that the reply on \( a \) has been \( T \). We notice that

\[
t \triangleleft a \triangleright r =_{cr} T_a(t) \triangleleft a \triangleright F_a(r).
\]
We define a term \( P \) to have the property \( \phi_{a,b,c} \) if

- the central atom of \( T_b(P) \) equals \( a \), \( T_a(T_b(P)) \in cr \{ T, F \} \) and \( T_a(T_b(P)) \neq cr F_a(T_b(P)) \),

- the central atom of \( F_b(P) \) equals \( c \), \( T_c(T_b(P)) \in cr \{ T, F \} \) and \( F_c(T_b(P)) \in cr \{ T, F \} \), and \( T_c(T_b(P)) \neq cr F_c(T_b(P)) \).

Typically, \( a \triangleleft b ⊶ c \) has property \( \phi_{a,b,c} \).

**Theorem 12.3.** If \( |A| > 2 \) then the conditional operator cannot be expressed modulo contractive valuation congruence in \( T^{1,2}_C(X) \).

**Proof.** Let \( a, b, c \in A \). It is sufficient to show that no term in \( T^{1,2}_C(A) \) has property \( \phi_{a,b,c} \). A detailed proof of this fact is included in Appendix A. \( \Box \)

Finally, we observe that \( x \triangleleft y ⊶ z \) is expressible in CP using \( \land \) and \( \neg \) only: first \( \lor \) is expressible, and

\[
\text{CP}_{\text{mem}} \vdash (y \land x) \lor (\neg y \land z) = T \triangleleft (x \triangleleft y \triangleright F) \triangleright (z \triangleleft (F \triangleleft y \triangleright T) \triangleright F) = T \triangleleft (x \triangleleft y \triangleright F) \triangleright (F \triangleleft y \triangleright z) = (T \triangleleft x \triangleright F) \triangleright (F \triangleleft y \triangleright z) \quad (11) = (T \triangleleft y \triangleright z) \quad (13) = x \triangleleft y \triangleright z.
\]

Thus, for \( x, y, z \in X \) it holds that \( (x \triangleleft y \triangleright z) \in mem T^{1,2}_C(X) \). We leave it as an open question whether \( (x \triangleleft y \triangleright z) \in wm T^{1,2}_C(X) \).

### 13. PROJECTIONS AND THE PROJECTIVE LIMIT MODEL

In this section we introduce the *projective limit model* \( \mathcal{A}_\infty \) which contains finite as well as infinite propositions. A simple, intuitive example of a potentially infinite proposition is

\[
\text{while } \neg a \text{ test } b,
\]

which as long as \( a \) yields \( F \) tests \( b \). An infinite proposition such as this one can be meaningful in free or reactive valuation semantics. In the next section we return to this example.

Let \( \mathcal{P} \) be the domain of the initial algebra of CP. We assume that each element in \( \mathcal{P} \) is represented by its (equivalent) basic form, but we shall often write \( a \) for \( T \triangleleft a \triangleright F \). Let \( \mathbb{N}^+ \) denote \( \mathbb{N} \setminus \{0\} \). We first define a so-called *projection* operator

\[
\pi : \mathbb{N}^+ \times \mathcal{P} \rightarrow \mathcal{P},
\]

which will be used to finitely approximate every proposition in \( \mathcal{P} \). We further write

\[
\pi_n(P)
\]

instead of \( \pi(n, P) \). The defining equations for the \( \pi_n \)-operators are these \((n \in \mathbb{N}^+)\):

\[
\begin{align*}
\pi_n(T) &= T, \\
\pi_n(F) &= F, \\
\pi_1(x \triangleleft a \triangleright y) &= a, \\
\pi_{n+1}(x \triangleleft a \triangleright y) &= \pi_n(x) \triangleleft a \triangleright \pi_n(y),
\end{align*}
\]

for all \( a \in A \). We write \( \text{PR} \) for this set of equations. It follows by structural induction on representatives that for each proposition \( P \in \mathcal{P} \) there exists \( n \in \mathbb{N}^+ \) such that for all \( j \in \mathbb{N} \),

\[ \pi_{n+j}(P) = P. \]

We state without proof that \( \text{CP} + \text{PR} \) is a conservative extension of \( \text{CP} \) and mention the following derivable identities in \( \text{CP} + \text{PR} \) for \( a \in A \) and \( n \in \mathbb{N}^+ \):

\[
\begin{align*}
\pi_n(a) &= \pi_n(T \triangleleft a \triangleright F) = a, \\
\pi_{n+1}(a \triangleright x) &= a \triangleright \pi_n(x).
\end{align*}
\]

The following lemma establishes how the arguments of the projection of a conditional composition can be restricted to certain projections, in particular

\[ \pi_n(P \triangleleft Q \triangleright R) = \pi_n(\pi_n(P) \triangleleft \pi_n(Q) \triangleright \pi_n(R)), \]

which is a property that we will use in the definition of our projective limit model.

**Lemma 13.1.** For all \( P, Q, R \in \mathcal{P} \) and all \( n \in \mathbb{N}^+ \), \( k, \ell, m \in \mathbb{N} \),

\[ \pi_n(P \triangleleft Q \triangleright R) = \pi_n(\pi_{n+k}(P) \triangleleft \pi_{n+\ell}(Q) \triangleright \pi_{n+m}(R)). \]

**Proof.** We may assume that \( Q \) is a basic form and we apply structural induction on \( Q \).

If \( Q \equiv T \) then we have to prove that for all \( n \in \mathbb{N}^+ \) and \( k \in \mathbb{N} \),

\[ \pi_n(P) = \pi_n(\pi_{n+k}(P)). \]

We apply structural induction on \( P \). If \( P \in \{ T, F \} \) we are done. If \( P \equiv P_1 \triangleleft a \triangleright P_2 \) then we proceed by induction on \( n \). The case \( n = 1 \) is trivial, and

\[
\begin{align*}
\pi_{n+1}(P) &= \pi_{n+1}(P_1 \triangleleft a \triangleright P_2) \\
&= \pi_n(P_1) \triangleleft a \triangleright \pi_n(P_2) \\
&= \pi_n(\pi_{n+k}(P_1)) \triangleleft a \triangleright \pi_n(\pi_{n+k}(P_2)) \\
&= \pi_{n+1}(\pi_{n+k}(P_1) \triangleleft a \triangleright \pi_{n+k}(P_2)) \\
&= \pi_{n+1}(\pi_{n+k+1}(P)).
\end{align*}
\]

If \( Q \equiv F \): similar.

If \( Q \equiv Q_1 \triangleleft a \triangleright Q_2 \) then we proceed by induction on \( n \). The case \( n = 1 \) is trivial,
and

\[ \pi_{n+1}(P \triangleleft Q \triangleright R) = \pi_{n+1}(P \triangleleft (Q_1 \triangleleft a \triangleright Q_2) \triangleright R) = \\pi_{n+1}((P \triangleleft Q_1 \triangleright R) \triangleleft a \triangleright (P \triangleleft Q_2 \triangleright R)) = \pi_n(P \triangleleft Q_1 \triangleright R) \triangleleft a \triangleright \pi_n(P \triangleleft Q_2 \triangleright R) \]

\[ \]

IH

\[ = \pi_n(\pi_{n+k+1}(P) \triangleleft \pi_{n+\ell}(Q_1) \triangleright \pi_{n+m+1}(R)) \triangleleft a \triangleright \pi_n(\pi_{n+k+1}(P) \triangleleft \pi_{n+\ell}(Q_2) \triangleright \pi_{n+m+1}(R)) = \pi_{n+1}(\pi_{n+k+1}(P) \triangleleft \pi_{n+\ell}(Q_1) \triangleright \pi_{n+m+1}(R)) \triangleleft a \triangleright \pi_n(\pi_{n+k+1}(P) \triangleleft \pi_{n+\ell}(Q_2) \triangleright \pi_{n+m+1}(R)) = \pi_{n+1}(\pi_{n+k+1}(P) \triangleleft (\pi_{n+\ell}(Q_1) \triangleleft a \triangleright \pi_{n+\ell}(Q_2)) \triangleright \pi_{n+m+1}(R)) = \pi_{n+1}(\pi_{n+1+k}(P) \triangleleft \pi_{n+1+\ell}(Q) \triangleright \pi_{n+1+m}(R)). \]

\[ \]

The projective limit model \( \mathcal{A}^\infty \) is defined as follows:

— The domain of \( \mathcal{A}^\infty \) is the set of projective sequences \( (P_n)_{n \in \mathbb{N}^+} \): these are all sequences with the property that all \( P_n \) are in \( \mathcal{P} \) and satisfy

\[ \pi_n(P_{n+1}) = P_n, \]

so that they can be seen as successive projections of the same infinite proposition (observe that \( \pi_n(P_n) = P_n \)). We further write \( (P_n)_n \) instead of \( (P_n)_{n \in \mathbb{N}^+} \).

— Equivalence of projective sequences in \( \mathcal{A}^\infty \) is defined component-wise, thus

\[ (P_n)_n = (Q_n)_n \]

if for all \( n, P_n = Q_n \).

— The constants \( T \) and \( F \) are interpreted in \( \mathcal{A}^\infty \) as the projective sequences that consist solely of these respective constants.

— An atomic proposition \( a \) is interpreted in \( \mathcal{A}^\infty \) as the projective sequence

\[ (a, a, a, \ldots). \]

— Projection in \( \mathcal{A}^\infty \) is defined component-wise, thus \( \pi_k((P_n)_n) = (\pi_k(P_n))_n \).

— Conditional composition in \( \mathcal{A}^\infty \) is defined using projections:

\[ (P_n)_n \triangleleft (Q_n)_n \triangleright (R_n)_n = (\pi_n(P_n \triangleleft Q_n \triangleright R_n))_n. \]

The projections are needed if the depth of a component \( P_n \triangleleft Q_n \triangleright R_n \) exceeds \( n \).

Equation (28) implies that this definition indeed yields a projective sequence:

\[ \pi_n(\pi_{n+1}(P_{n+1} \triangleleft Q_{n+1} \triangleright R_{n+1})) = \pi_n(P_{n+1} \triangleleft Q_{n+1} \triangleright R_{n+1}) = \pi_n(\pi_{n}(P_{n+1} \triangleleft Q_{n+1} \triangleright R_{n+1})) = \pi_n(P_n \triangleleft Q_n \triangleright R_n). \]

The following result can be proved straightforwardly:

**Theorem 13.2.** \( \mathcal{A}^\infty \models CP + PR. \)

The projective limit model \( \mathcal{A}^\infty \) contains elements that are not the interpretation of finite propositions in \( \mathcal{P} \) (in other words, elements of infinite depth). In the next section we discuss some examples.
14. RECURSIVE SPECIFICATIONS

In this section we discuss recursive specifications over $\Sigma_{CP}(A)$, which provide an alternative and simple way to define propositions in $\mathbb{A}^{\infty}$. We first restrict ourselves to a simple class of recursive specifications: Given $\ell > 0$, a set

$$E = \{X_i = t_i \mid i = 1, \ldots, \ell\}$$

of equations is a linear specification over $\Sigma_{CP}(A)$ if

$$t_i ::= T \mid F \mid X_j \triangleleft a \triangleright X_k$$

for $i, j, k \in \{1, \ldots, \ell\}$ and $a_i \in A$. A solution for $E$ in $\mathbb{A}^{\infty}$ is a series of propositions

$$(P_{1,n})_{n}, \ldots, (P_{\ell,n})_{n}$$

such that $(P_{i,n})_{n}$ solves the equation for $X_i$. In $\mathbb{A}^{\infty}$, solutions for linear specifications exist. This follows from the property that for each $m \in \mathbb{N}^+$, $\pi_m(X_i)$ can be computed as a proposition in $P$ by replacing variables $X_j$ by $t_j$ sufficiently often. For example, if

$$E = \{X_1 = X_3 \triangleleft a \triangleright X_2, \ X_2 = b \circ X_1, \ X_3 = T\}$$

we find $\pi_m(X_3) = \pi_m(T) = T$ for all $m \in \mathbb{N}^+$, and

$$\begin{align*}
\pi_1(X_2) &= \pi_1(b \circ X_1) = b, \\
\pi_{m+1}(X_2) &= \pi_{m+1}(b \circ X_1) = b \circ \pi_m(X_1), \\
\pi_1(X_1) &= \pi_1(X_3 \triangleleft a \triangleright X_2) = a, \\
\pi_{m+1}(X_1) &= \pi_{m+1}(X_3 \triangleleft a \triangleright X_2) = T \triangleleft a \triangleright \pi_m(X_2),
\end{align*}$$

and we can in this way construct a projective sequence per variable. We state without proof that for a linear specification $E = \{X_i = t_i \mid i = 1, \ldots, \ell\}$ such sequences model unique solutions in $\mathbb{A}^{\infty}$, and we write

$$\langle X_i | E \rangle$$

for the solution of $X_i$ as defined in $E$. In order to reason about linearly specified propositions, we add these constants to the signature $\Sigma_{CP}$. These constants satisfy the equations

$$\langle X_i | E \rangle = \langle t_i | E \rangle$$

where $\langle t_i | E \rangle$ is defined by replacing each $X_j$ in $t_i$ by $\langle X_j | E \rangle$. The proof principle introducing these identities is called the Recursive Definition Principle (RDP), and for linear specifications RDP is valid in the projective limit model $\mathbb{A}^{\infty}$.\footnote{The domain of $\mathbb{A}^{\infty}$ can be turned into a metric space by defining $d((P_n)_n, (Q_n)_n) = 2^{-n}$ for $n$ the least value with $P_n \neq Q_n$. The existence of unique solutions for linear specifications then follows from Banach’s fixed point theorem; a comparable and detailed account of this fact can be found in Vu [2008].}

\footnote{A nice and comparable account of the validity of RDP in the projective limit model for ACP is given in Baeten and Weijland [1990]. In that text book, a sharp distinction is made between RDP—stating that certain recursive specifications have at least a solution per variable—and the Recursive Specification Principle (RSP), stating that they have at most one solution per variable. The uniqueness of solutions per variable then follows by establishing the validity of both RDP and RSP.} As
illustrated above, all solutions satisfy
\[ \langle X_i | E \rangle = (\pi_n(\langle X_i | E \rangle))_n. \]

Some examples of propositions defined by recursive specifications are these:

(1) For \( E = \{ X_1 = X_2 \triangleleft a \triangleright X_3, \ X_2 = T, \ X_3 = F \} \) we find
\[ \langle X_1 | E \rangle = (a, a, a, \ldots) \]
which in the projective limit model represents the atomic proposition \( a \). Indeed, by RDP we find \( \langle X_1 | E \rangle = \langle X_2 | E \rangle \triangleleft a \triangleright \langle X_3 | E \rangle = T \triangleleft a \triangleright F = a \).

(2) For \( E = \{ X_1 = X_2 \triangleleft a \triangleright X_3, \ X_2 = T, \ X_3 = T \} \) we find
\[ \langle X_1 | E \rangle = (a \circ T, a \circ T, a \circ T, \ldots) \]
which in the projective limit model represents \( a \circ T \). By RDP we find \( \langle X_1 | E \rangle = a \circ T \).

(3) For \( E = \{ X_1 = X_3 \triangleleft a \triangleright X_2, \ X_2 = b \circ X_1, \ X_3 = T \} \) as discussed above, we find
\[ \langle X_1 | E \rangle = (a, T \triangleleft a \triangleright b, T \triangleleft a \triangleright b \circ a, T \triangleleft a \triangleright b \circ (T \triangleleft a \triangleright b), \ldots) \]
which in the projective limit model represents an infinite propositional statement, that is, one that satisfies
\[ \pi_i(\langle X_1 | E \rangle) = \pi_j(\langle X_1 | E \rangle) \quad \Rightarrow \quad i = j, \]
and thus has infinite depth. By RDP we find \( \langle X_1 | E \rangle = T \triangleleft a \triangleright b \circ \langle X_1 | E \rangle \). We note that the infinite propositional statement \( \langle X_1 | E \rangle \) can be characterized as
\[ \text{while } \neg a \text{ test } b. \]

An example of a projective sequence that cannot be defined by a linear specification, but that can be defined by the infinite linear specification \( I = \{ X_i = t_i \mid i \in \mathbb{N}^+ \} \) with
\[ t_i = \begin{cases} a \circ X_{i+1} & \text{if } i \text{ is prime}, \\ b \circ X_{i+1} & \text{otherwise}, \end{cases} \]
is \( \langle X_1 | I \rangle \), satisfying
\[ \langle X_1 | I \rangle = (b, b \circ a, b \circ a \circ a, b \circ a \circ a \circ b, b \circ a \circ a \circ b \circ a, \ldots). \]

Other examples of projective sequences that cannot be defined by a finite linear specification are \( \langle X_j | I \rangle \) for any \( j > 1 \).

Returning to Example (1) of a propositional statement sketched in Section 2, we can be more explicit now: the recursively defined proposition \( \langle X_1 | E \rangle \) with \( E \) containing
\[ X_1 = X_2 \triangleleft \text{green-light} \triangleright X_1, \]
\[ X_2 = X_3 \triangleleft (\text{look-left-and-check} \land \text{look-right-and-check} \land \text{look-left-and-check}) \triangleright X_1, \]
\[ X_3 = \ldots \]
models in a straightforward way a slightly larger part of the processing of a pedestrian planning to cross a road with two-way traffic driving on the right.
15. CONCLUSIONS

Proposition algebra in the form of CP for propositional statements with conditional composition and either enriched or not with negation and sequential connectives, is proposed as an abstract data type. Free valuations provide the natural semantics for CP and these are semantically at the opposite end of static valuations. It is shown that taking conditional composition and free valuations as a point of departure implies that a ternary connective is needed for functional completeness; binary connectives are not sufficient. Furthermore, CP admits a meaningful and non-trivial extension to projective limits, and this constitutes the most simple case of an inverse limit construction that we can think of.

The potential role of proposition algebra is only touched upon by some examples. It remains a challenge to find convincing examples that require reactive valuations, and to find earlier accounts of this type of semantics for propositional logic. The basic idea of proposition algebra with free and reactive valuations can be seen as a combination of the following two ideas:

—Consider atomic propositions as events (queries) that can have a side effect in a sequential system, and take McCarthy’s sequential evaluation [1963] to two-valued propositional logic; this motivates reactive valuations as those that define evaluation or computation as a sequential phenomenon.

—In the resulting setting, Hoare’s conditional composition [1985b] is more natural than the sequential, non-commutative versions of conjunction and disjunction, and (as it appears) more expressive: a ternary connective is needed anyhow.

For conditional composition we have chosen for the notation

\[ \neg \triangleleft \neg \triangleleft \neg \]

from Hoare [1985b] in spite of the fact that our theme is technically closer to thread algebra [Bergstra and Middelburg 2007] where a different notation is used. We chose for the notation \( \neg \triangleleft \neg \triangleleft \neg \) because its most well-known semantics is static valuation semantics (which is simply conventional propositional logic) for which this notation was introduced in Hoare [1985b].\(^3\) To some extent, thread algebra and propositional logic in the style of Hoare [1985b] are models of the same signature. A much more involved use of conditional composition can be found in Ponse and van der Zwaag [2007], where the propositional fragment of Belnap’s four-valued logic [1977] is characterized using only conditional composition and his four constants representing these truth values.

Apart from the valuation congruences introduced in Section 5 and variations thereof such as \( P_{\text{mem}} \) which was briefly discussed in Section 11, many more valuation congruences can be distinguished. As an example we mention here \( \text{mem}_k \) with \( k \in \mathbb{N} \): valuations that remember the result of each atomic evaluation during \( k \) subsequent atomic evaluations. The valuation congruence defined by \( \text{mem}_k \) can be seen as a natural generalization of repetition-proof valuation congruence

\(^3\)This notation was used by Hoare in his 1985 book on CSP [1985a] and in his well-known 1987 paper Laws of Programming [Hayes et al. 1987] for expressions \( P \triangleleft b \triangleleft Q \) with \( P \) and \( Q \) programs and \( b \) a Boolean expression without mention of [Hoare 1985b] that appeared in 1985.
\( (=_{rp}) \): \( \text{mem}_1 \)-congruence coincides with \( =_{rp} \) and \( \text{mem}_0 \)-congruence coincides with free valuation congruence.

In this paper we assumed that \(|A| > 1\). The case that \(|A| = 1\) is in detail described in Regenboog [2010]. In particular, \( =_{rp} \) and \( =_{st} \) and thus all valuation congruences in between coincide in this case.

**Related work.** We end with a few notes on related matters.

(1) In quite a few papers the ‘lazy evaluation’ semantics proposed in McCarthy’s work on conditional expressions in [1963] is discussed, or taken as a point of departure. We mention a few of these works in reverse chronological order:

(a) Hähnle states in his paper *Many-valued logic, partiality, and abstraction in formal specification languages* [2005] that

“sequential conjunction […] represents the idea that if the truth value can be determined after evaluation of the first argument, then the result is computed without looking at the second argument. Many programming languages contain operators that exhibit this kind of behavior”.

(b) Konikowska [1996] describes a model of so-called McCarthy algebras in terms of three-valued logic, while restricting to the well-known symmetric binary connectives, and provides sound axiomatizations and representation results. This is achieved by admitting only \( T \) and \( F \) as constants in a McCarthy algebra, and distinguishing an element \( a \) as in one of four possible classes (‘positive’ if \( a \lor x = a \), ‘negative’ if \( a \land x = a \), ‘defined’ if \( a \land \neg a = F \), and ‘strictly undefined’ if \( a = \neg a \)).

(c) Finally, Bloom and Tindell discuss in their paper *Varieties of “if-then-else”* [1983] various modelings of conditional composition, both with and without a truth value \( \text{undefined} \), while restricting to the “redundancy law”

\[
(x \triangleleft y \downarrow z) \downarrow y \uparrow u = x \triangleleft y \downarrow u,
\]

a law that we called \( \text{CPcontr} \) in Section 9 and that generalizes the axiomatization of contractive valuation congruence defined in that section to an extent in which only the difference between \( T \), \( F \) and \( \text{undefined} \) plays a significant role.

As far as we can see, none of the papers mentioned here even suggests the idea of free or reactive valuation semantics. Another example where sequential operators play a role is *Quantum logic* as formulated by [Rehder 1980; Mittelstaedt 2004], where next to normal conjunction a notion of sequential conjunction \( \sqcap \) is exploited that is very similar to \( \triangledown \) (and that despite its notation is certainly not symmetric).

(2) Concerning projections and the projective limit model \( \mathcal{A}^\infty \) we mention that in much current research and exposition, projections are defined also for depth 0 (see, e.g., [Bergstra and Middelburg 2007; Vu 2008] for the case of thread algebra, and [Fokkink 2000] for process algebra). However, CP does not have a natural candidate for \( \pi_0(P) \) and therefore we stick to the original approach as described in Bergstra and Klop [1984] (and overviewed in Baeten and Weijland [1990]) that starts from projections with depth 1.
(3) Free valuation semantics was in a different form employed in accounts of process algebra with propositional statements: in terms of operational semantics, this involves transitions

\[ P \xrightarrow{a,w} Q \]

for process expressions \( P \) and \( Q \) with \( a \) an action and \( w \) ranging over a class of valuations. In particular this approach deals with process expressions that contain propositional statements in the form of guarded commands, such as \( \phi :\rightarrow P \) that has a transition

\[(\phi :\rightarrow P) \xrightarrow{a,w} Q\]

if \( P \xrightarrow{a,w} Q \) and \( w(\phi) = T \). For more information about this approach, see, e.g., Bergstra and Ponse [1998b; 1998a].

A. APPENDIX

In this appendix we provide a detailed proof of Theorem 12.3.

**Proof of Theorem 12.3.** This proof has the same structure as the proof of Theorem 12.2, but a few cases require more elaboration.

Towards a contradiction, assume that \( t \in T_{C}^{1,2}(A) \) is a term with property \( \phi_{a,b,c} \) and \( \#_{2p}(t) \) is minimal.

We first argue that \( t \not\equiv r(f(b,t')) \) for some binary function \( f \) and term \( t' \). Suppose otherwise, then \( b \) must be the central condition in \( f(b,t') \), so \( f(b,t') = c_{r}g(b,t') \triangleq b \triangleright h(b,t') \) for certain binary functions \( g \) and \( h \) in \( T_{C}^{1,2}(X) \). Notice that because \( b \) is not central in \( g(b,t') \), a different atom must be central in this term, and this atom must be \( a \). For this to hold, \( a \) must be central in \( f(b,t') \) and no atom different from \( a \) can be tested by the first requirement of \( \phi_{a,b,c} \). So, after contraction of all further \( a \)'s we find

\[ T_{b}(t') = c_{r} P \triangleleft a \triangleright Q \]

with \( P, Q \in \{ T, F \} \), and similarly

\[ F_{b}(t') = c_{r} P' \triangleleft c \triangleright Q' \]

with \( P', Q' \in \{ T, F \} \). If \( P \not\equiv Q \) and \( P' \not\equiv Q' \), then \( t' \) is a term that satisfies \( \phi_{a,b,c} \), but \( t' \) is a term with lower \( \#_{2p} \)-value than \( g(b,t') \triangleq b \triangleright h(b,t') \), which is a contradiction. If either \( P \equiv Q \) or \( P' \equiv Q' \), then

\[ t \equiv c_{r} (P \triangleleft a \triangleright Q) \triangleleft b \triangleright (P' \triangleleft c \triangleright Q') \]

which contradicts \( \phi_{a,b,c} \).

So it must be the case that

\[ t \equiv r(f(b,t')) \]

for some term \( r(x) \in T_{C}^{1,2}(x) \) such that \( b \) is central in \( f(b,t') \) and \( x \) is central in \( r(x) \). If no such such term \( r(x) \) exists, then \( t \equiv f'(a') \) with \( f'(x) \) a unary operator definable in \( T_{C}^{1,2}(x) \) and \( a' \in A \), which cannot hold because \( t \) needs to contain \( a \), \( b \) and \( c \).
Also there cannot be a unary function \( f' \in T^{1,2}_C(\{x\}) \) with \( \rho(f'(b)) =_c \rho(f(b, t')) \), otherwise \( \rho(f'(b)) \in T^{1,2}_C(A) \) while \( \#_2\rho(r(f'(b))) < \#_2\rho(r(f(b, t'))) \), which is a contradiction.

As \( x \) is central in \( f(x, y) \) we may write
\[
f(x, y) =_c g(x, y) \circ x \triangleright h(x, y)
\]
for binary operators \( g \) and \( h \). Because \( b \) is central in \( t \) we find
\[
t =_c \rho(T_b(g(b, t')) \circ b \triangleright F_b(h(b, t'))).
\]

We proceed with a case distinction on the form that \( T_b(g(b, t')) \) and \( F_b(h(b, t')) \) may take. At least one of these is modulo \( c \) not equal to \( T \) or \( F \) (otherwise \( f(b, t') \) could be replaced by \( f'(b) \) for some unary function \( f' \) and this was excluded above).

1. Suppose \( T_b(g(b, t')) \not\equiv_c \{T, F\} \) and \( F_b(h(b, t')) \not\equiv_c \{T, F\} \). We show that this is not possible: first notice that because \( b \) is not central in \( T_b(g(b, t')) \), a different atom must be central in this term, and this atom must be \( a \). For this to hold, \( a \) must be central in \( T_b(t') \) and no atom different from \( a \) can be tested by the first requirement of \( \phi_{a,b,c} \). So, after contraction of all further \( a \)'s we find
\[
T_b(t') =_c P \circ a \triangleright Q
\]
with \( P, Q \in \{T, F\} \), and similarly \( F_b(t') =_c P' \circ c \triangleright Q' \) with \( P', Q' \in \{T, F\} \).

If \( P \not\equiv Q \) and \( P' \not\equiv Q' \), then \( t' \) is a term that satisfies \( \phi_{a,b,c} \), but \( t' \) is a term with lower \( \#_2\rho \)-value than \( \rho(g(b, t') \circ b \triangleright h(b, t')) \), which is a contradiction. Assume \( P \equiv Q \) (the case \( P' \equiv Q' \) is symmetric).

Now \( t =_c \rho(T_b(g(b, t')) \circ b \triangleright F_b(h(b, t'))) \), and no \( b \)'s can occur in \( T_b(g(b, t')) \), so
\[
T_b(g(b, t')) \in \{P \circ a \triangleright Q, \ F \circ (P \circ a \triangleright Q) \circ T, \ (P \circ a \triangleright Q) \circ T, \ (P \circ a \triangleright Q) \circ F\}.
\]

For \( F_b(h(b, t')) \) a similar argument applies, which implies that (recall \( P \equiv Q \))
\[
T_b(g(b, t')) =_c a \circ P \text{ and } F_b(h(b, t')) =_c P' \circ c \triangleright Q' \text{ with } P, P', Q' \in \{T, F\}.
\]
Assume \( P \equiv T \) (the case \( P \equiv F \) is symmetric). So in this case
\[
t =_c \rho((a \circ T) \circ b \triangleright (P' \circ c \triangleright Q')),
\]
and we distinguish two cases:

i. \( P' \equiv T \) or \( Q' \equiv T \). Now the reply to \( a \) in \( a \circ T \) following a positive reply to the initial \( b \) has no effect, so this \( a \) must be followed by another central \( a \). But this last \( a \) can also be reached after a \( b \) and a \( c \), which contradicts \( \phi_{a,b,c} \).

ii. \( P' \equiv Q' \equiv F \). Since property \( \phi_{a,b,c} \) holds it must be the case that \( a \) is a central condition in \( r(T) \) with the property that \( T_a(r(T)) \neq _c F_a(r(T)) \), otherwise the initial \( b \) that stems from the substitution \( x \mapsto (a \circ T) \circ b \triangleright (c \circ F) \) in \( r(x) \) is upon reply \( T \) immediately followed by \( a \circ T \) and each occurrence of this \( a \) is not able to yield both \( T \) and \( F \), contradicting \( \phi_{a,b,c} \). (And also because this substitution yields no further occurrences of \( b \) upon reply \( T \).)

Similarly, \( c \) is a central condition in \( r(F) \) with the property that \( T_c(r(F)) \neq _c F_c(r(F)) \). We find that \( r(b) \) also satisfies \( \phi_{a,b,c} \). Now observe that \( r(b) \) is a term with lower \( \#_2\rho \)-value than \( r(f(b, t')) \), which is a contradiction.
We are left with four cases: either \( a \) is central in \( T_b(g(b, t')) \) and \( F_b(h(b, t')) \in cr \{ T, F \} \), or \( c \) is central in \( F_b(h(b, t')) \) and \( T_b(g(b, t')) \in cr \{ T, F \} \). These cases are symmetric and it suffices to consider only the first one, the others can be dealt with similarly.

So assume \( a \) is central in \( T_b(g(b, t')) \) and \( F_b(h(b, t')) = cr \ T \). This implies \( T_b(g(b, t')) = cr \ P \prec a \triangleright Q \) for some \( P, Q \), and after contraction of all \( a \)'s in \( P \) and \( Q \),

\[
T_b(g(b, t')) = cr \ P' \prec a \triangleright Q' \quad \text{for some } P', Q' \in \{ T, F \}.
\]

We find

\[
t = cr \ r((P' \prec a \triangleright Q') \prec b \triangleright T),
\]

and we distinguish two cases:

i. \( P' \equiv T \) or \( Q' \equiv T \). Now \( c \) can be reached after a negative reply to \( b \) according to \( \phi_{a,b,c} \), but this \( c \) can also be reached after a positive reply to \( b \) and the appropriate reply to \( a \), which contradicts \( \phi_{a,b,c} \).

ii. \( P' \equiv Q' \equiv F \). Since property \( \phi_{a,b,c} \) holds it must be the case that \( a \) is a central condition in \( r(F) \) with the property that \( T_a(r(F)) \neq cr \ F_a(r(F)) \), otherwise the initial \( b \) that stems from the substitution \( x \mapsto (a \circ F) \prec b \triangleright T \) in \( r(x) \) is upon reply \( T \) immediately followed by \( a \circ F \) and each occurrence of this \( a \) is not able to yield both \( T \) and \( F \), contradicting \( \phi_{a,b,c} \). (And also because this substitution yields no further occurrences of \( b \) upon reply \( T \).)

Also, \( c \) is a central condition in \( r(T) \) with the property that \( T_c(r(T)) \neq cr \ F_c(r(T)) \). We find that \( r(b) \) also satisfies \( \phi_{a,b,c} \). Now observe that \( r(b) \) is a term with lower \( \#_{2p} \)-value than \( r(f(b, t')) \), which is a contradiction.

This concludes our proof. \( \square \)

ACKNOWLEDGMENTS

We thank Chris Regenboog for discussions about completeness results. We thank three referees for their careful reviews of a prior version, which we think improved the presentation considerably.

REFERENCES


Received February 2009; revised April 2010; accepted July 2010