Proposition algebra and short-circuit logic

Jan A. Bergstra and Alban Ponse

Section Theory of Computer Science
Informatics Institute, Faculty of Science
University of Amsterdam, The Netherlands
www.science.uva.nl/{~janb/,~alban/}

Abstract. Short-circuit evaluation denotes the semantics of propositional connectives in which the second argument is only evaluated if the first argument does not suffice to determine the value of the expression. In programming, short-circuit evaluation is widely used.
We review proposition algebra [2010], an algebraic approach to propositional logic with side effects that models short-circuit evaluation. Proposition algebra is based on Hoare's conditional [1985], which is a ternary connective comparable to if-then-else. Starting from McCarthy's notion of sequential evaluation [1963] we discuss a number of valuation congruences on propositional statements and we introduce Hoare-McCarthy algebras as the structures that model these congruences. We also briefly discuss the associated short-circuit logics, i.e., the logics that define these congruences if one restricts to sequential binary connectives.
Key words: Conditional composition, reactive valuation, sequential connective, short-circuit evaluation, side effect.

1 Introduction

Short-circuit evaluation is a folk term[1] that describes how the common propositional connectives are evaluated in a setting of programming languages: evaluation stops as soon as the value T (*true*) or F (*false*) of the expression is determined. In particular, the "conjunction" of `x` and `y` in a notation commonly used to prescribe short-circuit evaluation, is often explained by the identity

$$\texttt{x \&\& y} = \texttt{if x then y else } F,$$

and the connective *or* in short-circuit interpretation, notation `||` , is then explained by the identity

$$\texttt{x || y} = \texttt{if x then } T \texttt{ else y}.$$

So, evaluation of `x && y` stops if `x` yields F and then `y` is not evaluated, and similarly, evaluation of `x || y` stops if `x` yields T and then `y` is not evaluated. In the most general case, both `&&` and `||` are not commutative.

[1] Other names used for short-circuit evaluation are *Minimal evaluation* and *McCarthy evaluation*.

Following this lay-out, the evaluation of a "conditional expression" is considered a natural candidate for short-circuit evaluation, and hence justifies our choice for Hoare's ternary connective

$$x \triangleleft y \triangleright z,$$

i.e., the *conditional* connective that represents `if` y `then` x `else` z, as a basic connective. Hoare's conditional connective is introduced in 1985 in the paper [9] (accounts of a similar ternary connective can be found in [7, 8]). So,

$$x \;\texttt{\&\&}\; y = y \triangleleft x \triangleright F \quad \text{and} \quad x \;\texttt{||}\; y = T \triangleleft x \triangleright y. \tag{1}$$

The conditional connective satisfies the three equational laws

$$x \triangleleft T \triangleright y = x, \quad x \triangleleft F \triangleright y = y \quad \text{and} \quad T \triangleleft x \triangleright F = x. \tag{2}$$

Interestingly, in the most general case the conditional connective cannot be defined in terms of the common binary connectives, where by "most general" we refer to a semantics in which all possible "side effects" can occur, and thus a semantics that identifies least. As an example, in many imperative-based programming languages, assignments such as `x=x+1` when interpreted as atoms (i.e., atomic propositions) yield upon evaluation the interpretation of the assigned value next to having the intended side effect. It is trivial to find a propositional statement P such that $P \neq P \triangleleft (\texttt{x=x+1}) \triangleright F$, or equivalently, $P \neq (\texttt{x=x+1}) \;\texttt{\&\&}\; P$, e.g.,

$$(\texttt{x==2}) \neq (\texttt{x==2}) \triangleleft (\texttt{x=x+1}) \triangleright F$$

if the initial value of `x` is either `1` or `2`, `==` is interpreted as an equality test, and the interpretation of values different from zero is T. However, the three laws for the conditional (2) are valid in this most general case.

In case side effects do *not* occur, the conditional can be defined: [2] using the common notation for connectives, a definition is

$$y \triangleleft x \triangleright z = (x \wedge y) \vee (\neg x \wedge z),$$

which is easily seen by substituting T respectively F for x. An example in a setting *with* side effects that refutes this translation is

$$(\texttt{y==2}) \triangleleft ((\texttt{x=x+1}) \wedge (\texttt{x==2})) \triangleright (\texttt{y=y+1}).$$

This follows easily: if both `x` and `y` have initial value `1`, the interpretation of this conditional expression yields F with the side effect that `x` has final value `2`, while the above-mentioned translation yields T with the side effect that the final value of `x` is `3` and the final value of `y` is `2` (note that this argument holds irrespective of the question whether $\wedge$ is interpreted as a short-circuit operator).

[2] This is the semantical setting in Hoare's paper [9], where the conditional was introduced to provide an equational basis for propositional logic.

A way to settle whether side effects have impact, and if so, to what extent, is to distinguish various types of *valuation semantics.* Typically, and illustrated by the above examples, a valuation may return different values for the same atom during the sequential evaluation of a propositional statement (a closed term), and valuation semantics is about such reactive valuations. We adopt the conditional connective as a primitive connective and both T and F as constants. A *proposition algebra* is a model of the three axioms mentioned in (2) and an axiom for decomposing a compound central condition c in $x \triangleleft c \triangleright z$. By adding more axioms, more propositional statements are identified, and all proposition algebras we consider are defined by concise equational axiomatizations. Given some proposition algebra, a valuation semantics can be defined that is constructed from so-called valuation functions, that is, functions defined on sequences of atoms that return either T or F. Propositional statements are identified if they yield in each context for each valuation function the same result. This context requirement refers to the fact that upon the evaluation of an atom that yields a side effect, the valuation value of future atoms in the propositional statement under evaluation is possibly flipped, as is clear from the previous examples.

Concerning conjunction and disjunction, we will consider *sequential* versions of these connectives that by their notation prescribe short-circuit evaluation and that are defined with the conditional (cf. the equations in (1)). Also, negation can be easily defined in terms of the conditional:

$$\neg x = F \triangleleft x \triangleright T.$$

Given some axiomatization of a proposition algebra, a *short-circuit logic* is a logic that implies all consequences that can be expressed using only binary sequential conjunction, negation and the constant T. Typical examples are the associativity of sequential conjunction and the double negation shift $\neg\neg x = x$.

In this paper we present a survey of our work based on proposition algebra [4]. In the next section we briefly discuss so-called *Hoare-McCarthy algebras* (HMAs). HMAs were introduced in [6] in order to provide a more elegant and generic framework for the valuation semantics associated with proposition algebra (we return to this point in Section 7). We construct an HMA that identifies least and characterizes *structural congruence.* In Section 3 we consider the short-circuit logic that is associated with structural congruence (short-circuit logics were introduced in [5]). Section 4 is about *contractive congruence*, a congruence that identifies more propositional statements than structural congruence. We construct a characterizing HMA and we briefly consider the short-circuit logic associated with contractive congruence. In Section 5 we discuss *memorizing congruence*, a congruence that identifies more than contractive congruence and less than propositional logic, and we argue that the associated short-circuit logic also defines this congruence because the conditional is definable in this setting (whereas it is not in contractive congruence, see Section 7). In Section 6 we consider *static congruence* and its short-circuit logic; apart from the notation, this is the setting of conventional propositional logic and no side effects are possible. In Section 7 we end the paper with a brief summary and discussion about our work described in [4–6].

$$x \lhd T \rhd y = x \quad \text{(CP1)}$$
$$x \lhd F \rhd y = y \quad \text{(CP2)}$$
$$T \lhd x \rhd F = x \quad \text{(CP3)}$$
$$x \lhd (y \lhd z \rhd u) \rhd v = (x \lhd y \rhd v) \lhd z \rhd (x \lhd u \rhd v) \quad \text{(CP4)}$$

Table 1. The set CP of axioms for proposition algebra

2 Proposition algebras and HMAs

In this section we define proposition algebras, and in order to capture their valuation semantics we briefly discuss *Hoare-McCarthy algebras*, a certain type of two-sorted algebras that we introduced in [6].

Throughout this paper let A be a non-empty, denumerable set of atoms (atomic propositions) with typical elements $a, b, \ldots$. Define C as the sort of conditional expressions with signature

$$\Sigma_{ce}^A = \{a : C,\ T : C,\ F : C,\ \_ \lhd \_ \rhd \_ : C \times C \times C \to C \mid a \in A\},$$

thus each atom in A is a constant of sort C. In Σ_{ce}^A, *ce* stands for "conditional expressions". We write $\mathcal{T}_{\Sigma_{ce}^A}$ for the set of closed terms over Σ_{ce}^A. Given an expression $t_1 \lhd t_2 \rhd t_3$ we will sometimes refer to t_2 as the *central condition*. We assume that conditional composition satisfies the axioms in Table 1 and we refer to this set of axioms with CP (Conditional Propositions). Axiom (CP4) also stems from [9] and defines decomposition of the central condition by distributivity. We argue in Section 3 that CP characterizes all valid identities in the case that unrestricted side effects occur.

Definition 1. *A Σ_{ce}^A-algebra is a* ***proposition algebra*** *if it is a model of* CP*.*

A non-trivial initial algebra $I(\Sigma_{ce}^A, \text{CP})$ exists. This can be easily shown in the setting of term rewriting [12]. It is not hard to show that directing all CP-axioms from left to right yields a strongly normalizing TRS (term rewriting system) for closed terms. However, the normal forms resulting from this TRS are not particularly suitable for systematic reasoning, and we introduce another class of closed terms for this purpose.

Definition 2. *A term $t \in \mathcal{T}_{\Sigma_{ce}^A}$ is a* ***basic form*** *if for $a \in A$,*

$$t ::= T \mid F \mid t \lhd a \rhd t.$$

Lemma 1. *For each closed term $t \in \mathcal{T}_{\Sigma_{ce}^A}$ there exists a unique basic form t' with* $\text{CP} \vdash t = t'$*.*

Proof. Let t'' be the unique normal form of t. Replace in t'' each subterm that is a single atom a and occurs as an outer argument by $T \lhd a \rhd F$. This results in a unique basic form t' and clearly $\text{CP} \vdash t = t'$. □

Let S be a sort of states with constant c. We extend the signature Σ^A_{ce} to

$$\Sigma^A_{sce} = \Sigma^A_{ce} \cup \{c : S,\ . \triangleleft . \triangleright . : S \times C \times S \to S\},$$

where *sce* stands for "states and conditional expressions".

Definition 3. *A Σ^A_{sce}-algebra is a* ***two-sorted proposition algebra*** *if its Σ^A_{ce}-reduct is a proposition algebra, and if it satisfies the following axioms where x ranges over conditional expressions and s, s' range over states:*

$$s \triangleleft T \triangleright s' = s, \tag{TS1}$$
$$s \triangleleft F \triangleright s' = s', \tag{TS2}$$
$$x \neq T \wedge x \neq F \to s \triangleleft x \triangleright s' = c. \tag{TS3}$$

So, the state set of a two-sorted proposition algebra can be seen as one that is equipped with an if-then-else construct and conditions that stem from CP. We extend the signature Σ^A_{sce} to

$$\Sigma^A_{spa} = \Sigma^A_{sce} \cup \{\, ! : C \times S \to S,\ \bullet : C \times S \to C\},$$

where *spa* stands for "stateful proposition algebra" (see below). The operator ! is called "reply" and the operator $\bullet$ is called "apply" and we further assume that these operators bind stronger than conditional composition. The reply and apply operator are taken from [3].

Definition 4. *A Σ^A_{spa}-algebra is a* ***stateful proposition algebra****, SPA for short, if its reduct to Σ^A_{sce} is a two-sorted proposition algebra, and if it satisfies the following axioms where x, y, z range over conditional expressions and s ranges over states:*

$$T \,!\, s = T, \tag{SPA1}$$
$$F \,!\, s = F, \tag{SPA2}$$
$$(x \triangleleft y \triangleright z) \,!\, s = x \,!\, (y \bullet s) \triangleleft y \,!\, s \triangleright z \,!\, (y \bullet s), \tag{SPA3}$$
$$T \bullet s = s, \tag{SPA4}$$
$$F \bullet s = s, \tag{SPA5}$$
$$(x \triangleleft y \triangleright z) \bullet s = x \bullet (y \bullet s) \triangleleft y \,!\, s \triangleright z \bullet (y \bullet s), \tag{SPA6}$$
$$x \,!\, s = T \vee x \,!\, s = F, \tag{SPA7}$$
$$\forall s (x \,!\, s = y \,!\, s \wedge x \bullet s = y \bullet s) \to x = y. \tag{SPA8}$$

We refer to (SPA7) *as* ***two-valuedness*** *and we write* CTS *(abbreviating* C*P and* T*S and* S*PA) for the set that exactly contains all fifteen axioms involved.*

In a stateful proposition algebra $\mathbb{S}$ with domain C' of conditional expressions and domain S' of states, a propositional statement $t \in \mathcal{T}_{\Sigma^A_{ce}}$ can be associated with a 'valuation function' $t\,!\ : S' \to \{T, F\}$ (the evaluation of t according to some initial valuation function or 'state') and a 'state transformer' $t \bullet\, : S' \to S'$.

Definition 5. *A **Hoare-McCarthy algebra**, HMA for short, is the Σ_{ce}^A-reduct of a stateful proposition algebra.*

For each HMA $\mathbb{A}$ we have by definition $\mathbb{A} \models \text{CP}$. In Theorem 1 below we prove the existence of an HMA that characterizes CP in the sense that a closed equation is valid only if it is derivable from CP.

We define *structural congruence*, notation $=_{sc}$, on $\mathcal{T}_{\Sigma_{ce}^A}$ as the congruence generated by CP.

Theorem 1. *An HMA that characterizes* CP *exists: there is an HMA $\mathbb{A}^{sc}$ such that for all $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$,* $\quad \text{CP} \vdash t = t' \iff \mathbb{A}^{sc} \models t = t'$.

Proof. We construct the Σ_{spa}^A-algebra $\mathbb{S}^{sc}$ with $C' = \mathcal{T}_{\Sigma_{ce}^A}/_{=_{sc}}$ as its set of conditional expressions and, writing A^+ for the set of finite, non-empty strings over the set A of atoms, the function space

$$S' = \{T, F\}^{A^+}$$

as its set of states. For each state f and atom $a \in A$ define $a\,!\,f = f(a)$ and $a \bullet f$ as the function defined for $\sigma \in A^+$ by

$$(a \bullet f)(\sigma) = f(a\sigma).$$

The state constant c is given an arbitrary interpretation, and the axioms (TS1)–(TS3) define $\_ \lhd \_ \rhd \_ : S' \times C' \times S'$ in $\mathbb{S}^{sc}$. The axioms (SPA1)–(SPA6) fully determine the functions $!$ and $\bullet$, and this is well-defined: if $t =_{sc} t'$ then for all f, $t\,!\,f = t'\,!\,f$ and $t \bullet f = t' \bullet f$ (this follows by inspection of the CP axioms). The axiom (SPA7) holds by construction of S'. In order to prove that $\mathbb{S}^{sc}$ is a SPA it remains to be shown that axiom (SPA8) holds, i.e., for all $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$,

$$\forall f (t\,!\,f = t'\,!\,f \wedge t \bullet f = t' \bullet f) \rightarrow t =_{sc} t'.$$

This follows by contraposition. By Lemma 1 we may assume that t and t' are basic forms, and we apply induction on the complexity of t, where we use $\equiv$ to denote syntactic equivalence:

1. Suppose $t \equiv T$, then $t' \equiv F$ yields $t\,!\,f \neq t'\,!\,f$ for any f, and if $t' \equiv t_1 \lhd a \rhd t_2$ then consider f with $f(a) = T$ and $f(a\sigma) = F$ for $\sigma \in A^+$. We find $t \bullet f = f$ and $t' \bullet f \neq f$ because $(t' \bullet f)(a) = (t_1 \bullet f)(a\sigma) = F$.
2. If $t \equiv F$ a similar argument applies.
3. Suppose $t \equiv t_1 \lhd a \rhd t_2$, then the cases $t' \in \{T, F\}$ can be dealt with as above. If $t' \equiv t_3 \lhd a \rhd t_4$ then assume $t_1 \lhd a \rhd t_2 \neq_{sc} t_3 \lhd a \rhd t_4$ because $t_1 \neq_{sc} t_3$. By induction there exists f with $t_1 \bullet f \neq t_3 \bullet f$ or $t_1\,!\,f \neq t_3\,!\,f$. Take some g such that $a \bullet g = f$ and $a\,!\,g = T$, then g distinguishes $t_1 \lhd a \rhd t_2$ and $t_3 \lhd a \rhd t_4$. If $t_1 =_{sc} t_3$, then a similar argument applies for $t_2 \neq_{sc} t_4$.
   If $t' \equiv t_3 \lhd b \rhd t_4$ with a and b different, then $(t_1 \lhd a \rhd t_2) \bullet f \neq (t_3 \lhd b \rhd t_4) \bullet f$ for f defined by $f(a) = f(a\sigma) = T$ and $f(b) = f(b\sigma) = F$ because $((t_1 \lhd a \rhd t_2) \bullet f)(a) = (t_1 \bullet (a \bullet f))(a) = f(a\rho a) = T$, and $((t_3 \lhd b \rhd t_4) \bullet f)(a) = (t_4 \bullet (b \bullet f))(a) = f(b\rho' a) = F$ (where ρ, ρ' possibly equal the empty string).

So $\mathbb{S}^{sc}$ is a SPA. Define the HMA $\mathbb{A}^{sc}$ as the Σ^A_{ce}-reduct of $\mathbb{S}^{sc}$. The validity of axiom (SPA8) proves $\Longleftarrow$ as stated in the theorem (the implication $\Longrightarrow$ holds by definition of a SPA). $\square$

Observe that $\mathbb{A}^{sc} \cong I(\Sigma^A_{ce}, \mathrm{CP})$ and that by the proof of the above theorem we find for all $t, t' \in \mathcal{T}_{\Sigma^A_{ce}}$,

$$\mathrm{CP} \vdash t = t' \iff \mathbb{S}^{sc} \models t = t'.$$

In [10] it is shown that the axioms of CP are independent, and also that they are ω-complete if the set of atoms involved contains at least two elements.

3 Free short-circuit logic: FSCL

In this section we recall our generic definition of a short-circuit logic introduced in [5] and discuss *free short-circuit logic* (FSCL), the least identifying short-circuit logic we consider and that is associated with CP.

We first return to our discussion of short-circuit evaluation started in the Introduction. Our interest can be captured by the following question: Given some programming language, what is the logic that implies the equivalence of conditions, notably in if-then-else and while-do constructs and the like? In [5] we study sequential variants of propositional logic that are based on *left-sequential conjunction*, i.e., conjunction that prescribes short-circuit evaluation and that is defined by

$$x \circ\!\!\wedge\, y = y \triangleleft x \triangleright F$$

where the fresh symbol $\circ\!\!\wedge$ is taken from [1] (the small circle indicates that the left argument must be evaluated first). It is not hard to find examples that show that the laws $x \circ\!\!\wedge\, x$ and its weaker version $a \circ\!\!\wedge\, a = a$ are not valid in the most general case (cf. the examples discussed in the Introduction), which is the case characterized by CP. We define a set of equations that is sound in FSCL and raise the question of its completeness.

We define short-circuit logics such as FSCL in a generic way. Intuitively, a short-circuit logic is a logic that implies all consequences of CP that can be expressed in the signature $\{T, \neg, \circ\!\!\wedge\}$. The definition below uses the export-operator $\square$ of module algebra [2] to define this in a precise manner, where it is assumed that CP satisfies the format of a module specification. In module algebra, $\Sigma \,\square\, X$ is the operation that exports the signature Σ from module X while declaring other signature elements hidden. In this case it declares conditional composition to be an auxiliary operator.

Definition 6. *A* ***short-circuit logic*** *is a logic that implies the consequences of the module expression*

$$\mathrm{SCL} = \{T, \neg, \circ\!\!\wedge\} \,\square\, (\mathrm{CP} + \langle\, \neg x = F \triangleleft x \triangleright T \,\rangle + \langle\, x \circ\!\!\wedge\, y = y \triangleleft x \triangleright F \,\rangle).$$

For example, $\mathrm{SCL} \vdash \neg\neg x = x$ can be easily shown. Following Definition 6, the most basic (least identifying) short-circuit logic we distinguish is this one:

$$F = \neg T \tag{SCL1}$$
$$x \mathbin{{}^{\circ}\!\vee} y = \neg(\neg x \mathbin{{}_{\circ}\!\wedge} \neg y) \tag{SCL2}$$
$$\neg\neg x = x \tag{SCL3}$$
$$T \mathbin{{}_{\circ}\!\wedge} x = x \tag{SCL4}$$
$$x \mathbin{{}_{\circ}\!\wedge} T = x \tag{SCL5}$$
$$F \mathbin{{}_{\circ}\!\wedge} x = F \tag{SCL6}$$
$$(x \mathbin{{}_{\circ}\!\wedge} y) \mathbin{{}_{\circ}\!\wedge} z = x \mathbin{{}_{\circ}\!\wedge} (y \mathbin{{}_{\circ}\!\wedge} z) \tag{SCL7}$$
$$(x \mathbin{{}^{\circ}\!\vee} y) \mathbin{{}_{\circ}\!\wedge} (z \mathbin{{}_{\circ}\!\wedge} F) = (\neg x \mathbin{{}^{\circ}\!\vee} (z \mathbin{{}_{\circ}\!\wedge} F)) \mathbin{{}_{\circ}\!\wedge} (y \mathbin{{}_{\circ}\!\wedge} (z \mathbin{{}_{\circ}\!\wedge} F)) \tag{SCL8}$$
$$(x \mathbin{{}^{\circ}\!\vee} y) \mathbin{{}_{\circ}\!\wedge} (z \mathbin{{}^{\circ}\!\vee} T) = (x \mathbin{{}_{\circ}\!\wedge} (z \mathbin{{}^{\circ}\!\vee} T)) \mathbin{{}^{\circ}\!\vee} (y \mathbin{{}_{\circ}\!\wedge} (z \mathbin{{}^{\circ}\!\vee} T)) \tag{SCL9}$$
$$((x \mathbin{{}_{\circ}\!\wedge} F) \mathbin{{}^{\circ}\!\vee} y) \mathbin{{}_{\circ}\!\wedge} z = (x \mathbin{{}_{\circ}\!\wedge} F) \mathbin{{}^{\circ}\!\vee} (y \mathbin{{}_{\circ}\!\wedge} z) \tag{SCL10}$$

Table 2. EqFSCL, a set of equations for FSCL

Definition 7. FSCL *(free short-circuit logic)* *is the short-circuit logic that implies no other consequences than those of the module expression* SCL*.*

Although the constant F does not occur in the exported signature of SCL, we discuss FSCL using this constant to enhance readability. This is not problematic because

$$\mathrm{CP} + \langle\, \neg x = F \triangleleft x \triangleright T \,\rangle \vdash F = \neg T,$$

so F can be used as a shorthand for $\neg T$ in FSCL.

In Table 2 we provide equations for FSCL and we use the name EqFSCL for this set of equations. Some comments: equation (SCL1) defines the constant F, and equation (SCL2) defines $\mathbin{{}^{\circ}\!\vee}$, so-called *left-sequential disjunction*. Equations (SCL3) – (SCL7) need no comment. Equation (SCL8) defines a property of the mix of negation and the sequential connectives, and its soundness can perhaps be easily grasped by considering the evaluation values of x (observe that $z \mathbin{{}_{\circ}\!\wedge} F = (z \mathbin{{}_{\circ}\!\wedge} F) \mathbin{{}_{\circ}\!\wedge} ...$). Equation (SCL9) defines a restricted form of right-distributivity of $\mathbin{{}_{\circ}\!\wedge}$, and so does equation (SCL10) (because $(x \mathbin{{}_{\circ}\!\wedge} F) \mathbin{{}_{\circ}\!\wedge} z = x \mathbin{{}_{\circ}\!\wedge} F$).

We note that equations (SCL2) and (SCL3) imply sequential versions of De Morgan's laws, which allows us to use sequential versions of the duality principle. Furthermore, we note that the equation $x \mathbin{{}_{\circ}\!\wedge} F = F$ should not be a consequence of EqFSCL: it is easily seen that $\mathbb{A}^{sc} \not\models F \triangleleft a \triangleright F = F$ (see Theorem 1). A simple consequence of equation (SCL8) is

$$x \mathbin{{}_{\circ}\!\wedge} F = \neg x \mathbin{{}_{\circ}\!\wedge} F \tag{SCL8*}$$

(take $y = z = F$), which we will use in Section 5, and another interesting EqFSCL-consequence is $(x \mathbin{{}^{\circ}\!\vee} T) \mathbin{{}_{\circ}\!\wedge} y = (x \mathbin{{}_{\circ}\!\wedge} F) \mathbin{{}^{\circ}\!\vee} y$ (for a proof see [5]).

Proposition 1 (Soundness). *The equations in* EqFSCL *(see Table 2) are derivable in* FSCL*.*

While not having found any equations that are derivable in FSCL but not from EqFSCL, we failed to prove completeness of EqFSCL in the following sense (of course, $\Longrightarrow$ follows from Proposition 1):

$$\text{For all SCL-terms } t \text{ and } t', \quad \text{EqFSCL} \vdash t = t' \iff \text{FSCL} \vdash t = t'. \qquad (3)$$

4 Contractive congruence

In this section we consider the congruence defined by the axioms of CP and these axiom schemes ($a \in A$):

$$(x \triangleleft a \triangleright y) \triangleleft a \triangleright z = x \triangleleft a \triangleright z, \qquad \text{(CPcr1)}$$

$$x \triangleleft a \triangleright (y \triangleleft a \triangleright z) = x \triangleleft a \triangleright z. \qquad \text{(CPcr2)}$$

Following [4], we write CP_{cr} for this set of axioms. Typically, successive equal atoms are contracted according to the axiom schemes (CPcr1) and (CPcr2).

Let *contractive congruence*, notation $=_{cr}$, be the congruence on $\mathcal{T}_{\Sigma_{ce}^A}$ generated by the axioms of CP_{cr}.

Definition 8. *A term $t \in \mathcal{T}_{\Sigma_{ce}^A}$ is a* ***cr-basic form*** *if for $a \in A$,*

$$t ::= T \mid F \mid t_1 \triangleleft a \triangleright t_2$$

and t_i ($i = 1, 2$) is a cr-basic form with the restriction that the central condition (if present) is different from a.

Lemma 2. *For each $t \in \mathcal{T}_{\Sigma_{ce}^A}$ there exists a cr-basic form t' with $\text{CP}_{cr} \vdash t = t'$.*

Proof. By structural induction; see [4] for a full proof. □

Theorem 2. *For $|A| > 1$, an HMA that characterizes CP_{cr} exists, i.e. there is an HMA $\mathbb{A}^{cr}$ such that for all $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$, $\quad \text{CP}_{cr} \vdash t = t' \iff \mathbb{A}^{cr} \models t = t'$.*

Proof. Let $A^{cr} \subset A^+$ be the set of strings that contain no consecutive occurrences of the same atom. Construct the Σ_{spa}^A-algebra $\mathbb{S}^{cr}$ with $\mathcal{T}_{\Sigma_{ce}^A}/_{=_{cr}}$ as its set of conditional expressions and the function space

$$S' = \{T, F\}^{A^{cr}}$$

as its set of states. For each state f and atom $a \in A$ define $a\,!\,f = f(a)$ and $a \bullet f$ by

$$(a \bullet f)(\sigma) = \begin{cases} f(\sigma) & \text{if } \sigma = a \text{ or } \sigma = a\rho, \\ f(a\sigma) & \text{otherwise.} \end{cases}$$

Clearly, $a \bullet f \in \{T, F\}^{A^{cr}}$ if $f \in \{T, F\}^{A^{cr}}$. Similar as in the proof of Theorem 1, the state constant c is given an arbitrary interpretation, and the axioms (TS1)–(TS3) define the function $s \triangleleft f \triangleright s'$ in $\mathbb{S}^{cr}$. The axioms (SPA1)–(SPA6) fully determine the functions $!$ and $\bullet$, and this is well-defined: if $t =_{cr} t'$ then for all

f, $t\,!\,f = t'\,!\,f$ and $t \bullet f = t' \bullet f$ follow by inspection of the CP_{cr} axioms. We show soundness of the axiom scheme (CPcr1): note that $a\,!\,(a \bullet f) = a\,!\,f$ and $a \bullet (a \bullet f) = a \bullet f$, and derive

$$\begin{aligned}((t_1 \lhd a \rhd t_2) \lhd a \rhd t)\,!\,f &= (t_1 \lhd a \rhd t_2)\,!\,(a \bullet f) \lhd a\,!\,f \rhd t\,!\,(a \bullet f)\\ &= t_1\,!\,(a \bullet (a \bullet f)) \lhd a\,!\,f \rhd t\,!\,(a \bullet f)\\ &= (t_1 \lhd a \rhd t)\,!\,f,\end{aligned}$$

and

$$\begin{aligned}((t_1 \lhd a \rhd t_2) \lhd a \rhd t) \bullet f &= (t_1 \lhd a \rhd t_2) \bullet (a \bullet f) \lhd a\,!\,f \rhd t \bullet (a \bullet f)\\ &= t_1 \bullet (a \bullet (a \bullet f)) \lhd a\,!\,f \rhd t \bullet (a \bullet f)\\ &= (t_1 \lhd a \rhd t) \bullet f.\end{aligned}$$

The soundness of (CPcr2) follows in a similar way. The axiom (SPA7) holds by construction of S'. In order to prove that $\mathbb{S}^{cr}$ is a SPA it remains to be shown that axiom (SPA8) holds. This follows by contraposition: by Lemma 2 we may assume that both t and t' are cr-basic forms, and apply induction on the complexity of t (for a detailed proof of this, see [6]). Now define the HMA $\mathbb{A}^{cr}$ as the Σ^A_{ce}-reduct of $\mathbb{S}^{cr}$. The above argument on the soundness of the axiom schemes (CPcr1) and (CPcr2) proves $\Longrightarrow$ as stated in the theorem, and the validity of axiom (SPA8) proves $\Longleftarrow$. Finally, note that $\mathbb{A}^{cr} \cong I(\Sigma^A_{ce}, \mathrm{CP}_{cr})$. □

In the proof above we defined the SPA $\mathbb{S}^{cr}$ and we found that if $|A| > 1$, then for all $t, t' \in \mathcal{T}_{\Sigma^A_{ce}}$,

$$\mathrm{CP}_{cr} \vdash t = t' \iff \mathbb{S}^{cr} \models t = t'. \tag{4}$$

If $A = \{a\}$ then $A^{cr} = A$ and $\mathbb{S}^{cr}$ as defined above has only two states, say f and g with $f(a) = T$ and $g(a) = F$. It easily follows that

$$\mathbb{A}^{cr} \models T \lhd a \rhd T = T,$$

so $\mathbb{A}^{cr} \not\cong I(\Sigma^A_{ce}, \mathrm{CP}_{cr})$ if $A = \{a\}$. The following corollary is related to Theorem 2 and characterizes contractive congruence in terms of a quasivariety of SPAs that satisfy an extra condition.

Corollary 1. *Let $|A| > 1$. Let $\mathcal{C}_{cr}$ be the class of SPAs that satisfy for all $a \in A$ and $s \in S$,*

$$a\,!\,(a \bullet s) = a\,!\,s \;\wedge\; a \bullet (a \bullet s) = a \bullet s.$$

Then for all $t, t' \in \mathcal{T}_{\Sigma^A_{ce}}$, $\quad \mathcal{C}_{cr} \models t = t' \iff \mathrm{CP}_{cr} \vdash t = t'$.

Proof. By its definition, $\mathbb{S}^{cr} \in \mathcal{C}_{cr}$, which by (4) implies $\Longrightarrow$. For the converse, it is sufficient to show that the axioms (CPcr1) and (CPcr2) hold in any SPA that is in $\mathcal{C}_{cr}$. Let such $\mathbb{S}$ be given. Consider (CPcr1): if for an interpretation of s in $\mathbb{S}$, $a\,!\,s = F$ the proof is trivial, and if $a\,!\,s = T$, then $a\,!\,(a \bullet s) = T$ and thus

$$\begin{aligned}((t_1 \lhd a \rhd t_2) \lhd a \rhd t)\,!\,s &= t_1\,!\,(a \bullet (a \bullet s))\\ &= t_1\,!\,(a \bullet s)\\ &= (t_1 \lhd a \rhd t)\,!\,s,\end{aligned}$$

and $((t_1 \lhd a \rhd t_2) \lhd a \rhd t) \bullet s = (t_1 \lhd a \rhd t) \bullet s$ can be proved in a similar way. □

Finally, we briefly discuss a variant of short-circuit logic that is based on CP_{cr}. We write $\mathrm{CP}_{cr}(A)$ to denote CP_{cr} in a notation close to module algebra [2].

Definition 9. CSCL ***(contractive short-circuit logic)*** *is the short-circuit logic that implies no other consequences than those of the module expression*

$$\{T, \neg, \circ\!\!\wedge, a \mid a \in A\} \,\square\, (\mathrm{CP}_{cr}(A) + \langle\, \neg x = F \triangleleft x \triangleright T \,\rangle + \langle\, x \circ\!\!\wedge y = y \triangleleft x \triangleright F \,\rangle).$$

The equations defined by CSCL include those derivable from EqFSCL, and

$$a \circ\!\!\wedge (a \circ\!\!\vee x) = a,$$
$$a \circ\!\!\vee (a \circ\!\!\wedge x) = a.$$

It is an open question whether the extension of EqFSCL with these two equations yields an axiomatization of CSCL. Observe that the following equations are consequences in CSCL:

$$a \circ\!\!\wedge a = a, \qquad a \circ\!\!\vee a = a,$$
$$\neg a \circ\!\!\wedge (\neg a \circ\!\!\vee x) = \neg a, \qquad \neg a \circ\!\!\vee (\neg a \circ\!\!\wedge x) = \neg a,$$
$$\neg a \circ\!\!\wedge \neg a = \neg a, \qquad \neg a \circ\!\!\vee \neg a = \neg a.$$

An example that illustrates the use of CSCL concerns atoms that define manipulation of Boolean registers:

- Consider atoms `set:`i`:`j and `eq:`i`:`j with $i \in \{1, ..., n\}$ (the number of registers) and $j \in \{T, F\}$ (the value of registers).
- An atom `set:`i`:`j can have a side effect (it sets register i to value j) and yields upon evaluation always T.
- An atom `eq:`i`:`j has no side effect but yields upon evaluation only T if register i has value j.

Clearly, the CSCL-consequences mentioned above are valid in the setting of this example, but $x \circ\!\!\wedge x = x$ is not: assume register 1 has value F and let $t =$ `eq:1:`$F \circ\!\!\wedge$ `set:1:`T. Then t yields T upon evaluation in this state, while $t \circ\!\!\wedge t$ yields F.

5 Memorizing congruence

In this section we consider the congruence defined by the axioms of CP and this axiom:

$$x \triangleleft y \triangleright (z \triangleleft u \triangleright (v \triangleleft y \triangleright w)) = x \triangleleft y \triangleright (z \triangleleft u \triangleright w). \qquad \text{(CPmem)}$$

Following [4], we write CP_{mem} for this set of axioms. Axiom (CPmem) defines how the central condition y may recur in a propositional statement, and defines a general form of contraction: with $u = F$ we find

$$x \triangleleft y \triangleright (v \triangleleft y \triangleright w) = x \triangleleft y \triangleright w. \qquad (5)$$

The symmetric variants of (CPmem) and (5) all follow easily with $y \triangleleft x \triangleright z = (z \triangleleft F \triangleright y) \triangleleft x \triangleright (z \triangleleft T \triangleright y) = z \triangleleft (F \triangleleft x \triangleright T) \triangleright y$ (which is a CP-derivation), e.g.,

$$(x \triangleleft y \triangleright (z \triangleleft u \triangleright v)) \triangleleft u \triangleright w = (x \triangleleft y \triangleright z) \triangleleft u \triangleright w. \tag{6}$$

Let *memorizing congruence*, notation $=_{mem}$, be the congruence on $\mathcal{T}_{\Sigma_{ce}^A}$ generated by the axioms of CP_{mem}. As in the preceding cases, a special type of basic forms can be used to construct a SPA $\mathbb{S}^{mem}$ that defines the HMA $\mathbb{A}^{mem}$, which in turn characterizes $=_{mem}$ (for closed terms). Because this construction is quite involved, we here only define the state set of $\mathbb{S}^{mem}$ in order to illustrate the valuation semantics that goes with CP_{mem}, and refer to [6] for all further details and proofs. Let $A^{core} \subset A^+$ be the set of strings in which each element of A occurs at most once. Then the function space

$$M = \{T, F\}^{A^{core}}$$

is the state set of $\mathbb{S}^{mem}$. Define for $f \in M$ the following: $a\,!\,f = f(a)$ and for $\sigma \in A^{core}$,

$$(a \bullet f)(\sigma) = \begin{cases} f(a) & \text{if } \sigma = a \text{ or } \sigma = \rho a, \\ f(a(\sigma{-}a)) & \text{otherwise, where } (\sigma{-}a) \text{ is as } \sigma \text{ but with } a \text{ left out.} \end{cases}$$

For example, $(a \bullet f)(a) = (a \bullet f)(ba) = f(a)$ and $(a \bullet f)(b) = (a \bullet f)(ab) = f(ab)$. In [6] we proved the following result.

Theorem 3. *For $|A| > 1$, an HMA that characterizes CP_{mem} exists, i.e. there is an HMA $\mathbb{A}^{mem}$ such that for all $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$,*

$$\mathrm{CP}_{mem} \vdash t = t' \iff \mathbb{A}^{mem} \models t = t'.$$

Note that if $A = \{a\}$ then M has only two states, say f and g with $f(a) = T$ and $g(a) = F$. It then easily follows that $\mathbb{A}^{mem} \models T \triangleleft a \triangleright T = T$ so in that case $\mathbb{A}^{mem} \not\cong I(\Sigma_{ce}^A, \mathrm{CP}_{mem})$. Furthermore, note that if $A \supseteq \{a, b\}$, it easily follows that $\mathbb{S}^{mem} \not\models a \mathbin{\circ\!\!\wedge} b = b \mathbin{\circ\!\!\wedge} a$: take f such that $f(a) = f(ab) = T$ and $f(b) = F$. The following corollary is related to Theorem 3 and characterizes memorizing congruence in terms of a quasivariety of SPAs that satisfy an extra condition.

Corollary 2. *Let $|A| > 1$. Let $\mathcal{C}_{mem}$ be the class of SPAs that satisfy for all $a \in A$ and $s \in S$,*

$$a\,!\,(x \bullet (a \bullet s)) = a\,!\,s \;\wedge\; a \bullet (x \bullet (a \bullet s)) = x \bullet (a \bullet s).$$

(Note that with $x = T$ this yields the axiom scheme from Corollary 1 that characterizes contractive congruence.) Then for all $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$,

$$\mathcal{C}_{mem} \models t = t' \iff \mathrm{CP}_{mem} \vdash t = t'.$$

Proof. Somewhat involved; see [6]. □

$$F = \neg T \tag{SCL1}$$
$$x \mathbin{{}^{\circ}\!\vee} y = \neg(\neg x \mathbin{{}_{\circ}\!\wedge} \neg y) \tag{SCL2}$$
$$\neg\neg x = x \tag{SCL3}$$
$$T \mathbin{{}_{\circ}\!\wedge} x = x \tag{SCL4}$$
$$x \mathbin{{}_{\circ}\!\wedge} T = x \tag{SCL5}$$
$$F \mathbin{{}_{\circ}\!\wedge} x = F \tag{SCL6}$$
$$(x \mathbin{{}_{\circ}\!\wedge} y) \mathbin{{}_{\circ}\!\wedge} z = x \mathbin{{}_{\circ}\!\wedge} (y \mathbin{{}_{\circ}\!\wedge} z) \tag{SCL7}$$
$$x \mathbin{{}_{\circ}\!\wedge} F = \neg x \mathbin{{}_{\circ}\!\wedge} F \tag{SCL8*}$$
$$x \mathbin{{}_{\circ}\!\wedge} (x \mathbin{{}^{\circ}\!\vee} y) = x \tag{MSCL1}$$
$$x \mathbin{{}_{\circ}\!\wedge} (y \mathbin{{}^{\circ}\!\vee} z) = (x \mathbin{{}_{\circ}\!\wedge} y) \mathbin{{}^{\circ}\!\vee} (x \mathbin{{}_{\circ}\!\wedge} z) \tag{MSCL2}$$
$$(x \mathbin{{}^{\circ}\!\vee} y) \mathbin{{}_{\circ}\!\wedge} (\neg x \mathbin{{}^{\circ}\!\vee} z) = (\neg x \mathbin{{}^{\circ}\!\vee} z) \mathbin{{}_{\circ}\!\wedge} (x \mathbin{{}^{\circ}\!\vee} y) \tag{MSCL3}$$
$$((x \mathbin{{}_{\circ}\!\wedge} y) \mathbin{{}^{\circ}\!\vee} (\neg x \mathbin{{}_{\circ}\!\wedge} z)) \mathbin{{}_{\circ}\!\wedge} u = (x \mathbin{{}^{\circ}\!\vee} (z \mathbin{{}_{\circ}\!\wedge} u)) \mathbin{{}_{\circ}\!\wedge} (\neg x \mathbin{{}^{\circ}\!\vee} (y \mathbin{{}_{\circ}\!\wedge} u)) \tag{MSCL4}$$

Table 3. EqMSCL, a set of axioms for MSCL

We conclude with a brief discussion about the short-circuit logic that is based on CP_{mem}. In this logic, only *static side effects* can occur: during the evaluation of a propositional statement, the value of each atom remains fixed after its first evaluation, which is a typical property axiomatized by CP_{mem}. A major difference with the short-circuit logics discussed in the previous sections is that in CP_{mem} the conditional is definable:

$$\begin{aligned}
(y \mathbin{{}_{\circ}\!\wedge} x) \mathbin{{}^{\circ}\!\vee} (\neg y \mathbin{{}_{\circ}\!\wedge} z) &= T \triangleleft (x \triangleleft y \triangleright F) \triangleright (z \triangleleft (F \triangleleft y \triangleright T) \triangleright F) \\
&= T \triangleleft (x \triangleleft y \triangleright F) \triangleright (F \triangleleft y \triangleright z) \\
&= (T \triangleleft x \triangleright (F \triangleleft y \triangleright z)) \triangleleft y \triangleright (F \triangleleft y \triangleright z) \\
&= (T \triangleleft x \triangleright F) \triangleleft y \triangleright (F \triangleleft y \triangleright z) && \text{by (6)} \\
&= x \triangleleft y \triangleright z. && \text{by (5)}
\end{aligned}$$

Definition 10. MSCL ***(memorizing short-circuit logic)*** *is the short-circuit logic that implies no other consequences than those of the module expression*

$$\{T, \neg, \mathbin{{}_{\circ}\!\wedge}\} \ \square \ (\mathrm{CP}_{mem} + \langle\, \neg x = F \triangleleft x \triangleright T \,\rangle + \langle\, x \mathbin{{}_{\circ}\!\wedge} y = y \triangleleft x \triangleright F \,\rangle).$$

In Table 3 we present a set of axioms for MSCL and we refer to this set by EqMSCL. Axioms (SCL1) – (SCL7) occur in EqFSCL (see Table 2) and thus need no further comment, and neither does axiom (SCL8*). The EqFSCL-equations (SCL8) – (SCL10) are derivable from EqMSCL. For any further comments, intuitions and proofs on MSCL we refer to [5], and we end this section by recalling the main result from that paper:

Theorem 4 (Completeness). *For all* SCL*-terms t and t',*

$$\mathrm{EqMSCL} \vdash t = t' \iff \mathrm{MSCL} \vdash t = t'.$$

An interesting aspect of this result is that we have a complete axiomatization EqMSCL of a logic in which $\circ\!\!\wedge$ is not commutative and in which $x \circ\!\!\wedge F = F$ does not hold, but that is otherwise very close to propositional logic.

6 Static congruence (Propositional logic)

In this section we consider *static congruence* defined by the axioms of CP and the axioms

$$(x \lhd y \rhd z) \lhd u \rhd v = (x \lhd u \rhd v) \lhd y \rhd (z \lhd u \rhd v), \qquad \text{(CPstat)}$$
$$(x \lhd y \rhd z) \lhd y \rhd u = x \lhd y \rhd u. \qquad \text{(CPcontr)}$$

Following [4], we write CP_{stat} for this set of axioms. Note that the symmetric variants of the axioms (CPstat) and (CPcontr), say

$$x \lhd y \rhd (z \lhd u \rhd v) = (x \lhd y \rhd z) \lhd u \rhd (x \lhd y \rhd v), \qquad \text{(CPstat}')$$
$$x \lhd y \rhd (z \lhd y \rhd u) = x \lhd y \rhd u, \qquad \text{(CPcontr}')$$

easily follow with the (derivable) identity $y \lhd x \rhd z = z \lhd (F \lhd x \rhd T) \rhd y$. Moreover, in CP_{stat} it follows that

$$\begin{aligned} x &= (x \lhd y \rhd z) \lhd F \rhd x \\ &= (x \lhd F \rhd x) \lhd y \rhd (z \lhd F \rhd x) && \text{by (CPstat)} \\ &= x \lhd y \rhd x. \end{aligned}$$

We define *static congruence* $=_{stat}$ on $\mathcal{T}_{\Sigma_{ce}^A}$ as the congruence generated by CP_{stat}. Let $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$. Then under static congruence, t and t' can be rewritten into the following special type of basic form: assume the atoms occurring in t and t' are $a_1, ..., a_n$, and 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 the axioms in CP_{stat} are sufficient to rewrite both t and t' into exactly one such special basic form.

Theorem 5. *There exists an HMA that characterizes static congruence, i.e. there is an HMA $\mathbb{A}^{stat}$ such that for all $t, t' \in \mathcal{T}_{\Sigma_{ce}^A}$,*

$$\mathrm{CP}_{stat} \vdash t = t' \iff \mathbb{A}^{stat} \models t = t'.$$

Proof. Construct the Σ_{spa}^A-algebra $\mathbb{S}^{stat}$ with $\mathcal{T}_{\Sigma_{ce}^A}/_{=_{stat}}$ as the set of conditional expressions and the function space

$$S' = \{T, F\}^A$$

as the set of states. For each state f and atom $a \in A$ define $a\,!\,f = f(a)$ and $a \bullet f = f$. Similar as in the proof of Theorem 1, the state constant c is given an arbitrary interpretation, and the axioms (TS1)–(TS3) define the function $s \lhd f \rhd s'$

in $\mathbb{S}^{stat}$. The axioms (SPA1)–(SPA6) fully determine the functions ! and $\bullet$, and this is well-defined: if $t =_{stat} t'$ then for all f, $t!f = t'!f$ and $t\bullet f = t'\bullet f$ follow by inspection of the CP_{stat} axioms. The axiom (SPA7) holds by construction of S'. In order to prove that $\mathbb{S}^{stat}$ is a SPA it remains to be shown that axiom (SPA8) holds. This follows by contraposition. We may assume that both t and t' are in the basic form described above: if t and t' are different in some leaf then the reply function f leading to this leaf satisfies $t \,!\, f \neq t' \,!\, f$.

Define the HMA $\mathbb{A}^{stat}$ as the Σ^A_{ce}-reduct of $\mathbb{S}^{stat}$. The above argument on the soundness of the axioms (CPstat) and (CPcontr) proves $\Longrightarrow$ as stated in the theorem, and the soundness of axiom (SPA8) proves $\Longleftarrow$. Moreover, $\mathbb{A}^{stat} \cong I(\Sigma^A_{ce}, \mathrm{CP}_{stat})$.

From the proof above it follows that for all $t, t' \in \mathcal{T}_{\Sigma^A_{ce}}$,

$$\mathrm{CP}_{stat} \vdash t = t' \iff \mathbb{S}^{stat} \models t = t'. \tag{7}$$

Corollary 3. *Let $\mathcal{C}_{stat}$ be the class of SPAs that satisfy for all $a \in A$ and $s \in S$,*

$$a \bullet s = s.$$

Then for all $t, t' \in \mathcal{T}_{\Sigma^A_{ce}}$, $\quad \mathcal{C}_{stat} \models t = t' \iff \mathrm{CP}_{stat} \vdash t = t'$.

Proof. By its definition, $\mathbb{S}^{stat} \in \mathcal{C}_{stat}$, which by (7) implies $\Longrightarrow$. For the converse, it is sufficient to show that the axioms (CPstat) and (CPcontr) hold in each SPA in $\mathcal{C}_{stat}$. This follows easily from the $\mathcal{C}_{stat}$-identity $t \bullet s = s$ that holds for all $t \in \mathcal{T}_{\Sigma^A_{ce}}$ (see [6] for a detailed proof). □

Finally, we return to short-circuit logic. It appears to be the case that the axiom

$$x \mathbin{\circ\!\!\wedge} F = F \tag{8}$$

marks the distinction between MSCL and propositional logic (PL): adding this axiom to EqMSCL yields an equational characterization of PL (be it in sequential notation and defined with short-circuit evaluation).

We write SSCL (static short-circuit logic) for the extension of the short-circuit logic MSCL obtained by adding the associated axiom $F \triangleleft x \triangleright F = F$ to CP_{mem}, and we write EqSSCL for the extension of the axiom set EqMSCL with axiom (8). It easily follows that

$$\mathrm{EqSSCL} \vdash x \mathbin{\circ\!\!\wedge} \neg x = F,$$

and hence F and T are definable in SSCL. Also, commutativity of $\mathbin{\circ\!\!\wedge}$ is derivable from EqSSCL (see [5]). By duality it follows that full distributivity holds in EqSSCL, and it is not difficult to see that EqSSCL defines the mentioned variant of PL: this follows for example immediately from [11] in which equational bases for Boolean algebra are provided, and each of these bases can be easily derived from EqSSCL (we return to this point in Section 7).

7 Discussion

In this section we further discuss our papers on proposition algebra and short-circuit logic and briefly mention some issues not considered earlier.

In [4] we introduce 'proposition algebra' as a generic term for algebras that model four basic axioms for Hoare's conditional connective $x \triangleleft y \triangleright z$ (introduced in [9]). We define valuation semantics using valuation algebras (VAs), which are algebras over a signature that contains the Boolean constants and valuations as sorts, and that satisfy axioms comparable to those that define a stateful proposition algebra (a SPA). A valuation variety defines a valuation equivalence by identifying all propositional statements that yield the same evaluation result in all VAs in that variety. For example, T and $T \triangleleft a \triangleright T$ are valuation equivalent in all valuation varieties we consider. The largest congruence contained in a given valuation equivalence is then the 'valuation congruence' to be considered. Main results in [4] are the concise axiomatizations of various valuation congruences (some more than discussed in this paper), and a proof that modulo contractive congruence (or any finer congruence), the conditional, in particular $a \triangleleft b \triangleright c$ with a, b and c atoms, is not definable by sequential binary operators. The axiom set CP characterizes the least identifying valuation congruence we consider, and CP extended with the axiom (CPmem) characterizes memorizing congruence, the most identifying valuation congruence below propositional logic that we distinguish. These valuation congruences are ordered in an incremental way, gradually identifying more propositional statements, and have axiomatizations that all share the axioms of CP.[3] In [4] we also consider some complexity issues: in each VA the satisfiability problem SAT can be defined in a natural way and in all valuation congruences defined thus far, SAT is in NP, and in some cases even in P: in the free CP-algebra SAT is polynomial, while in memorizing congruence the complexity of SAT is increased to NP-complete.

In our report [6] we provide an alternative valuation semantics for proposition algebra in the form of HMAs that appears to be more elegant: HMA-based semantics has the advantage that one can define a valuation congruence without first defining the valuation equivalence it is contained in. Furthermore, we show in [6] that not all proposition algebras are HMAs. In particular, we prove that CP + $\langle T \triangleleft x \triangleright T = T \rangle$ has a non-trivial initial algebra (which by definition is a proposition algebra) that is not an HMA because each HMA satisfies the conditional equation $((T \triangleleft x \triangleright T = T) \wedge (T \triangleleft y \triangleright T = T)) \rightarrow T \triangleleft x \triangleright y = T \triangleleft y \triangleright x$, while CP + $\langle T \triangleleft x \triangleright T = T \rangle \nvdash T \triangleleft a \triangleright b = T \triangleleft b \triangleright a$ for distinct atoms a and b.

In [5] we introduce *short-circuit logic*: we show that the extension of CP_{mem} with $\neg$ and $\mathbin{{}_{\circ}\!\wedge}$ (and with F and $\mathbin{{}^{\circ}\!\vee}$ being definable) characterizes a reasonable logic if one restricts to identities defined over the signature $\{T, \neg, \mathbin{{}_{\circ}\!\wedge}\}$. As recalled in the present paper, we provide an axiomatization of MSCL (memorizing short-circuit logic) and we define FSCL (free short-circuit logic) as the most basic (least identifying) short-circuit logic. Each valuation congruence defines a short-

[3] In [10] it is noted that if the set A of atoms contains one element, all valuation congruences other than structural congruence coincide with static valuation congruence.

circuit logic, and these logics are put forward for modeling conditions as used in programming with short-circuit evaluation and for that reason we named them "short-circuit logics". Typical axioms that are valid in FSCL (and thus in each short-circuit logic) are the associativity of ${}_{\circ}\!\wedge$, the double negation shift and $F \mathbin{{}_{\circ}\!\wedge} x = F$, and we conjecture that FSCL is axiomatized by the equations in Table 2. Furthermore, as noted in Section 5, a typical non-validity is $x \mathbin{{}_{\circ}\!\wedge} F = F$, which does not hold modulo memorizing congruence (or any finer congruence). The extension of CP_{mem} with the axiom $F \triangleleft x \triangleright F = F$ that defines SSCL (static short-circuit logic, comprising $x \mathbin{{}_{\circ}\!\wedge} F = F$), or equivalently, the extension of CP_{mem} with the axiom $T \triangleleft x \triangleright T = T$, yields an axiomatization of static valuation congruence that is perhaps more elegant than our axiomatization CP_{stat}: using the expressibility of conditional composition and the commutativity of ${}_{\circ}\!\wedge$ and ${}^{\circ}\!\vee$ (and hence full distributivity), it is not hard to derive the axiom (CPstat). In [5, Appendix C] we provide another axiomatization of static valuation congruence that is even more elegant than $\mathrm{CP}_{mem} + \langle F \triangleleft x \triangleright F = F \rangle$. This axiomatization consists of the five axioms (CP1), (CP2), (CP4) (see Table 1),

$$T \triangleleft x \triangleright y = T \triangleleft y \triangleright x \quad \text{and} \quad (x \triangleleft y \triangleright z) \triangleleft y \triangleright F = x \triangleleft y \triangleright F,$$

and we also prove that it is independent.

References

1. J.A. Bergstra, I. Bethke, and P.H. Rodenburg. A propositional logic with 4 values: true, false, divergent and meaningless. *Journal of Applied Non-Classical Logics*, 5(2):199–218, 1995.
2. J.A. Bergstra, J. Heering, and P. Klint. Module algebra. *Journal of the ACM*, 37(2):335–372, 1990.
3. J.A. Bergstra and C.A. Middelburg. Instruction sequence processing operators. Available at `http://arxiv.org/:ArXiv:0910.5564v2` [cs.LO], 2009.
4. J.A. Bergstra and A. Ponse. Proposition algebra. *ACM Transactions on Computational Logic* 12(3), Article 21 (36 pp), 2011.
5. J.A. Bergstra and A. Ponse. Short-circuit logic. Available at `http://arxiv.org/abs/1012.3674v3` [cs.LO], 2010/2011.
6. J.A. Bergstra and A. Ponse. On Hoare-McCarthy algebras. Available at `http://arxiv.org/abs/1012.5059v1` [cs.LO], 2010.
7. S.L. Bloom and R. Tindell, R. Varieties of "if-then-else". *SIAM Journal of Computing*, 12(4), 677–707, 1983.
8. A.H. Mekler and E.M. Nelson. Equational bases for if-then-else. *SIAM Journal of Computing*, 16(3), 465–485, 1987.
9. C.A.R. Hoare. A couple of novelties in the propositional calculus. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 31(2):173–178, 1985.
10. B.C. Regenboog. Reactive valuations. MSc. thesis Logic, University of Amsterdam. December 2010. Available at `http://arxiv.org/abs/1101.3132v1` [cs.LO], 2011.
11. F.M. Sioson. Equational bases of Boolean algebras. *Journal of Symbolic Logic*, 29(3):115–124, 1964.
12. Terese. *Term Rewriting Systems.* Cambridge Tracts in Theoretical Computer Science, Vol. 55, Cambridge University Press, 2003.