An independent axiomatisation for free short-circuit logic

Ponse, A.; Staudt, D.J.C.

DOI
10.1080/11663081.2018.1448637
Publication date
2018
Document Version
Final published version
Published in
Journal of Applied Non-Classical Logics

JOURNAL OF APPLIED NON-CLASSICAL LOGICS
2018, VOL. 28, NO. 1, 35–71
https://doi.org/10.1080/11663081.2018.1448637

Taylor & Francis
Taylor & Francis Group

An independent axiomatisation for free short-circuit logic

Alban Ponse and Daan J. C. Staudt

Section Theory of Computer Science, Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands

ABSTRACT

Short-circuit evaluation denotes the semantics of propositional connectives in which the second argument is evaluated only if the first argument does not suffice to determine the value of the expression. Free short-circuit logic is the equational logic in which compound statements are evaluated from left to right, while atomic evaluations are not memorised throughout the evaluation, i.e. evaluations of distinct occurrences of an atom in a compound statement may yield different truth values. We provide a simple semantics for free short-circuit logic (SCL) and an independent axiomatisation. Finally, we discuss evaluation strategies, some other SCLs, and side effects.

ARTICLE HISTORY
Received 19 July 2017
Accepted 31 December 2017

KEYWORDS
Logic in computer science; short-circuit evaluation; non-commutative conjunction; sequential connectives; conditional composition; side effect

1. Introduction

Short-circuit(ed) evaluation denotes the semantics of binary propositional connectives in which the second argument is evaluated only if the first argument does not suffice to determine the value of the expression. In the setting of computer science, connectives that prescribe short-circuit evaluation tend to have specific names or notations, such as Dijkstra's **cand** (conditional and) and **cor** (see Dijkstra, 1976; Gries, 1981), or the short-circuited connectives `&&` and `||` as used in programming languages such as C, Go, Java and Perl. Short-circuit evaluation in C is discussed in e.g. Zimmermann and Dold (2003), or in a context with partial predicates, in McCarthy (1963).

A motivation for short-circuit evaluation arises in the setting in which the evaluation of atomic propositions can be state dependent, and atomic evaluations may change the (evaluation) state. As an example, consider the short-circuit evaluation of this program fragment:

```
(f(x) > 5)  &&  (g(x) < 3)
```

(which can occur in the condition of an if-then-else or while construct), the result of which can be different from the short-circuit evaluation of

```
(g(x) < 3)  &&  (f(x) > 5)
```

CONTACT Alban Ponse A.Ponse@uva.nl

if a *side effect* in the evaluation of the atomic propositions `(f(x) > 5)` or `(g(x) < 3)` changes the value of `x`. Note that in this example, the expressions `(f(x) > 5)` and `(g(x) < 3)` are considered to be propositional variables, or, as we will henceforth call these, 'atoms'. In Section 4, we briefly discuss side effects and an example that shows that short-circuit conjunction is not a commutative operation.

Following Bergstra, Bethke, and Rodenburg (1995), we write $P \mathbin{\circ\!\!\wedge} Q$ for the sequential conjunction of P and Q that prescribes short-circuit evaluation (the small circle indicates that the left argument must be evaluated first). Similarly, we write $P \mathbin{\circ\!\!\vee} Q$ for the sequential disjunction of P and Q that prescribes short-circuit evaluation.

Another motivation for short-circuit evaluation arises in the setting in which intermediate evaluation results are not at all memorised throughout the evaluation of a propositional statement, i.e. evaluations of distinct occurrences of an atom in a propositional statement may yield different truth values. A simple example of this phenomenon, taken from Bergstra and Ponse (2011), is the compound statement a pedestrian evaluates before crossing a road with two-way traffic driving on the right:

$$\mathit{look\text{-}left\text{-}and\text{-}check} \mathbin{\circ\!\!\wedge} (\mathit{look\text{-}right\text{-}and\text{-}check} \mathbin{\circ\!\!\wedge} \mathit{look\text{-}left\text{-}and\text{-}check}).$$

This statement requires one, or two, or three atomic evaluations and cannot be simplified to one that requires less. In particular, the evaluation result of the second occurrence of the atom *look-left-and-check* may be *false*, while its first occurrence was evaluated *true*. Observe that the associative variant

$$(\mathit{look\text{-}left\text{-}and\text{-}check} \mathbin{\circ\!\!\wedge} \mathit{look\text{-}right\text{-}and\text{-}check}) \mathbin{\circ\!\!\wedge} \mathit{look\text{-}left\text{-}and\text{-}check}$$

prescribes the same short-circuit evaluation.

In this paper, we restrict evaluations to the truth values *true* and *false* (in the conclusions, we include a few words on a truth value for 'undefined'). Given this restriction, a natural question is 'which logical laws axiomatise short-circuit evaluation?', and in this paper, we provide an answer by considering *short-circuit logic* (SCL). Different SCLs can be distinguished based on the extent to which atomic evaluation results are memorised. We discuss in detail the SCL associated with the last example, which is called *free short-circuit logic* (FSCL); thus, the short-circuit logic in which the second evaluation of an atom can be different from its first evaluation. With help of *evaluation trees*, we can give a simple and natural definition of short-circuit evaluation, and we provide a complete and independent equational axiomatisation of FSCL.

The paper is structured as follows: in Section 2, we define evaluation trees, equational axioms for FSCL and normal forms. In Section 3, we prove that these axioms are complete for the restriction to closed terms. In Section 4, we recall FSCL as defined earlier and consider evaluation strategies and some other variants of SCL that identify more propositional statements and side effects. We end the paper in Section 5 with some conclusions. The paper contains three appendices, containing detailed proofs on independence, normalisation and a quote of an earlier completeness proof.

Note. Considerable parts of the text in the forthcoming sections stem from Staudt (2012), Bergstra, Ponse, and Staudt (2013).

2. Evaluation trees, axioms for free short-circuit logic and normal forms

We define evaluation trees and provide an equational axiomatisation for free short-circuit logic (Section 2.1). Then we define normal forms for closed propositional statements (Section 2.2).

2.1. Evaluation trees and axioms

Given a non-empty set A of atoms, we define evaluation trees, where T represents the truth value *true*, and F represents the truth value *false*.

Definition 2.1.1: The set $\mathcal{T}_A$ of **evaluation trees** over A with leaves in $\{\mathsf{T},\mathsf{F}\}$ is defined inductively by

$$\mathsf{T} \in \mathcal{T}_A, \quad \mathsf{F} \in \mathcal{T}_A, \quad (X \trianglelefteq a \trianglerighteq Y) \in \mathcal{T}_A \text{ for any } X, Y \in \mathcal{T}_A \text{ and } a \in A.$$

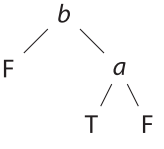
The operator $\_ \trianglelefteq a \trianglerighteq \_$ is called **tree composition over** a. In the evaluation tree $X \trianglelefteq a \trianglerighteq Y$, the root is represented by a, the left branch by X and the right branch by Y. The **depth** $d(..)$ of an evaluation tree is defined by

$$d(\mathsf{T}) = d(\mathsf{F}) = 0 \quad \text{and} \quad d(Y \trianglelefteq a \trianglerighteq Z) = 1 + \max(d(Y), d(Z)).$$

We refer to trees in $\mathcal{T}_A$ as evaluation trees, or trees for short. Next to the formal notation for evaluation trees, we will also use a more pictorial representation. For example, the tree

$$\mathsf{F} \trianglelefteq b \trianglerighteq (\mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F})$$

can be depicted as follows, where $\trianglelefteq$ yields a left branch and $\trianglerighteq$ a right branch:



(1)

In order to define a short-circuit semantics for negation and the sequential connectives, we first define the *leaf replacement* operator, 'replacement' for short, on trees in $\mathcal{T}_A$ as follows. For $X \in \mathcal{T}_A$, the replacement of T with Y and F with Z in X, denoted

$$X[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto Z]$$

is defined recursively by

$$\begin{aligned} \mathsf{T}[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto Z] &= Y, \\ \mathsf{F}[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto Z] &= Z, \\ (X_1 \trianglelefteq a \trianglerighteq X_2)[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto Z] &= X_1[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto Z] \trianglelefteq a \trianglerighteq X_2[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto Z]. \end{aligned}$$

We note that the order in which the replacements of leaves of X is listed is irrelevant, and we adopt the convention of not listing identities inside the brackets, e.g. $X[\mathsf{F} \mapsto Z] = X[\mathsf{T} \mapsto \mathsf{T}, \mathsf{F} \mapsto Z]$. Repeated replacements satisfy the following identity:

$$X[\mathsf{T} \mapsto Y_1, \mathsf{F} \mapsto Z_1][\mathsf{T} \mapsto Y_2, \mathsf{F} \mapsto Z_2] \\ = X[\mathsf{T} \mapsto Y_1[\mathsf{T} \mapsto Y_2, \mathsf{F} \mapsto Z_2],\ \mathsf{F} \mapsto Z_1[\mathsf{T} \mapsto Y_2, \mathsf{F} \mapsto Z_2]]. \quad (2)$$

This identity easily follows by structural induction on evaluation trees.

We define the set $\mathcal{S}_A$ of closed (sequential) propositional statements over A by the following grammar:

$$P ::= a \mid \mathsf{T} \mid \mathsf{F} \mid \neg P \mid P \mathbin{\diamondsuit\!\!\wedge} P \mid P \mathbin{\diamondsuit\!\!\vee} P,$$

where $a \in A$, T is a constant for the truth value *true*, and F for *false*, and $\neg$ is negation. The underlying signature of $\mathcal{S}_A$ is $\Sigma_{\mathrm{SCL}}(A) = \{\diamondsuit\!\!\wedge, \diamondsuit\!\!\vee, \neg, \mathsf{T}, \mathsf{F}, a \mid a \in A\}$. We now have the terminology and notation to formally define the interpretation of propositional statements in $\mathcal{S}_A$ as evaluation trees by a function *se* (abbreviating short-circuit evaluation).

Definition 2.1.2: The unary **short-circuit evaluation function** $se : \mathcal{S}_A \to \mathcal{T}_A$ is defined as follows, where $a \in A$:

$$se(\mathsf{T}) = \mathsf{T}, \qquad se(\neg P) = se(P)[\mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}],$$
$$se(\mathsf{F}) = \mathsf{F}, \qquad se(P \mathbin{\diamondsuit\!\!\wedge} Q) = se(P)[\mathsf{T} \mapsto se(Q)],$$
$$se(a) = \mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F}, \qquad se(P \mathbin{\diamondsuit\!\!\vee} Q) = se(P)[\mathsf{F} \mapsto se(Q)].$$

The overloading of the notation T in $se(\mathsf{T}) = \mathsf{T}$ is harmless and will turn out to be useful (and similarly for F). As a simple example, we derive the evaluation tree for $\neg b \mathbin{\diamondsuit\!\!\wedge} a$:

$$se(\neg b \mathbin{\diamondsuit\!\!\wedge} a) = se(\neg b)[\mathsf{T} \mapsto se(a)] = (\mathsf{F} \trianglelefteq b \trianglerighteq \mathsf{T})[\mathsf{T} \mapsto se(a)] = \mathsf{F} \trianglelefteq b \trianglerighteq (\mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F}),$$

which can be depicted as in (1). Also, $se(\neg(b \mathbin{\diamondsuit\!\!\vee} \neg a)) = \mathsf{F} \trianglelefteq b \trianglerighteq (\mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F})$. An evaluation tree $se(P)$ represents short-circuit evaluation in a way that can be compared to the notion of a truth table for propositional logic (PL) in that it represents each possible evaluation of P. However, there are some important differences with truth tables: in $se(P)$, the sequentiality of P's evaluation is represented, and the same atom may occur multiple times in $se(P)$.

We are interested in the set $\{P = Q \mid P, Q \in \mathcal{S}_A \text{ and } se(P) = se(Q)\}$, and we will show that all equations in this set are derivable from the axioms in Table 2 by equational logic. In Bergstra and Ponse (2012) and Bergstra et al. (2013), we defined *free short-circuit logic*, notation FSCL, and that is why this set of axioms is named EqFSCL. In Section 4.2, we will return to the definition of FSCL.

The axioms and rules for equational logic for axiom set E and terms s, t are those in Table 1 (cf. Burris & Sankappanavar, 2012), where we write $\sigma(s)$ for the application of substitution σ to term s. If for $\Sigma_{\mathrm{SCL}}(A)$-terms s and t, the equation $s = t$ is derivable from the axioms in EqFSCL by equational logic, we write $\mathrm{EqFSCL} \vdash s = t$. Some comments on these axioms:

- Axioms (F1) and (F2) can be seen as defining equations for F and $\diamondsuit\!\!\vee$.
- Axioms (F1)–(F3) imply sequential versions of De Morgan's laws, which implies a left sequential version of the duality principle. Below, we elaborate on this.
- Axioms (F4)–(F7) define some standard identities.
- Axiom (F8) illustrates a typical property of a logic that models immunity for side effects: although it is the case that for each $P \in \mathcal{S}_A$, the evaluation result of $P \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}$ is *false*, the evaluation of P might also yield a side effect. However, the same side effect and evaluation result are obtained upon evaluation of $\neg P \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}$.

Table 1. Axioms and rules of the equational logic E.

Axioms:	$s = t$	for all equations $s = t$ in E	(E)
	$s = s$	for every term t	(Reflexivity)
Rules:	$\dfrac{s = t}{t = s}$		(Symmetry)
	$\dfrac{s = t,\ t = v}{s = v}$		(Transitivity)
	$\dfrac{s_1 = t_1,\ \ldots,\ s_n = t_n}{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n))}$	for every n-ary f	(Congruence)
	$\dfrac{s = t}{\sigma(s) = \sigma(t)}$	for σ a substitution	(Substitution)

Table 2. EqFSCL, a set of axioms for FSCL.

$$\mathsf{F} = \neg\mathsf{T} \tag{F1}$$
$$x \mathbin{\overset{\circ}{\vee}} y = \neg(\neg x \mathbin{\overset{\circ}{\wedge}} \neg y) \tag{F2}$$
$$\neg\neg x = x \tag{F3}$$
$$\mathsf{T} \mathbin{\overset{\circ}{\wedge}} x = x \tag{F4}$$
$$x \mathbin{\overset{\circ}{\vee}} \mathsf{F} = x \tag{F5}$$
$$\mathsf{F} \mathbin{\overset{\circ}{\wedge}} x = \mathsf{F} \tag{F6}$$
$$(x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\wedge}} z = x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\wedge}} z) \tag{F7}$$
$$\neg x \mathbin{\overset{\circ}{\wedge}} \mathsf{F} = x \mathbin{\overset{\circ}{\wedge}} \mathsf{F} \tag{F8}$$
$$(x \mathbin{\overset{\circ}{\wedge}} \mathsf{F}) \mathbin{\overset{\circ}{\vee}} y = (x \mathbin{\overset{\circ}{\vee}} \mathsf{T}) \mathbin{\overset{\circ}{\wedge}} y \tag{F9}$$
$$(x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\vee}} (z \mathbin{\overset{\circ}{\wedge}} \mathsf{F}) = (x \mathbin{\overset{\circ}{\vee}} (z \mathbin{\overset{\circ}{\wedge}} \mathsf{F})) \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\vee}} (z \mathbin{\overset{\circ}{\wedge}} \mathsf{F})) \tag{F10}$$

- Axiom (F9) characterises another property that concerns possible side effects: because the evaluation result of $P \mathbin{\overset{\circ}{\wedge}} \mathsf{F}$ for each possible evaluation of the atoms in P is *false*, Q is always evaluated in $(P \mathbin{\overset{\circ}{\wedge}} \mathsf{F}) \mathbin{\overset{\circ}{\vee}} Q$ and determines the evaluation result. For a similar reason, Q is always evaluated in $(P \mathbin{\overset{\circ}{\vee}} \mathsf{T}) \mathbin{\overset{\circ}{\wedge}} Q$ and determines the evaluation result. Note that the evaluations of $P \mathbin{\overset{\circ}{\vee}} \mathsf{T}$ and $P \mathbin{\overset{\circ}{\wedge}} \mathsf{F}$ accumulate the same side effects, which perhaps is more easily seen if one replaces Q by either T or F.
- Axiom (F10) defines a restricted form of right distributivity of $\mathbin{\overset{\circ}{\vee}}$ over $\mathbin{\overset{\circ}{\wedge}}$. This axiom holds because if x evaluates to *true*, both sides further evaluate $y \mathbin{\overset{\circ}{\vee}} (z \mathbin{\overset{\circ}{\wedge}} \mathsf{F})$, and if x evaluates to *false*, $z \mathbin{\overset{\circ}{\wedge}} \mathsf{F}$ determines the further evaluation result (which is then *false*, and by axiom (F6)), $y \mathbin{\overset{\circ}{\vee}} (z \mathbin{\overset{\circ}{\wedge}} \mathsf{F})$ is not evaluated in the right-hand side).

The dual of $P \in \mathcal{S}_A$, notation P^{dl}, is defined as follows (for $a \in A$):

$$\mathsf{T}^{dl} = \mathsf{F}, \qquad a^{dl} = a, \qquad (P \mathbin{\overset{\circ}{\wedge}} Q)^{dl} = P^{dl} \mathbin{\overset{\circ}{\vee}} Q^{dl},$$
$$\mathsf{F}^{dl} = \mathsf{T}, \qquad (\neg P)^{dl} = \neg P^{dl}, \qquad (P \mathbin{\overset{\circ}{\vee}} Q)^{dl} = P^{dl} \mathbin{\overset{\circ}{\wedge}} Q^{dl}.$$

The duality mapping $(\,)^{dl} : \mathcal{S}_A \to \mathcal{S}_A$ is an involution, that is, $(P^{dl})^{dl} = P$. Setting $x^{dl} = x$ for each variable x, the duality principle extends to equations, e.g. the dual of axiom (F7) is

$(x \mathbin{\overset{\circ}{\vee}} y) \mathbin{\overset{\circ}{\vee}} z = x \mathbin{\overset{\circ}{\vee}} (y \mathbin{\overset{\circ}{\vee}} z)$. It immediately follows that EqFSCL satisfies the duality principle, that is, for all terms s, t over $\Sigma_{\mathrm{SCL}}(A)$,

$$\mathrm{EqFSCL} \vdash s = t \quad \Longleftrightarrow \quad \mathrm{EqFSCL} \vdash s^{dl} = t^{dl}.$$

In order to use some standard notation and terminology of model theory, we define a model that follows the definition of the function *se* from Definition 2.1.2.

Definition 2.1.3 (The short-circuit evaluation model): Let $\mathbb{M}_{se}$ be the $\Sigma_{\mathrm{SCL}}(A)$-algebra with domain $Dom\,(\mathbb{M}_{se}) = \{se(P) \mid P \in \mathcal{S}_A\}$ in which the interpretation of the constants is defined by

$$[\![\mathsf{T}]\!]^{\mathbb{M}_{se}} = \mathsf{T}, \quad [\![\mathsf{F}]\!]^{\mathbb{M}_{se}} = \mathsf{F}, \quad [\![a]\!]^{\mathbb{M}_{se}} = \mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F} \quad \text{for all } a \in A,$$

and in which the interpretation of the connectives has equal notation and is defined by

$$\neg X = X[\mathsf{T} \mapsto Y, \mathsf{F} \mapsto \mathsf{T}], \quad X \mathbin{\overset{\circ}{\wedge}} Y = X[\mathsf{T} \mapsto Y], \quad X \mathbin{\overset{\circ}{\vee}} Y = X[\mathsf{F} \mapsto Y].$$

We will show that $\mathbb{M}_{se}$ is an initial model for EqFSCL, that is, if $X \in Dom\,(\mathbb{M}_{se})$ then for some $P \in \mathcal{S}_A$, $X = [\![P]\!]^{\mathbb{M}_{se}}$, and if $\mathbb{M}_{se} \models P = Q$, then $\mathrm{EqFSCL} \vdash P = Q$. The first property holds by construction of $\mathbb{M}_{se}$. Note that for all $P \in \mathcal{S}_A$, $[\![P]\!]^{\mathbb{M}_{se}} = se(P)$ (this follows easily by structural induction). Hence, for all $P, Q \in \mathcal{S}_A$, $\mathbb{M}_{se} \models P = Q$ if, and only if, $se(P) = se(Q)$. However, first of all, we have to show that $\mathbb{M}_{se}$ is indeed a model of the equational logic EqFSCL.

Theorem 2.1.4 (Soundness): *For all* $\Sigma_{\mathrm{SCL}}(A)$*-terms* s, t*, if* $\mathrm{EqFSCL} \vdash s = t$ *then* $\mathbb{M}_{se} \models s = t$.

***Proof*:** It is immediately clear that reflexivity, symmetry and transitivity hold. For congruence, we show only that for each $\Sigma_{\mathrm{SCL}}(A)$-term u,

$$\mathbb{M}_{se} \models s = t \quad \Rightarrow \quad \mathbb{M}_{se} \models u \mathbin{\overset{\circ}{\wedge}} s = u \mathbin{\overset{\circ}{\wedge}} t.$$

Fix u and let i be an interpretation of variables in $\mathbb{M}_{se}$. We have to show $\mathbb{M}_{se}, i \models u \mathbin{\overset{\circ}{\wedge}} s = u \mathbin{\overset{\circ}{\wedge}} t$ if $\mathbb{M}_{se} \models s = t$. Now there must be $P, Q, R \in \mathcal{S}_A$ with $i(s) = se(P)$, $i(t) = se(Q)$, and $i(u) = se(R)$. If $\mathbb{M}_{se} \models s = t$, then $\mathbb{M}_{se}, i \models s = t$, and thus $se(P) = se(Q)$. Hence,

$$se(R)[\mathsf{T} \mapsto se(P)] = se(R)[\mathsf{T} \mapsto se(Q)],$$

and thus $\mathbb{M}_{se}, i \models u \mathbin{\overset{\circ}{\wedge}} s = u \mathbin{\overset{\circ}{\wedge}} t$. In a similar way, it follows that substitution holds.

Verifying the validity of the axioms in EqFSCL is cumbersome, but not difficult. As an example, we show this for (F3). Fix some interpretation i of variables and assume $i(x) = se(P)$, then

$$\mathbb{M}_{se}, i \models \neg\neg se(P) = se(P)[\mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}][\mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}] = se(P),$$

where the latter equality follows from identity (2) for repeated replacements. □

The following result is non-trivial and proved in Section 3.2.

Theorem 2.1.5 (Completeness of EqFSCL for closed terms): *For all* $P, Q \in \mathcal{S}_A$,

$$\text{EqFSCL} \vdash P = Q \quad \Longleftrightarrow \quad \mathbb{M}_{se} \models P = Q.$$

Observe that EqFSCL does *not* imply the following properties:

- idempotence, e.g. $se(a \between a) \neq se(a)$,
- commutativity, e.g. $se(a \between b) \neq se(b \between a)$,
- absorption, e.g. $se(a \between (a \vee b)) \neq se(a)$,
- distributivity, e.g. $se((a \between b) \vee c) \neq se((a \vee c) \between (b \vee c))$.

The following lemma is used in the proof of Theorem 2.1.5. We note that the lemma's identity was presented as an EqFSCL-axiom in Bergstra and Ponse (2012) and is now replaced by the current axiom (F8).

Lemma 2.1.6: $\text{EqFSCL} \vdash (x \vee y) \between (z \between \mathsf{F}) = (\neg x \vee (z \between \mathsf{F})) \between (y \between (z \between \mathsf{F}))$.

***Proof*:**

$$\begin{aligned}
(x \vee y) \between (z \between \mathsf{F}) &= (x \vee y) \between ((z \between \mathsf{F}) \between \mathsf{F}) && \text{by (F6), (F7)}\\
&= ((x \vee y) \between \neg(z \between \mathsf{F})) \between \mathsf{F} && \text{by (F8), (F7)}\\
&= ((\neg x \between \neg y) \vee (z \between \mathsf{F})) \between \mathsf{F} && \text{by (F8), (F2), (F3)}\\
&= ((\neg x \vee (z \between \mathsf{F})) \between (\neg y \vee (z \between \mathsf{F}))) \between \mathsf{F} && \text{by (F10)}\\
&= (\neg x \vee (z \between \mathsf{F})) \between ((\neg y \vee (z \between \mathsf{F})) \between \mathsf{F}) && \text{by (F7)}\\
&= (\neg x \vee (z \between \mathsf{F})) \between ((y \between \neg(z \between \mathsf{F})) \between \mathsf{F}) && \text{by (F8), (F2), (F3)}\\
&= ((\neg x \vee (z \between \mathsf{F})) \between y) \between (\neg(z \between \mathsf{F}) \between \mathsf{F}) && \text{by (F7)}\\
&= ((\neg x \vee (z \between \mathsf{F})) \between y) \between ((z \between \mathsf{F}) \between \mathsf{F}) && \text{by (F8)}\\
&= ((\neg x \vee (z \between \mathsf{F})) \between y) \between (z \between \mathsf{F}) && \text{by (F7), (F6)}\\
&= (\neg x \vee (z \between \mathsf{F})) \between (y \between (z \between \mathsf{F})). && \text{by (F7)}
\end{aligned}$$

□

We conclude this section with some facts about EqFSCL. First, we prove that axioms (F1) and (F3) are derivable from the remaining axioms, and then we show that these remaining axioms are independent. For both results, we used tools from McCune (2008). We note that Lemma 2.1.6 was also checked with the theorem prover *Prover9* from McCune (2008).

Definition 2.1.7: Let $\text{EqFSCL}^- = \text{EqFSCL} \setminus \{(\text{F1}), (\text{F3})\}$.

Proposition 2.1.8: $\text{EqFSCL}^- \setminus \{(\text{F8}), (\text{F10})\} \vdash (\text{F1}), (\text{F3})$.

***Proof*:** Distilled from output of *Prover9* (McCune, 2008). In order to derive axiom (F1), we start with some auxiliary results:

$$\begin{aligned}
\neg x \between \neg \mathsf{F} &= (\neg x \between \neg \mathsf{F}) \vee \mathsf{F} && \text{by (F5)}\\
&= \neg(\neg(\neg x \between \neg \mathsf{F}) \between \neg \mathsf{F}) && \text{by (F2)}\\
&= \neg((x \vee \mathsf{F}) \between \neg \mathsf{F}) && \text{by (F2)}\\
&= \neg(x \between \neg \mathsf{F}), && \text{by (F5)} \qquad \text{(Aux1)}
\end{aligned}$$

hence,

$$\neg\neg x \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F} \overset{(\mathrm{Aux1})}{=} \neg(\neg x \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) \overset{(\mathrm{F2})}{=} x \mathbin{\diamondsuit\!\!\vee} \mathsf{F} \overset{(\mathrm{F5})}{=} x, \tag{Aux2}$$

$$\neg\neg \mathsf{F} \overset{(\mathrm{F4})}{=} \neg(\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) \overset{(\mathrm{Aux1})}{=} \neg \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}, \tag{Aux3}$$

$$\neg \mathsf{F} \overset{(\mathrm{F6})}{=} \neg(\mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) \overset{(\mathrm{Aux1})}{=} \neg \mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}. \tag{Aux4}$$

Next,

$$\begin{aligned} \mathsf{F} &= \neg\neg \mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F} && \text{by (Aux2)} \\ &= (\neg \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F} && \text{by (Aux3)} \\ &= \neg \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} (\neg \mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) && \text{by (F7)} \\ &= \neg \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}, && \text{by (Aux4)} \end{aligned} \tag{Aux5}$$

hence,

$$\neg \mathsf{F} \overset{(\mathrm{Aux5})}{=} \neg(\neg \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) \overset{(\mathrm{F2})}{=} \mathsf{T} \mathbin{\diamondsuit\!\!\vee} \mathsf{F} \overset{(\mathrm{F5})}{=} \mathsf{T}. \tag{Aux6}$$

With these auxiliary results, we derive axiom (F1):

$$\mathsf{F} \overset{(\mathrm{Aux5})}{=} \neg \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F} \overset{(\mathrm{Aux1})}{=} \neg(\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{F}) \overset{(\mathrm{F4})}{=} \neg\neg \mathsf{F} \overset{(\mathrm{Aux6})}{=} \neg \mathsf{T}.$$

Finally, we derive axiom (F3) and start with an auxiliary result:

$$\mathsf{F} \mathbin{\diamondsuit\!\!\vee} \mathsf{T} \overset{(\mathrm{F2})}{=} \neg(\neg \mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{T}) \overset{(\mathrm{Aux6})}{=} \neg(\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg \mathsf{T}) \overset{(\mathrm{F4})}{=} \neg\neg \mathsf{T} \overset{(\mathrm{F1})}{=} \neg \mathsf{F} \overset{(\mathrm{Aux6})}{=} \mathsf{T}, \tag{Aux7}$$

and thus

$$\begin{aligned} \neg\neg x &= \neg(\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} \neg x) && \text{by (F4)} \\ &= \neg(\neg \mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \neg x) && \text{by (Aux6)} \\ &= \mathsf{F} \mathbin{\diamondsuit\!\!\vee} x && \text{by (F2)} \\ &= (\mathsf{F} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}) \mathbin{\diamondsuit\!\!\vee} x && \text{by (F6)} \\ &= (\mathsf{F} \mathbin{\diamondsuit\!\!\vee} \mathsf{T}) \mathbin{\diamondsuit\!\!\wedge} x && \text{by (F9)} \\ &= \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} x && \text{by (Aux7)} \\ &= x. && \text{by (F4)} \end{aligned}$$

□

Theorem 2.1.9: *The axioms of* EqFSCL$^-$ *are independent if A contains at least two atoms.*

***Proof*:** With the tool *Mace4* (McCune, 2008), one easily obtains for each of the axioms of EqFSCL$^-$ an independence model (a model in which that axiom is not valid, while all remaining axioms are). We show one of the eight cases here and defer the remaining cases to Appendix 1.

In order to prove independence of axiom (F10), that is, $(x \mathbin{\circledwedge} y) \mathbin{\circledvee} (z \mathbin{\circledwedge} \mathsf{F}) = (x \mathbin{\circledvee} (z \mathbin{\circledwedge} \mathsf{F})) \mathbin{\circledwedge} (y \mathbin{\circledvee} (z \mathbin{\circledwedge} \mathsf{F}))$, assume $A \supseteq \{a, b\}$, and consider the model $\mathbb{M}$ with domain $\{0, 1, 2, 3\}$ in which the interpretation of the constants is defined by

$$[\![\mathsf{T}]\!]^{\mathbb{M}} = 1, \quad [\![\mathsf{F}]\!]^{\mathbb{M}} = 0, \quad [\![a]\!]^{\mathbb{M}} = 2, \quad [\![c]\!]^{\mathbb{M}} = 3 \text{ for all } c \in A \setminus \{a\},$$

and in which the connectives are defined by

$\neg$	
0	1
1	0
2	2
3	3

$\circledwedge$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	0
3	3	3	3	3

$\circledvee$	0	1	2	3
0	0	1	2	3
1	1	1	1	1
2	2	1	1	1
3	3	3	3	3

Then all axioms from EqFSCL$^-$ \ {(F10)} are valid in $\mathbb{M}$, while $[\![(a \mathbin{\circledwedge} a) \mathbin{\circledvee} (b \mathbin{\circledwedge} \mathsf{F})]\!]^{\mathbb{M}} = 3$ and $[\![(a \mathbin{\circledvee} (b \mathbin{\circledwedge} \mathsf{F})) \mathbin{\circledwedge} (a \mathbin{\circledvee} (b \mathbin{\circledwedge} \mathsf{F}))]\!]^{\mathbb{M}} = 1$. □

2.2. Normal forms

To aid in the forthcoming proof of Theorem 2.1.5, we define normal forms for $\mathcal{S}_A$-terms. When considering trees in $se[\mathcal{S}_A]$ (the image of se for $\mathcal{S}_A$-terms), we note that some trees only have T-leaves, some only F-leaves and some both T-leaves and F-leaves. For any $\mathcal{S}_A$-term P,

$$se(P \mathbin{\circledvee} \mathsf{T})$$

is a tree with only T-leaves, as can easily be seen from the definition of se:

$$se(P \mathbin{\circledvee} \mathsf{T}) = se(P)[\mathsf{F} \mapsto \mathsf{T}].$$

Similarly, for any $\mathcal{S}_A$-term P, $se(P \mathbin{\circledwedge} \mathsf{F})$ is a tree with only F-leaves. The simplest trees in the image of se that have both types of leaves are $se(a)$ and $se(\neg a)$ for $a \in A$.

We define the grammar for our normal form before we motivate it.

Definition 2.2.1: A term $P \in \mathcal{S}_A$ is said to be in **SCL Normal Form** (***SNF***) if it is generated by the following grammar:

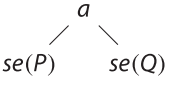
$$\begin{aligned}
P &::= P^{\mathsf{T}} \mid P^{\mathsf{F}} \mid P^{\mathsf{T}} \mathbin{\circledwedge} P^* && (SNF\text{-terms})\\
P^{\mathsf{T}} &::= \mathsf{T} \mid (a \mathbin{\circledwedge} P^{\mathsf{T}}) \mathbin{\circledvee} P^{\mathsf{T}} && (\mathsf{T}\text{-terms})\\
P^{\mathsf{F}} &::= \mathsf{F} \mid (a \mathbin{\circledvee} P^{\mathsf{F}}) \mathbin{\circledwedge} P^{\mathsf{F}} && (\mathsf{F}\text{-terms})\\
P^* &::= P^c \mid P^d && (*\text{-terms})\\
P^c &::= P^\ell \mid P^* \mathbin{\circledwedge} P^d\\
P^d &::= P^\ell \mid P^* \mathbin{\circledvee} P^c\\
P^\ell &::= (a \mathbin{\circledwedge} P^{\mathsf{T}}) \mathbin{\circledvee} P^{\mathsf{F}} \mid (\neg a \mathbin{\circledwedge} P^{\mathsf{T}}) \mathbin{\circledvee} P^{\mathsf{F}} && (\ell\text{-terms})
\end{aligned}$$

where $a \in A$. We refer to P^{T}-forms as T-terms, to P^{F}-forms as F-terms, to P^ℓ-forms as ℓ-terms (the name refers to literal terms) and to P^*-forms as $*$-terms. Finally, a term of the form $P^{\mathsf{T}} \mathbin{\circledwedge} P^*$ is referred to as a T-$*$-term.

For each T-term P, $se(P)$ is a tree with only T-leaves. $\mathcal{S}_A$-terms that have in their se-image only T-leaves will be rewritten to T-terms. Similarly, terms that have in their se-image only F-leaves will be rewritten to F-terms. Note that $\mathbin{\diamondsuit\!\!\vee}$ is right-associative in T-terms, e.g.

$$(a \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} ((b \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} \mathsf{T}) \quad \text{is a T-term, but } ((a \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} (b \mathbin{\diamondsuit\!\!\wedge} \mathsf{T})) \mathbin{\diamondsuit\!\!\vee} \mathsf{T} \text{ is not,}$$

and that $\mathbin{\diamondsuit\!\!\wedge}$ is right-associative in F-terms. Furthermore, the se-images of T-terms and F-terms follow a simple pattern: observe that for $P, Q \in P^{\mathsf{T}}$, $se((a \mathbin{\diamondsuit\!\!\wedge} P) \mathbin{\diamondsuit\!\!\vee} Q)$ is of the form



Before we discuss the T-$*$-terms — the third type of our *SNF* normal forms — we consider the $*$-terms, which are $\mathbin{\diamondsuit\!\!\wedge}$-$\mathbin{\diamondsuit\!\!\vee}$-combinations of ℓ-terms with the restriction that $\mathbin{\diamondsuit\!\!\wedge}$ and $\mathbin{\diamondsuit\!\!\vee}$ associate to the left. This restriction is defined with help of the syntactical categories P^c and P^d. From now on we shall use P^{T}, P^*, etc. both to denote grammatical categories and as variables for terms in those categories. As an example,

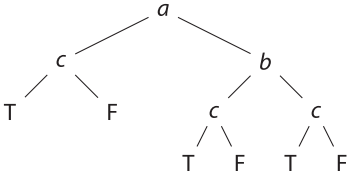
$$(P^\ell \mathbin{\diamondsuit\!\!\wedge} Q^\ell) \mathbin{\diamondsuit\!\!\wedge} R^\ell$$

is a $*$-term (it is in P^c-form), while $P^\ell \mathbin{\diamondsuit\!\!\wedge} (Q^\ell \mathbin{\diamondsuit\!\!\wedge} R^\ell)$ is not a $*$-term. We consider ℓ-terms to be 'basic' in $*$-terms in the sense that they are the smallest grammatical unit that generate se-images in which both T and F occur. More precisely, the se-image of an ℓ-term has precisely one node (its root) that has paths to both T and F.

$\mathcal{S}_A$-terms that have both T and F in their se-image will be rewritten to T-$*$-terms. A T-$*$-term is the conjunction of a T-term and a $*$-term. The first conjunct is necessary to encode a term such as

$$[a \mathbin{\diamondsuit\!\!\vee} (b \mathbin{\diamondsuit\!\!\vee} \mathsf{T})] \mathbin{\diamondsuit\!\!\wedge} c$$

where the evaluation values of a and b are not relevant, but where their side effects may influence the evaluation value of c, as can be clearly seen from its se-image that has three different nodes that model the evaluation of c:



From this example, it can be easily seen that the above T-$*$-term can be also represented as the disjunction of an F-term and a $*$-term, namely of the F-term that encodes $a \mathbin{\diamondsuit\!\!\wedge} (b \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})$ and the $*$-term that encodes c, thus as

$$[(a \mathbin{\diamondsuit\!\!\vee} \mathsf{F}) \mathbin{\diamondsuit\!\!\wedge} ((b \mathbin{\diamondsuit\!\!\vee} \mathsf{F}) \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})] \mathbin{\diamondsuit\!\!\vee} [(c \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} \mathsf{F}].$$

However, we chose to use a T-term and a conjunction for this purpose.

The remainder of this section is concerned with defining and proving correct a normalisation function

$$f : \mathcal{S}_A \to \mathit{SNF}.$$

We will define f recursively using the functions

$$f^n : SNF \to SNF \quad \text{and} \quad f^c : SNF \times SNF \to SNF\,.$$

The first of these will be used to rewrite negated *SNF*-terms to *SNF*-terms and the second to rewrite the conjunction of two *SNF*-terms to an *SNF*-term. By (F2), we have no need for a dedicated function that rewrites the disjunction of two *SNF*-terms to an *SNF*-term. The normalisation function $f : \mathcal{S}_A \to SNF$ is defined recursively, using f^n and f^c, as follows:

$$f(a) = \mathsf{T} \mathbin{\diamondsuit\!\!\wedge} ((a \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\overset{\circ}{\vee}} \mathsf{F}) \tag{3}$$
$$f(\mathsf{T}) = \mathsf{T} \tag{4}$$
$$f(\mathsf{F}) = \mathsf{F} \tag{5}$$
$$f(\neg P) = f^n(f(P)) \tag{6}$$
$$f(P \mathbin{\diamondsuit\!\!\wedge} Q) = f^c(f(P), f(Q)) \tag{7}$$
$$f(P \mathbin{\overset{\circ}{\vee}} Q) = f^n(f^c(f^n(f(P)), f^n(f(Q)))). \tag{8}$$

Observe that $f(a)$ is indeed the unique T-$*$-term with the property that $se(a) = se(f(a))$, and also that $se(\mathsf{T}) = se(f(\mathsf{T}))$ and $se(\mathsf{F}) = se(f(\mathsf{F}))$ (cf. Theorem 2.2.2).

We proceed by defining f^n. Analysing the semantics of T-terms and F-terms together with the definition of se on negations, it becomes clear that f^n must turn T-terms into F-terms and vice versa. We also remark that f^n must preserve the left associativity of the $*$-terms in T-$*$-terms, modulo the associativity within ℓ-terms. We define $f^n : SNF \to SNF$ as follows, using the auxiliary function $f^n_1 : P^* \to P^*$ to push in the negation symbols when negating a T-$*$-term. We note that there is no ambiguity between the different grammatical categories present in an *SNF*-term, i.e. any *SNF*-term is in exactly one of the grammatical categories identified in Definition 2.2.1, and that all right-hand sides are of the intended grammatical category.

$$f^n(\mathsf{T}) = \mathsf{F} \tag{9}$$
$$f^n((a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\overset{\circ}{\vee}} Q^{\mathsf{T}}) = (a \mathbin{\overset{\circ}{\vee}} f^n(Q^{\mathsf{T}})) \mathbin{\diamondsuit\!\!\wedge} f^n(P^{\mathsf{T}}) \tag{10}$$
$$f^n(\mathsf{F}) = \mathsf{T} \tag{11}$$
$$f^n((a \mathbin{\overset{\circ}{\vee}} P^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{F}}) = (a \mathbin{\diamondsuit\!\!\wedge} f^n(Q^{\mathsf{F}})) \mathbin{\overset{\circ}{\vee}} f^n(P^{\mathsf{F}}) \tag{12}$$
$$f^n(P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} Q^*) = P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} f^n_1(Q^*) \tag{13}$$
$$f^n_1((a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\overset{\circ}{\vee}} Q^{\mathsf{F}}) = (\neg a \mathbin{\diamondsuit\!\!\wedge} f^n(Q^{\mathsf{F}})) \mathbin{\overset{\circ}{\vee}} f^n(P^{\mathsf{T}}) \tag{14}$$
$$f^n_1((\neg a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\overset{\circ}{\vee}} Q^{\mathsf{F}}) = (a \mathbin{\diamondsuit\!\!\wedge} f^n(Q^{\mathsf{F}})) \mathbin{\overset{\circ}{\vee}} f^n(P^{\mathsf{T}}) \tag{15}$$
$$f^n_1(P^* \mathbin{\diamondsuit\!\!\wedge} Q^d) = f^n_1(P^*) \mathbin{\overset{\circ}{\vee}} f^n_1(Q^d) \tag{16}$$
$$f^n_1(P^* \mathbin{\overset{\circ}{\vee}} Q^c) = f^n_1(P^*) \mathbin{\diamondsuit\!\!\wedge} f^n_1(Q^c). \tag{17}$$

Now we turn to defining f^c. We distinguish the following cases:

(1) $f^c(P^{\mathsf{T}}, Q)$
(2) $f^c(P^{\mathsf{F}}, Q)$
(3) $f^c(P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} P^*, Q)$

In case (1), it is apparent that the conjunction of a T-term with another term always yields a term of the same grammatical category as the second conjunct. We define f^c recursively by a case distinction on its first argument, and in the second case by a further case distinction on its second argument.

$$f^c(\mathsf{T}, P) = P \tag{18}$$
$$f^c((a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{T}, R^\mathsf{T}) = (a \mathbin{\diamondsuit\!\!\wedge} f^c(P^\mathsf{T}, R^\mathsf{T})) \mathbin{\heartsuit\!\!\vee} f^c(Q^\mathsf{T}, R^\mathsf{T}) \tag{19}$$
$$f^c((a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{T}, R^\mathsf{F}) = (a \mathbin{\heartsuit\!\!\vee} f^c(Q^\mathsf{T}, R^\mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} f^c(P^\mathsf{T}, R^\mathsf{F}) \tag{20}$$
$$f^c((a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{T}, R^\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} S^*) = f^c((a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{T}, R^\mathsf{T}) \mathbin{\diamondsuit\!\!\wedge} S^*. \tag{21}$$

For case (2) (the first argument is an F-term), we make use of (F6). This immediately implies that the conjunction of an F-term with another term is itself an F-term.

$$f^c(P^\mathsf{F}, Q) = P^\mathsf{F} \tag{22}$$

For the remaining case (3) (the first argument is an T-$*$-term), we distinguish three subcases:

(3.1) The second argument is a T-term,
(3.2) The second argument is an F-term, and
(3.3) The second argument is a T-$*$-term.

For case (3.1), we will use an auxiliary function $f_1^c : P^* \times P^\mathsf{T} \to P^*$ to turn conjunctions of a $*$-term with a T-term into $*$-terms. We define f_1^c recursively by a case distinction on its first argument. Together with (F7) (associativity) this allows us to define f^c for this case. Observe that the right-hand sides of the clauses defining f_1^c are indeed $*$-terms.

$$f^c(P^\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} Q^*, R^\mathsf{T}) = P^\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} f_1^c(Q^*, R^\mathsf{T}) \tag{23}$$
$$f_1^c((a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F}, R^\mathsf{T}) = (a \mathbin{\diamondsuit\!\!\wedge} f^c(P^\mathsf{T}, R^\mathsf{T})) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F} \tag{24}$$
$$f_1^c((\neg a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F}, R^\mathsf{T}) = (\neg a \mathbin{\diamondsuit\!\!\wedge} f^c(P^\mathsf{T}, R^\mathsf{T})) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F} \tag{25}$$
$$f_1^c(P^* \mathbin{\diamondsuit\!\!\wedge} Q^d, R^\mathsf{T}) = P^* \mathbin{\diamondsuit\!\!\wedge} f_1^c(Q^d, R^\mathsf{T}) \tag{26}$$
$$f_1^c(P^* \mathbin{\heartsuit\!\!\vee} Q^c, R^\mathsf{T}) = f_1^c(P^*, R^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} f_1^c(Q^c, R^\mathsf{T}). \tag{27}$$

For case (3.2), we need to define $f^c(P^\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} Q^*, R^\mathsf{F})$, which will be an F-term. Using (F7), we reduce this problem to converting Q^* to an F-term, for which we use the auxiliary function $f_2^c : P^* \times P^\mathsf{F} \to P^\mathsf{F}$ that we define recursively by a case distinction on its first argument. Observe that the right-hand sides of the clauses defining f_2^c are all F-terms.

$$f^c(P^\mathsf{T} \mathbin{\diamondsuit\!\!\wedge} Q^*, R^\mathsf{F}) = f^c(P^\mathsf{T}, f_2^c(Q^*, R^\mathsf{F})) \tag{28}$$
$$f_2^c((a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F}, R^\mathsf{F}) = (a \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F}) \mathbin{\diamondsuit\!\!\wedge} f^c(P^\mathsf{T}, R^\mathsf{F}) \tag{29}$$
$$f_2^c((\neg a \mathbin{\diamondsuit\!\!\wedge} P^\mathsf{T}) \mathbin{\heartsuit\!\!\vee} Q^\mathsf{F}, R^\mathsf{F}) = (a \mathbin{\heartsuit\!\!\vee} f^c(P^\mathsf{T}, R^\mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} Q^\mathsf{F} \tag{30}$$
$$f_2^c(P^* \mathbin{\diamondsuit\!\!\wedge} Q^d, R^\mathsf{F}) = f_2^c(P^*, f_2^c(Q^d, R^\mathsf{F})) \tag{31}$$
$$f_2^c(P^* \mathbin{\heartsuit\!\!\vee} Q^c, R^\mathsf{F}) = f_2^c(f^n(f_1^c(P^*, f^n(R^\mathsf{F}))), f_2^c(Q^c, R^\mathsf{F})). \tag{32}$$

For case (3.3), we need to define $f^c(P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} Q^*, R^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} S^*)$. We use the auxiliary function $f_3^c : P^* \times (P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} P^*) \to P^*$ to ensure that the result is a T-$*$-term, and we define f_3^c by a case distinction on its second argument. Observe that the right-hand sides of the clauses defining f_3^c are all $*$-terms.

$$f^c(P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} Q^*, R^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} S^*) = P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} f_3^c(Q^*, R^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} S^*) \tag{33}$$

$$f_3^c(P^*, Q^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} R^\ell) = f_1^c(P^*, Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\wedge} R^\ell \tag{34}$$

$$f_3^c(P^*, Q^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} (R^* \mathbin{\diamondsuit\!\!\wedge} S^d)) = f_3^c(P^*, Q^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} R^*) \mathbin{\diamondsuit\!\!\wedge} S^d \tag{35}$$

$$f_3^c(P^*, Q^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} (R^* \mathbin{\diamondsuit\!\!\vee} S^c)) = f_1^c(P^*, Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\wedge} (R^* \mathbin{\diamondsuit\!\!\vee} S^c). \tag{36}$$

Theorem 2.2.2 (Normal forms): *For any $P \in \mathcal{S}_A$, $f(P)$ terminates, $f(P) \in SNF$ and*

$$\mathrm{EqFSCL} \vdash f(P) = P.$$

In Appendix 1, we first prove a number of lemmas showing that the definitions f^n and f^c are correct and use those to prove the above theorem. We have chosen to define normalisation by a function rather than by a rewriting system because this is more simple and, if desirable, more appropriate for tool implementations.

3. A completeness proof

We analyse the *se*-images of $\mathcal{S}_A$-terms and provide some results on uniqueness of such trees (Section 3.1). Then we define an inverse function of *se* (on the appropriate domain) with which we can complete the proof of the announced completeness theorem (Section 3.2).

3.1. Tree structure and decompositions

In Section 3.2, we will prove that on *SNF* we can invert the function *se*. To do this, we need to prove several structural properties of the trees in $se[SNF]$, the image of *se*. In the definition of *se*, we can see how $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ is assembled from $se(P)$ and $se(Q)$ and similarly for $se(P \mathbin{\diamondsuit\!\!\vee} Q)$. To decompose trees in $se[SNF]$, we will introduce some notation. The trees in the image of *se* are all finite binary trees over A with leaves in $\{\mathsf{T}, \mathsf{F}\}$, i.e. $se[\mathcal{S}_A] \subseteq \mathcal{T}_A$. We will now also consider the set $\mathcal{T}_{A,\triangle}$ of binary trees over A with leaves in $\{\mathsf{T}, \mathsf{F}, \triangle\}$. The triangle will be used as a placeholder when composing or decomposing trees. Replacement of the leaves of trees in $\mathcal{T}_{A,\triangle}$ by trees in $\mathcal{T}_A$ or $\mathcal{T}_{A,\triangle}$ is defined analogous to replacement for trees in $\mathcal{T}_A$, adopting the same notational conventions. As a first example, we have by definition of *se* that $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ can be decomposed as

$$se(P)[\mathsf{T} \mapsto \triangle][\triangle \mapsto se(Q)],$$

where $se(P)[\mathsf{T} \mapsto \triangle] \in \mathcal{T}_{A,\triangle}$ and $se(Q) \in \mathcal{T}_A$. We note that this only works because the trees in the image of *se*, or in $\mathcal{T}_A$ in general, do not contain any triangles. Of course, each tree $X \in \mathcal{T}_A$ has the *trivial decomposition* that involves a replacement of the form $[\triangle \mapsto Y]$, namely

$$\triangle[\triangle \mapsto X].^{1}$$

We start with some simple properties of the *se*-images of T-terms, F-terms and $*$-terms.

Lemma 3.1.1 (Leaf occurrences):

(1) *For any* T*-term* P, $se(P)$ *contains* T, *but not* F,
(2) *For any* F*-term* P, $se(P)$ *contains* F, *but not* T,
(3) *For any* $*$*-term* P, $se(P)$ *contains both* T *and* F.

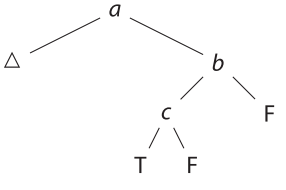
***Proof*:** By induction on the structure of P. A proof of the first two statements is trivial. For the third statement, if P is an ℓ-term, we find that by definition of the grammar of P that one branch from the root of $se(P)$ will only contain T and not F, and for the other branch this is the other way around.

For the induction, we have to consider both $se(P_1 \mathbin{\diamondsuit\!\!\wedge} P_2)$ and $se(P_1 \mathbin{\diamondsuit\!\!\vee} P_2)$. Consider $se(P_1 \mathbin{\diamondsuit\!\!\wedge} P_2)$, which equals by definition $se(P_1)[\mathsf{T} \mapsto se(P_2)]$. By induction, both $se(P_1)$ and $se(P_2)$ contain both T and F, so $se(P_1 \mathbin{\diamondsuit\!\!\wedge} P_2)$ contains both T and F. The case $se(P_1 \mathbin{\diamondsuit\!\!\vee} P_2)$ can be dealt with in a similar way. □

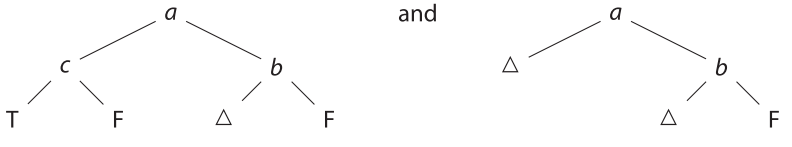
Decompositions of the *se*-image of $*$-terms turn out to be crucial in our approach. As an example, the *se*-image of the $*$-term

$$(P^\ell \mathbin{\diamondsuit\!\!\vee} Q^\ell) \mathbin{\diamondsuit\!\!\wedge} R^\ell \quad \text{with} \quad P^\ell = ((a \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} \mathsf{F}),\ Q^\ell = ((b \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} \mathsf{F}),\ R^\ell = ((c \mathbin{\diamondsuit\!\!\wedge} \mathsf{T}) \mathbin{\diamondsuit\!\!\vee} \mathsf{F})$$

can be decomposed as $X_1[\triangle \mapsto Y]$ with $X_1 \in \mathcal{T}_{A,\triangle}$ as follows:



and $Y = se(R^\ell)$, thus $Y = \mathsf{T} \trianglelefteq c \trianglerighteq \mathsf{F}$, and two other decompositions are $X_2[\triangle \mapsto Y] = X_3[\triangle \mapsto Y]$ with $X_2, X_3 \in \mathcal{T}_{A,\triangle}$ as follows:



Observe that the first two decompositions have the property that Y is a subtree of X_1 and X_2, respectively. Furthermore, observe that $X_3 = se(P^\ell \mathbin{\diamondsuit\!\!\vee} Q^\ell)[\mathsf{T} \mapsto \triangle]$, and hence that this decomposition agrees with the definition of the function *se*. When we want to express that a certain decomposition $X[\triangle \mapsto Y]$ has the property that Y is not a subtree of X, we say that $X[\triangle \mapsto Y]$ is a *strict decomposition*. Finally observe that each of these decompositions satisfies the property that X_i contains T or F, which is a general property of decompositions of $*$-terms and a consequence of Lemma 3.1.3 (see below). The following lemma provides the *se*-image of the rightmost ℓ-term in a $*$-term as a witness.

Lemma 3.1.2 (Witness decomposition): *For all* $*$*-terms* P, $se(P)$ *can be decomposed as* $X[\triangle \mapsto Y]$ *with* $X \in \mathcal{T}_{A,\triangle}$ *and* $Y \in \mathcal{T}_A$ *such that* X *contains* $\triangle$ *and* $Y = se(R)$ *for the rightmost* ℓ*-term* R *in* P. *Note that* X *may be* $\triangle$.

We will refer to Y as ***the witness*** *for this lemma for P.*

Proof**:** By induction on the number of ℓ-terms in P. In the base case, P is an ℓ-term and $se(P) = \triangle[\triangle \mapsto se(P)]$ is the desired decomposition. For the induction, we have to consider both $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ and $se(P \mathbin{\overset{\circ}{\vee}} Q)$.

We start with $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ and let $X[\triangle \mapsto Y]$ be the decomposition for $se(Q)$ which we have by induction hypothesis, so Y is the witness for this lemma for Q and the se-image of its rightmost ℓ-term, say R. Since by definition of se on $\diamondsuit\!\!\wedge$, we have $se(P \mathbin{\diamondsuit\!\!\wedge} Q) = se(P)[\mathsf{T} \mapsto se(Q)]$ we also have

$$se(P \mathbin{\diamondsuit\!\!\wedge} Q) = se(P)[\mathsf{T} \mapsto X[\triangle \mapsto Y]] = se(P)[\mathsf{T} \mapsto X][\triangle \mapsto Y].$$

The last equality is due to the fact that $se(P)$ does not contain any triangles. This gives our desired decomposition: $se(P)[\mathsf{T} \mapsto X]$ contains $\triangle$ because $se(P)$ contains T (Lemma 3.1.1) and X contains $\triangle$, and Y is the se-image of the rightmost ℓ-term R in $P \mathbin{\diamondsuit\!\!\wedge} Q$.

The case for $se(P \mathbin{\overset{\circ}{\vee}} Q)$ is analogous. □

The following lemma illustrates another structural property of trees in the image of $*$-terms under se, namely that each non-trivial decomposition $X[\triangle \mapsto Y]$ of a $*$-term has the property that at least one of T and F occurs in X.

Lemma 3.1.3 (Non-decomposition): *There is no $*$-term P such that $se(P)$ can be decomposed as $X[\triangle \mapsto Y]$ with $X \in \mathcal{T}_{A,\triangle}$ and $Y \in \mathcal{T}_A$, where $X \neq \triangle$ and X contains $\triangle$, but not* T *or* F.

Proof**:** By induction on the number of ℓ-terms in P. Let P be a single ℓ-term. When we analyse the grammar of P we find that one branch from the root of $se(P)$ only contains T and not F, and the other way around for the other branch. Hence if $se(P) = X[\triangle \mapsto Y]$ and X does not contain T or F, then Y contains occurrences of both T and F. Hence, Y must contain the root and $X = \triangle$.

For the induction, we assume that the lemma holds for all $*$-terms that contain fewer ℓ-terms than $P \mathbin{\diamondsuit\!\!\wedge} Q$ and $P \mathbin{\overset{\circ}{\vee}} Q$. We start with the case for $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$. Towards a contradiction, suppose that for some $*$-terms P and Q,

$$se(P \mathbin{\diamondsuit\!\!\wedge} Q) = X[\triangle \mapsto Y] \tag{37}$$

with $X \neq \triangle$ and X not containing any occurrences of T or F. Let Z be the witness of Lemma 3.1.2 for P (so one branch of the root of Z contains only F-leaves, and the other only T-leaves). Observe that $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ has one or more occurrences of the subtree

$$Z[\mathsf{T} \mapsto se(Q)].$$

The interest of this observation is that one branch of the root of this subtree contains only F, and the other branch contains both T and F (because $se(Q)$ does by Lemma 3.1.1). It follows that all occurrences of $Z[\mathsf{T} \mapsto se(Q)]$ in $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ are subtrees in Y after being substituted in X:

- Because X does not contain T and F, Lemma 3.1.1 and (37) imply that Y contains both T and F.

- Assume there is an occurrence of $Z[\mathsf{T} \mapsto se(Q)]$ in $X[\triangle \mapsto Y]$ that has its root in X. Hence the parts of the two branches from this root node that are in X must have $\triangle$ as their leaves. For the branch that only has F-leaves this implies that Y does not contain T, which is a contradiction.

So, Y contains at least one occurrence of $Z[\mathsf{T} \mapsto se(Q)]$, hence

$$se(Q) \text{ is a } \textit{proper} \text{ subtree of } Y. \tag{38}$$

This implies that *each* occurrence of $se(Q)$ in $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ is an occurrence in Y (after being substituted): if this were not the case, the root of $se(Q)$ occurs also in X and the parts of the two branches from this node that are in X must have $\triangle$ as their leaves, which implies that Y after being substituted in X is a proper subtree of $se(Q)$. By (38), this implies that $se(Q)$ is a proper subtree of itself, which is a contradiction.

Because each occurrence of $se(Q)$ in $se(P \mathbin{\diamondsuit\!\!\wedge} Q) = X[\triangle \mapsto Y]$ is an occurrence in Y (after being substituted) and because $se(P \mathbin{\diamondsuit\!\!\wedge} Q) = se(P)[\mathsf{T} \mapsto se(Q)]$, it follows that $se(P) = X[\triangle \mapsto V]$ where V is obtained from Y by replacing all occurrences of the subtree $se(Q)$ by T. But this violates the induction hypothesis. This concludes the induction step for the case of $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$.

A proof for the case $se(P \mathbin{\overset{\circ}{\vee}} Q)$ is symmetric. □

We now arrive at two crucial definitions concerning decompositions. When considering $*$-terms, we already know that $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ can be decomposed as

$$se(P)[\mathsf{T} \mapsto \triangle][\triangle \mapsto se(Q)].$$

Our goal now is to give a definition for a kind of decomposition so that this is the only such decomposition for $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$. We also ensure that $se(P \mathbin{\overset{\circ}{\vee}} Q)$ does not have a decomposition of that kind, so that we can distinguish $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ from $se(P \mathbin{\overset{\circ}{\vee}} Q)$. Similarly, we need to define another kind of decomposition so that $se(P \mathbin{\overset{\circ}{\vee}} Q)$ can only be decomposed as

$$se(P)[\mathsf{F} \mapsto \triangle][\triangle \mapsto se(Q)]$$

and that $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ does not have a decomposition of that kind.

Definition 3.1.4: The pair $(Y, Z) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ is a **candidate conjunction decomposition (ccd)** of $X \in \mathcal{T}_A$, if

- $X = Y[\triangle \mapsto Z]$,
- Y contains $\triangle$,
- Y contains F, but not T, and
- Z contains both T and F.

Similarly, (Y, Z) is a **candidate disjunction decomposition (cdd)** of X, if

- $X = Y[\triangle \mapsto Z]$,
- Y contains $\triangle$,
- Y contains T, but not F, and
- Z contains both T and F.

Observe that any ccd or cdd (Y, Z) is *strict* because Z contains both T and F, and thus cannot be a subtree of Y. A first, crucial property of ccd's and cdd's is the following connection with *se*-images of $*$-terms.

Lemma 3.1.5: *For any $*$-term $P \mathbin{\curlywedge} Q$, $se(P \mathbin{\curlywedge} Q)$ has no cdd. Similarly, for any $*$-term $P \mathbin{\curlyvee} Q$, $se(P \mathbin{\curlyvee} Q)$ has no ccd.*

***Proof*:** We first treat the case for $P \mathbin{\curlywedge} Q$, so $P \in P^*$ and $Q \in P^d$. Towards a contradiction, suppose that (Y, Z) is a cdd of $se(P \mathbin{\curlywedge} Q)$. Let Z' be the witness of Lemma 3.1.2 for P. Observe that $se(P \mathbin{\curlywedge} Q)$ has one or more occurrences of the subtree

$$Z'[\mathsf{T} \mapsto se(Q)].$$

It follows that all occurrences of $Z'[\mathsf{T} \mapsto se(Q)]$ in $se(P \mathbin{\curlywedge} Q)$ are subtrees in Z after being substituted in Y, which can be argued in a similar way as in the proof of Lemma 3.1.3:

- Assume there is an occurrence of $Z'[\mathsf{T} \mapsto se(Q)]$ in $Y[\triangle \mapsto Z]$ that has its root in Y. Following the branch from this node that only has F-leaves and that leads in Y to one or more $\triangle$-leaves, this implies that Z does not contain T, which is a contradiction by definition of a cdd.

So, Z contains at least one occurrence of $Z'[\mathsf{T} \mapsto se(Q)]$. This implies that *each* occurrence of $se(Q)$ in $se(P \mathbin{\curlywedge} Q)$ is an occurrence in Z (after being substituted): if this were not the case, the root of $se(Q)$ occurs in Y and this implies that $se(Q)$ is a proper subtree of itself, which is a contradiction. By definition of *se*, all the occurrences of T in $se(P \mathbin{\curlywedge} Q)$ are in occurrences of the subtree $se(Q)$. Because Y does not contain the root of an $se(Q)$-occurrence, Y does not contain any occurrences of T, so (Y, Z) is not a cdd of $se(P \mathbin{\curlywedge} Q)$. A proof for the case $se(P \mathbin{\curlyvee} Q)$ is symmetric. □

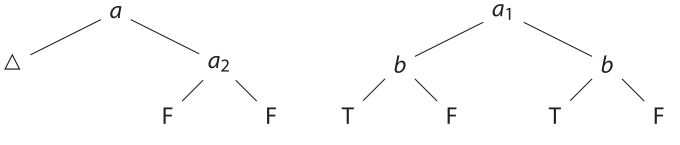
However, the ccd and cdd are not necessarily the decompositions we are looking for, because, for example, $se((P \mathbin{\curlywedge} Q) \mathbin{\curlywedge} R)$ has a ccd

$$(se(P)[\mathsf{T} \mapsto \triangle], se(Q \mathbin{\curlywedge} R)),$$

while the decomposition we need to reconstruct the constituents of a $*$-term is

$$(se(P \mathbin{\curlywedge} Q)[\mathsf{T} \mapsto \triangle], se(R)).$$

A more intricate example of a ccd (Y, Z) that does not produce the constituents of a $*$-term is this pair of trees Y and Z:



It is clear that (Y, Z) is a ccd of $se(P^\ell \mathbin{\curlywedge} Q^\ell)$ with P^ℓ and Q^ℓ these ℓ-terms:

$$P^\ell = (a \mathbin{\curlywedge} ((a_1 \mathbin{\curlywedge} \mathsf{T}) \mathbin{\curlyvee} \mathsf{T})) \mathbin{\curlyvee} ((a_2 \mathbin{\curlyvee} \mathsf{F}) \mathbin{\curlywedge} \mathsf{F}), \quad Q^\ell = (b \mathbin{\curlywedge} \mathsf{T}) \mathbin{\curlyvee} \mathsf{F}.$$

Therefore, we refine Definition 3.1.4 to obtain the decompositions we seek.

Definition 3.1.6: The pair $(Y, Z) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ is a **conjunction decomposition (cd)** of $X \in \mathcal{T}_A$, if it is a ccd of X and there is no other ccd (Y', Z') of X where the depth of Z' is smaller than that of Z.

Similarly, the pair $(Y, Z) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ is a **disjunction decomposition (dd)** of X, if it is a cdd of X and there is no other cdd (Y', Z') of X where the depth of Z' is smaller than that of Z.

Theorem 3.1.7: *For any $*$-term $P \mathbin{\diamondsuit\!\!\wedge} Q$, i.e. with $P \in P^*$ and $Q \in P^d$, $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ has the unique cd*

$$(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$$

and no dd. For any $*$-term $P \mathbin{\diamondsuit\!\!\vee} Q$, i.e. with $P \in P^*$ and $Q \in P^c$, $se(P \mathbin{\diamondsuit\!\!\vee} Q)$ has no cd and its unique dd is

$$(se(P)[\mathsf{F} \mapsto \triangle], se(Q)).$$

***Proof*:** By simultaneous induction on the number of ℓ-terms in $P \mathbin{\diamondsuit\!\!\wedge} Q$ and $P \mathbin{\diamondsuit\!\!\vee} Q$.

In the basis we have to consider, for ℓ-terms P^ℓ and Q^ℓ, the terms $P^\ell \mathbin{\diamondsuit\!\!\wedge} Q^\ell$ and $P^\ell \mathbin{\diamondsuit\!\!\vee} Q^\ell$. By symmetry, it is sufficient to consider the first case. By definition of a ccd and Lemma 3.1.1, $(se(P^\ell)[\mathsf{T} \mapsto \triangle], se(Q^\ell))$ is a ccd of $se(P^\ell \mathbin{\diamondsuit\!\!\wedge} Q^\ell)$. Furthermore observe that the smallest subtree in $se(P^\ell \mathbin{\diamondsuit\!\!\wedge} Q^\ell)$ that contains both T and F is $se(Q^\ell)$. Therefore, $(se(P^\ell)[\mathsf{T} \mapsto \triangle], se(Q^\ell))$ is the *unique* cd of $se(P^\ell \mathbin{\diamondsuit\!\!\wedge} Q^\ell)$. Now for the dd. It suffices to show that there is no cdd of $se(P^\ell \mathbin{\diamondsuit\!\!\wedge} Q^\ell)$ and this follows from Lemma 3.1.5.

For the induction, we assume that the theorem holds for all $*$-terms with fewer ℓ-terms than $P \mathbin{\diamondsuit\!\!\wedge} Q$ and $P \mathbin{\diamondsuit\!\!\vee} Q$. We will first treat the case for $P \mathbin{\diamondsuit\!\!\wedge} Q$ and show that $(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$ is the unique cd of $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$. In this case, observe that for any other ccd (Y, Z) either Z is a proper subtree of $se(Q)$, or vice versa: if this were not the case, then there are occurrences of Z and $se(Q)$ in $Y[\triangle \mapsto Z] = se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ that are disjoint and at least one of the following cases applies:

- Y contains an occurrence of $se(Q)$, and hence of T, which is a contradiction.
- $se(P)[\mathsf{T} \mapsto \triangle]$ contains an occurrence of Z, and hence of T, which is a contradiction.

Hence, by definition of a cd, it suffices to show that there is no ccd (Y, Z) where Z is a proper subtree of $se(Q)$. Towards a contradiction, suppose that such a ccd (Y, Z) does exist. By definition of $*$-terms, Q is either an ℓ-term or a disjunction.

- If Q is an ℓ-term and Z a proper subtree of $se(Q)$, then Z does not contain both T and F because one branch from the root of $se(Q)$ will only contain T and not F, and the other branch vice versa. Therefore, $(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$ is the *unique* cd of $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$.
- If Q is a disjunction and Z a proper subtree of $se(Q)$, then we can decompose $se(Q)$ as $se(Q) = U[\triangle \mapsto Z]$ for some $U \in \mathcal{T}_{A,\triangle}$ that contains but is not equal to $\triangle$ and such that $U[\triangle \mapsto Z]$ is strict, i.e. Z is not a subtree of U. By Lemma 3.1.3, this implies that U contains either T or F.
  - If U contains T, then so does Y, because $Y = se(P)[\mathsf{T} \mapsto U]$, which is the case because

$$\begin{aligned} Y[\triangle \mapsto Z] &= se(P \mathbin{\diamondsuit\!\!\wedge} Q) \\ &= se(P)[\mathsf{T} \mapsto U[\triangle \mapsto Z]] \end{aligned}$$

$$= se(P)[\mathsf{T} \mapsto U][\triangle \mapsto Z],$$

and the only way in which $Y \neq se(P)[\mathsf{T} \mapsto U]$ is possible is that U contains an occurrence of Z, which is excluded because $U[\triangle \mapsto Z]$ is strict. Because Y contains an occurrence of T, (Y, Z) is not a ccd of $se(P \diamondsuit Q)$.

- If U only contains F then (U, Z) is a ccd of $se(Q)$ which violates the induction hypothesis.

Therefore, $(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$ is the *unique* cd of $se(P \diamondsuit Q)$.

Now for the dd. By Lemma 3.1.5, there is no cdd of $se(P \diamondsuit Q)$, so there is neither a dd of $se(P \diamondsuit Q)$. A proof for the case $se(P \heartsuit Q)$ is symmetric. □

At this point, we have the tools necessary to invert *se* on $*$-terms, at least down to the level of ℓ-terms. We can easily detect if a tree in the image of *se* is in the image of P^{ℓ}, because all leaves to the left of the root are one truth value, while all the leaves to the right are the other. To invert *se* on T-$*$-terms, we still need to be able to reconstruct $se(P^{\mathsf{T}})$ and $se(Q^*)$ from $se(P^{\mathsf{T}} \diamondsuit Q^*)$. To this end, we define a T-$*$-decomposition, and as with cd's and dd's we first define a candidate T-$*$-decomposition.

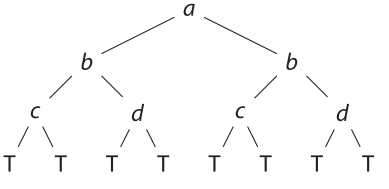
Definition 3.1.8: The pair $(Y, Z) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ is a **candidate** T-$*$**-decomposition (ctsd)** of $X \in \mathcal{T}_A$, if

- $X = Y[\triangle \mapsto Z]$,
- Y does not contain T or F,
- Z contains both T and F,

and there is no decomposition $(U, V) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ of Z such that

- $Z = U[\triangle \mapsto V]$,
- U contains $\triangle$,
- $U \neq \triangle$, and
- U contains neither T nor F.

However, this is not necessarily the decomposition we seek in this case. Consider for example the T-term P^{T} with the following semantics:



and observe that $se(P^{\mathsf{T}} \diamondsuit Q^*)$ has a ctsd

$$(\triangle \trianglelefteq a \trianglerighteq \triangle, (se(Q^*) \trianglelefteq c \trianglerighteq se(Q^*)) \trianglelefteq b \trianglerighteq (se(Q^*) \trianglelefteq d \trianglerighteq se(Q^*))).$$

But the decomposition we seek is $(se(P^{\mathsf{T}})[\mathsf{T} \mapsto \triangle], se(Q^*))$. Hence we will refine the above definition to aid in the theorem below.

Definition 3.1.9: The pair $(Y, Z) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ is a T-$*$**-decomposition (tsd)** of $X \in \mathcal{T}_A$, if it is a ctsd of X and there is no other ctsd (Y', Z') of X where the depth of Z' is smaller than that of Z.

Theorem 3.1.10: *For any* T*-term* P *and* $*$*-term* Q*, the unique tsd of* $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ *is*

$$(se(P)[\mathsf{T} \mapsto \triangle], se(Q)).$$

***Proof*:** First observe that $(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$ is a ctsd because by definition $se(P)[\mathsf{T} \mapsto se(Q)] = se(P \mathbin{\diamondsuit\!\!\wedge} Q)$ and $se(Q)$ is non-decomposable by Lemma 3.1.3.

Towards a contradiction, suppose there exists a ctsd (Y, Z) such that the depth of Z is smaller than that of $se(Q)$. Now either Z is a proper subtree of $se(Q)$, or vice versa, for otherwise there would be occurrences of Z and $se(Q)$ in $Y[\triangle \mapsto Z] = se(P)[\mathsf{T} \mapsto se(Q)]$ that are disjoint and at least one of the following cases applies:

- Y contains an occurrence of $se(Q)$, and hence of T and F, which is a contradiction.
- $se(P)[\mathsf{T} \mapsto \triangle]$ contains an occurrence of Z, and hence of T and F, which is a contradiction.

By definition of a tsd, if suffices to only consider the case that Z is a proper subtree of $se(Q)$. If this is the case, then $se(Q) = U[\triangle \mapsto Z]$ for some $U \in \mathcal{T}_{A,\triangle}$ that is not equal to $\triangle$ and does not contain T or F (because then Y would too). But this violates Lemma 3.1.3, which states that no such decomposition exists. Hence, $(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$ is the *unique* tsd of $se(P \mathbin{\diamondsuit\!\!\wedge} Q)$. □

3.2. Defining an inverse and proving completeness

The two decomposition theorems from the previous section enable us to prove the intermediate result that we used in our completeness proof for FSCL. We define three auxiliary functions to aid in our definition of the inverse of se on SNF. Let

$$cd : \mathcal{T}_A \to \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$$

be the function that returns the conjunction decomposition of its argument, dd of the same type its disjunction decomposition and tsd, also of the same type, its T-$*$-decomposition. Naturally, these functions are undefined when their argument does not have a decomposition of the specified type. Each of these functions returns a pair, and we will use cd_1 (dd_1, tsd_1) to denote the first element of this pair and cd_2 (dd_2, tsd_2) to denote the second element.

We define $g : \mathcal{T}_A \to \mathcal{S}_A$ using the functions $g^{\mathsf{T}} : \mathcal{T}_A \to \mathcal{S}_A$ for inverting trees in the image of T-terms and g^{F}, g^{ℓ} and g^* of the same type for inverting trees in the image of F-terms, ℓ-terms and $*$-terms, respectively. These functions are defined as follows:

$$g^{\mathsf{T}}(X) = \begin{cases} \mathsf{T} & \text{if } X = \mathsf{T}, \\ (a \mathbin{\diamondsuit\!\!\wedge} g^{\mathsf{T}}(Y)) \mathbin{\diamondsuit\!\!\vee} g^{\mathsf{T}}(Z) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z. \end{cases} \tag{39}$$

$$g^{\mathsf{F}}(X) = \begin{cases} \mathsf{F} & \text{if } X = \mathsf{F}, \\ (a \mathbin{\diamondsuit\!\!\vee} g^{\mathsf{F}}(Z)) \mathbin{\diamondsuit\!\!\wedge} g^{\mathsf{F}}(Y) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z. \end{cases} \tag{40}$$

$$g^{\ell}(X) = \begin{cases} (a \mathbin{\diamondsuit\!\!\wedge} g^{\mathsf{T}}(Y)) \mathbin{\diamondsuit\!\!\vee} g^{\mathsf{F}}(Z) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z \text{ and } Y \text{ only has T-leaves}, \\ (\neg a \mathbin{\diamondsuit\!\!\wedge} g^{\mathsf{T}}(Z)) \mathbin{\diamondsuit\!\!\vee} g^{\mathsf{F}}(Y) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z \text{ and } Z \text{ only has T-leaves}. \end{cases} \tag{41}$$

$$g^*(X) = \begin{cases} g^*(cd_1(X)[\Delta \mapsto \mathsf{T}]) \mathbin{\diamondsuit\!\!\wedge} g^*(cd_2(X)) & \text{if } X \text{ has a cd,} \\ g^*(dd_1(X)[\Delta \mapsto \mathsf{F}]) \mathbin{\diamondsuit\!\!\vee} g^*(dd_2(X)) & \text{if } X \text{ has a dd,} \\ g^\ell(X) & \text{otherwise.} \end{cases} \tag{42}$$

$$g(X) = \begin{cases} g^{\mathsf{T}}(X) & \text{if } X \text{ has only T-leaves,} \\ g^{\mathsf{F}}(X) & \text{if } X \text{ has only F-leaves,} \\ g^{\mathsf{T}}(tsd_1(X)[\Delta \mapsto \mathsf{T}]) \mathbin{\diamondsuit\!\!\wedge} g^*(tsd_2(X)) & \text{otherwise.} \end{cases} \tag{43}$$

We use the symbol $\equiv$ to denote 'syntactic equivalence', and we have the following result on our normal forms.

Theorem 3.2.1: *For all* $P \in SNF$, $g(se(P)) \equiv P$.

***Proof*:** We first prove that for all T-terms P, $g^{\mathsf{T}}(se(P)) \equiv P$, by induction on P. In the base case $P \equiv \mathsf{T}$ and we have by (39) that $g^{\mathsf{T}}(se(P)) \equiv g^{\mathsf{T}}(\mathsf{T}) \equiv \mathsf{T} \equiv P$. For the inductive case, we have $P \equiv (a \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} R^{\mathsf{T}}$ and

$$\begin{aligned} g^{\mathsf{T}}(se(P)) &\equiv g^{\mathsf{T}}(se(Q^{\mathsf{T}}) \trianglelefteq a \trianglerighteq se(R^{\mathsf{T}})) && \text{by definition of } se \\ &\equiv (a \mathbin{\diamondsuit\!\!\wedge} g^{\mathsf{T}}(se(Q^{\mathsf{T}}))) \mathbin{\diamondsuit\!\!\vee} g^{\mathsf{T}}(se(R^{\mathsf{T}})) && \text{by (39)} \\ &\equiv (a \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} R^{\mathsf{T}} && \text{by induction hypothesis} \\ &\equiv P. \end{aligned}$$

In a similar way, it follows by (40) that for all F-terms P, $g^{\mathsf{F}}(se(P)) \equiv P$.

Next we check that for all ℓ-terms P, $g^\ell(se(P)) \equiv P$. We observe that either $P \equiv (a \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} R^{\mathsf{F}}$ or $P \equiv (\neg a \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} R^{\mathsf{F}}$. In the first case, we have

$$\begin{aligned} g^\ell(se(P)) &\equiv g^\ell(se(Q^{\mathsf{T}}) \trianglelefteq a \trianglerighteq se(R^{\mathsf{F}})) && \text{by definition of } se \\ &\equiv (a \mathbin{\diamondsuit\!\!\wedge} g^{\mathsf{T}}(se(Q^{\mathsf{T}}))) \mathbin{\diamondsuit\!\!\vee} g^{\mathsf{F}}(se(R^{\mathsf{F}})) && \text{by (41), first case} \\ &\equiv (a \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} R^{\mathsf{F}} && \text{as shown above} \\ &\equiv P. \end{aligned}$$

The second case follows in a similar way.

We now prove that for all $*$-terms P, $g^*(se(P)) \equiv P$, by induction on the number of ℓ-terms in P. In the base case, we are dealing with ℓ-terms. Because an ℓ-term has neither a cd nor a dd we have $g^*(se(P)) \equiv g^\ell(se(P)) \equiv P$, where the first identity is by (42) and the second identity was shown above. For the induction, we have either $P \equiv Q \mathbin{\diamondsuit\!\!\wedge} R$ or $P \equiv Q \mathbin{\diamondsuit\!\!\vee} R$. In the first case note that by Theorem 3.1.7, $se(P)$ has a unique cd and no dd. So we have

$$\begin{aligned} g^*(se(P)) &\equiv g^*(cd_1(se(P))[\Delta \mapsto \mathsf{T}]) \mathbin{\diamondsuit\!\!\wedge} g_*(cd_2(se(P))) && \text{by (42)} \\ &\equiv g^*(se(Q)) \mathbin{\diamondsuit\!\!\wedge} g^*(se(R)) && \text{by Theorem 3.1.7} \\ &\equiv Q \mathbin{\diamondsuit\!\!\wedge} R && \text{by induction hypothesis} \\ &\equiv P. \end{aligned}$$

In the second case, again by Theorem 3.1.7, P has a unique dd and no cd. So we have that

$$\begin{aligned} g^*(se(P)) &\equiv g^*(dd_1(se(P))[\triangle \mapsto \mathsf{F}]) \mathbin{\diamondsuit\!\!\vee} g_*(dd_2(se(P))) && \text{by (42)} \\ &\equiv g^*(se(Q)) \mathbin{\diamondsuit\!\!\vee} g^*(se(R)) && \text{by Theorem 3.1.7} \\ &\equiv Q \mathbin{\diamondsuit\!\!\vee} R && \text{by induction hypothesis} \\ &\equiv P. \end{aligned}$$

Finally, we prove the theorem's statement by making a case distinction on the grammatical category of P. If P is a T-term, then $se(P)$ has only T-leaves and hence $g(se(P)) \equiv g^{\mathsf{T}}(se(P)) \equiv P$, where the first identity is by definition (43) of g and the second identity was shown above. If P is an F-term, then $se(P)$ has only F-leaves and hence $g(se(P)) \equiv g^{\mathsf{F}}(se(P)) \equiv P$, where the first identity is by definition (43) of g and the second one was shown above. If P is a T-$*$-term, then it has both T and F-leaves and hence, letting $P \equiv Q \mathbin{\diamondsuit\!\!\wedge} R$,

$$\begin{aligned} g(se(P)) &\equiv g^{\mathsf{T}}(tsd_1(se(P))[\triangle \mapsto \mathsf{T}]) \mathbin{\diamondsuit\!\!\wedge} g^*(tsd_2(se(P))) && \text{by (43)} \\ &\equiv g^{\mathsf{T}}(se(Q)) \mathbin{\diamondsuit\!\!\wedge} g^*(se(R)) && \text{by Theorem 3.1.10} \\ &\equiv Q \mathbin{\diamondsuit\!\!\wedge} R && \text{as shown above} \\ &\equiv P, \end{aligned}$$

which completes the proof. □

Theorem 3.2.2 (Completeness of EqFSCL for closed terms): *For all $P, Q \in \mathcal{S}_A$,*

$$\text{EqFSCL} \vdash P = Q \quad \Longleftrightarrow \quad \mathbb{M}_{se} \models P = Q.$$

***Proof*:** $\Rightarrow$ follows from Theorem 2.1.4. For $\Leftarrow$, assume $\mathbb{M}_{se} \models P = Q$, thus $se(P) = se(Q)$. By Theorem 2.2.2, P is derivably equal to an *SNF*-term P', i.e. EqFSCL $\vdash P = P'$, and Q is derivably equal to an *SNF*-term Q', i.e. EqFSCL $\vdash Q = Q'$. By Theorem 2.1.4, $se(P) = se(P')$ and $se(Q) = se(Q')$, so $g(se(P')) \equiv g(se(Q'))$. By Theorem 3.2.1 it follows that $P' \equiv Q'$ and hence EqFSCL $\vdash P' = Q'$, and thus EqFSCL $\vdash P = Q$. □

4. Short-circuit logic, evaluation strategies, and side effects

We consider Hoare's *conditional*, a ternary connective that can be used for defining the connectives of $\Sigma_{\text{SCL}}(A)$ (Section 4.1). Then we recall the definition of free short-circuit logic (FSCL) that was published earlier and give a partial solution of an open question (Section 4.2). Next we consider some other evaluation strategies leading to short-circuit logics that identify more sequential propositional statements than FSCL does (Section 4.3). Finally, we briefly discuss *side effects* (Section 4.4).

4.1. Hoare's conditional connective

In 1985, Hoare introduced the *conditional* (Hoare, 1985), a ternary connective with notation

$$x \lhd y \rhd z.$$

Table 3. The set CP of axioms for proposition algebra.

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

A more common expression for the conditional $x \lhd y \rhd z$ is

$$\texttt{if } y \texttt{ then } x \texttt{ else } z,$$

which emphasises that y is evaluated *first*, and depending on the outcome of this partial evaluation, either x or z is evaluated, which determines the evaluation result. So, the evaluation strategy prescribed by this form of if-then-else is a prime example of a sequential evaluation strategy. In order to reason algebraically with conditional expressions, Hoare's more 'operator like' notation $x \lhd y \rhd z$ seems indispensable. In Hoare (1985), an equational axiomatisation of propositional logic is provided that only uses the conditional. Furthermore it is described how the binary connectives and negation are expressed in his set-up (however, the sequential nature of the conditional's evaluation is not discussed in this paper). This axiomatisation over the signature $\Sigma_{\mathrm{CP}}(A) = \{\_ \lhd \_ \rhd \_, \mathsf{T}, \mathsf{F}, a \mid a \in A\}$ consists of eleven axioms and includes the four axioms in Table 3, and some more axioms, for example

$$x \lhd y \rhd (z \lhd y \rhd w) = x \lhd y \rhd w.$$

The four axioms in Table 3, named CP (for Conditional Propositions), establish a complete axiomatisation of *free valuation congruence* defined in Bergstra and Ponse (2011).

By extending the definition of the function *se* (Definition 2.1.2) to closed terms over $\Sigma_{\mathrm{CP}}(A)$ with

$$se(P \lhd Q \rhd R) = se(Q)[\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(R)],$$

we can now characterise the completeness of CP (mentioned above) by

$$\text{For all closed terms } P, Q \text{ over } \Sigma_{\mathrm{CP}}(A), \mathrm{CP} \vdash P = Q \iff se(P) = se(Q). \tag{44}$$

A simple and concise proof of this fact is recorded in Bergstra and Ponse (2015, Theorem 2.11) and repeated in Appendix 1.

With the conditional connective and the constants T and F, the sequential connectives prescribing short-circuit evaluation are definable:

$$\neg x = \mathsf{F} \lhd x \rhd \mathsf{T}, \tag{45}$$
$$x \,{}_{\circ\!}\wedge\, y = y \lhd x \rhd \mathsf{F}, \tag{46}$$
$$x \,{}^{\circ}\!\vee\, y = \mathsf{T} \lhd x \rhd y. \tag{47}$$

Observe that these equations satisfy the extended definition of the function *se*, that is, $se(\neg P) = se(\mathsf{F} \lhd P \rhd \mathsf{T})$, $se(P \,{}_{\circ\!}\wedge\, Q) = se(Q \lhd P \rhd \mathsf{F})$, and $se(P \,{}^{\circ}\!\vee\, Q) = se(\mathsf{T} \lhd P \rhd Q)$.

Thus, the axioms in Table 3 combined with Equations (45)–(47), say $\mathrm{CP}(\neg, \overset{\circ}{\wedge}, \overset{\circ}{\vee})$, axiomatise equality of evaluation trees for closed terms over the enriched signature $\Sigma_{\mathrm{CP}}(A) \cup \Sigma_{\mathrm{SCL}}(A)$.

4.2. The definition of free short-circuit logic

In Bergstra and Ponse (2012) and Bergstra et al. (2013), a set-up is provided for defining short-circuit logics in a generic way with help of the conditional by restricting the enriched language of $\mathrm{CP}(\neg, \overset{\circ}{\wedge}, \overset{\circ}{\vee})$ to the signature $\Sigma_{\mathrm{SCL}}(A)$. The conditional connective is declared as a *hidden operator*.

Intuitively, a short-circuit logic is a logic that implies all consequences of some CP-axiomatisation that can be expressed in the signature $\{\mathsf{T}, \neg, \overset{\circ}{\wedge}\}$. The definition below uses the export-operator $\square$ of *Module algebra* (Bergstra, Heering, & Klint, 1990) to express this in a concise way: in module algebra, $S \square X$ is the operation that exports the signature S from module X while declaring other signature elements hidden.

Definition 4.2.1: A **short-circuit logic** is a logic that implies the consequences of the module expression

$$\mathrm{SCL} = \{\mathsf{T}, \neg, \overset{\circ}{\wedge}\} \square (\mathrm{CP} \cup \{(45), (46)\}).$$

As a first example, $\mathrm{SCL} \vdash \neg\neg x = x$ can be formally proved as follows:

$$\begin{aligned}
\neg\neg x &= \mathsf{F} \lhd (\mathsf{F} \lhd x \rhd \mathsf{T}) \rhd \mathsf{T} && \text{by (45)} \\
&= (\mathsf{F} \lhd \mathsf{F} \rhd \mathsf{T}) \lhd x \rhd (\mathsf{F} \lhd \mathsf{T} \rhd \mathsf{T}) && \text{by (CP4)} \\
&= \mathsf{T} \lhd x \rhd \mathsf{F} && \text{by (CP2), (CP1)} \\
&= x. && \text{by (CP3)} \qquad (48)
\end{aligned}$$

The constant F and the connective $\overset{\circ}{\vee}$ are not in the exported signature of SCL, but can be easily added to SCL by their defining axioms, just as was done in EqFSCL:

$$\mathrm{SCL} \vdash \neg\mathsf{T} = \mathsf{F} \text{ because } \mathrm{CP} \cup \{(45)\} \vdash \neg\mathsf{T} = \mathsf{F} \lhd \mathsf{T} \rhd \mathsf{T} = \mathsf{F},$$

and $\mathrm{SCL} \vdash x \overset{\circ}{\vee} y = \neg(\neg x \overset{\circ}{\wedge} \neg y)$ because in $\mathrm{CP} \cup \{(45), (46)\}$ one can derive

$$\begin{aligned}
\neg(\neg x \overset{\circ}{\wedge} \neg y) &= \mathsf{F} \lhd (\neg y \lhd [\mathsf{F} \lhd x \rhd \mathsf{T}] \rhd \mathsf{F}) \rhd \mathsf{T} && \text{by definition} \\
&= \mathsf{F} \lhd (\mathsf{F} \lhd x \rhd \neg y) \rhd \mathsf{T} && \text{by (CP4), (CP2), (CP1)} \\
&= (\mathsf{F} \lhd \mathsf{F} \rhd \mathsf{T}) \lhd x \rhd (\mathsf{F} \lhd \neg y \rhd \mathsf{T}) && \text{by (CP4)} \\
&= \mathsf{T} \lhd x \rhd y. && \text{by (CP2), (45), (48)}
\end{aligned}$$

In Bergstra and Ponse (2012) and Bergstra et al. (2013), *free* short-circuit logic is defined as the least identifying short-circuit logic:

Definition 4.2.2: **Free short-circuit logic** (**FSCL**) is the short-circuit logic that implies no other consequences than those of the module expression SCL.

An open problem posed in Bergstra and Ponse (2012) is to prove that for all terms s and t over $\Sigma_{\mathrm{SCL}}(A)$, EqFSCL $\vdash s = t \iff$ FSCL $\vdash s = t$. The following result solves this problem for closed terms only.

Theorem 4.2.3: *For all* $P, Q \in \mathcal{S}_A$, EqFSCL $\vdash P = Q \iff$ FSCL $\vdash P = Q$.

***Proof*:** ($\Rightarrow$) All axioms of EqFSCL are derivable in FSCL. As an example, we derive (F9):

$$\begin{aligned}
(x \mathbin{\overset{\circ}{\wedge}} \mathrm{F}) \mathbin{\overset{\circ}{\vee}} y &= (\mathrm{T} \lhd \mathrm{F} \rhd y) \lhd x \rhd (\mathrm{T} \lhd \mathrm{F} \rhd y) && \text{by definition}\\
&= (y \lhd \mathrm{T} \rhd \mathrm{F}) \lhd x \rhd (y \lhd \mathrm{T} \rhd \mathrm{F}) && \text{by (CP1), (CP2)}\\
&= y \lhd (\mathrm{T} \lhd x \rhd \mathrm{T}) \rhd \mathrm{F} && \text{by (CP4)}\\
&= (x \mathbin{\overset{\circ}{\vee}} \mathrm{T}) \mathbin{\overset{\circ}{\wedge}} y. && \text{by definition}
\end{aligned}$$

($\Leftarrow$) Assume FSCL $\vdash P = Q$. By the extended definition of se, we find with equivalence (44) that $se(P) = se(Q)$. By Theorem 3.2.2, EqFSCL $\vdash P = Q$. □

4.3. More short-circuit logics and evaluation strategies

Following Definition 4.2.2, variants of FSCL that identify more sequential propositions can be easily defined. As an example, adding in SCL's definition to CP the two equation schemes

$$(x \lhd a \rhd y) \lhd a \rhd z = (x \lhd a \rhd x) \lhd a \rhd z \quad \text{and} \quad x \lhd a \rhd (y \lhd a \rhd z) = x \lhd a \rhd (z \lhd a \rhd z), \tag{49}$$

where a ranges over A defines *repetition-proof* SCL (RPSCL), in which subsequent atomic evaluations of a yield the same atomic evaluation results. For example,

$$\mathrm{RPSCL} \vdash a \mathbin{\overset{\circ}{\wedge}} (a \mathbin{\overset{\circ}{\vee}} x) = a \mathbin{\overset{\circ}{\wedge}} a. \tag{50}$$

For RPSCL there exist natural examples (in Section 4.4, we sketch one briefly). Evaluation trees for RPSCL can be defined by a transformation of se-trees according to the axiom schemes (49), see Bergstra and Ponse (2015).

For another example, adding in SCL's definition to CP the two axioms

$$x \lhd y \rhd (z \lhd u \rhd (v \lhd y \rhd w)) = x \lhd y \rhd (z \lhd u \rhd w) \quad \text{and} \quad \mathrm{F} \lhd x \rhd \mathrm{F} = \mathrm{F} \tag{51}$$

defines *static* SCL (see Bergstra et al., 2013), which is a sequential form of propositional logic. Note that the first axiom in (51) and those of CP (in Table 3) imply the axioms schemes (49) (set $u = \mathrm{F}$ and $y = a, y = \mathrm{F} \lhd a \rhd \mathrm{T}$, respectively).

Another sequential evaluation strategy is so-called *full sequential evaluation*, which evaluates *all* atoms in a compound statement from left to right. We use the notations $x \mathbin{\overset{\bullet}{\wedge}} y$ and $x \mathbin{\overset{\bullet}{\vee}} y$ for the connectives that prescribe full sequential evaluation. The setting with only full sequential connectives (thus, without short-circuit connectives) can be called 'free full sequential logic', and an axiomatisation is provided in Staudt (2012). This axiomatisation also comprises axioms (F1) and (F3), and a typical axiom is

$$x \mathbin{\overset{\bullet}{\wedge}} \mathrm{F} = \mathrm{F} \mathbin{\overset{\bullet}{\wedge}} x. \tag{52}$$

With the tool *Prover9* (McCune, 2008) it follows that (F1) is derivable, and with the tool *Mace4* (McCune, 2008) it follows that the remaining axioms in Staudt (2012) are independent (even if $|A| = 1$). Furthermore, both (F1) and (F3) become derivable if the axiom $x \mathbin{\overset{\bullet}{\wedge}} \mathrm{T} = x$ is replaced by $x \mathbin{\overset{\bullet}{\vee}} \mathrm{F} = x$, and the remaining axioms are again independent (even if $|A| = 1$).

As is also noted in Staudt (2012), the 'full sequential connectives' can be defined in terms of $\mathbin{\overset{\circ}{\wedge}}$ and $\mathbin{\overset{\circ}{\vee}}$, and the constants T and F:

$$x \mathbin{\overset{\bullet}{\wedge}} y = (x \mathbin{\overset{\circ}{\vee}} (y \mathbin{\overset{\circ}{\wedge}} \mathrm{F})) \mathbin{\overset{\circ}{\wedge}} y \quad \text{and} \quad x \mathbin{\overset{\bullet}{\vee}} y = (x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\vee}} \mathrm{T})) \mathbin{\overset{\circ}{\vee}} y.$$

Hence, full sequential evaluation can be seen as a special case of short-circuit evaluation. For example, it is a simple exercise to derive the SCL-translation of (52) in EqFSCL.

We finally note that a perhaps interesting variant of FSCL is obtained by leaving out the constants T and F. Such a variant could be motivated by the fact that these constants are usually absent in conditions in imperative programs. However, in most programming languages, the effect of T in a condition can be mimicked by a void equality test such as `(1==1)`, or in an expression-evaluated programming language such as Perl, simply by the number `1` (or any other non-zero number). In 'FSCL without T and F', the only EqFSCL-axioms that remain are (F2), (F3) and (F7), expressing duality and associativity. Moreover, these axioms then yield a complete axiomatisation of this restricted form of *se*-congruence. Note that in this approach, connectives prescribing full sequential evaluation are not definable, hence full sequential evaluation is not a special case of short-circuit evaluation.

However, we think that 'SCL without T and F' does not yield an appropriate point of view: in a sequential logic about truth and falsity one should be able to express the value *true* itself.

4.4. Side effects

Although side effects seem to be well understood in programming (see e.g. Black & Windley, 1996, 1998; Kneuss, Kuncak, & Suter, 2014; Norrish, 1997), they are often explained without a general definition. In the following, we consider side effects in the context of the evaluation of propositional statements. The general question whether the sequential evaluation of a propositional statement has one or more side effects is context dependent. Consider a toy programming language where assignments when evaluated as Boolean expressions always yield *true* and tests evaluate as expected. Some typical observations are these:

(1) Consider the assignment `(v:=5)` and observe its effect in the compound statements

`(v:=5) && (v:=7)` and `(v:=5) && (v==5)`.

In the first statement, we cannot observe any side effect of the first assignment, i.e. changing it to assign a different value will never cause a different evaluation result, not even when the statement is embedded in a larger statement. We can say that the side effect of the first assignment is *unobserved* in this context.

In the second compound statement, however, changing the assigned value will yield a different truth value for the compound statement, and we can say that the side effect of the assignment is *observable* here. Note however that in a larger context such as `((...) && (v:=6)) || (v:=6)` the side effect will again be unobserved.

(2) The side effect of the assignment `(v:=v+1)` is observable in a larger context, as is that of `(v:=v-1)`. The side effect of the compound statement `(v:=v+1) && (v:=v-1)` is however *unobservable*, i.e. unobserved in all contexts. We can say that the side effects of these two assignments cancel out provided these assignments occur adjacently.
(3) The question whether a test like `(f(x)==5)` has an observable side effect cannot be answered without examining the definition of the function `f`. Even if a programmer assumes that evaluating a call of `f` has one or more observable side effects, it is still possible to reason about the equivalence of compound statements containing this test.

The above observations suggest that certain statements such as assignments and tests are natural units for reasoning about side effects, and can be considered atomic when reasoning about Boolean conditions as used in a programming language. According to this view, FSCL preserves side effects of atoms in a very general sense because it identifies only propositional statements with identical evaluation trees.

The setting of short-circuit logic admits formal reasoning about side effects. An example of such reasoning, building on observations 1 and 2 mentioned above, is recorded in Bergstra and Ponse (2015, Example 7.2):

> Assume atoms are of the form $(e{==}e')$ and $(\mathtt{v}{:=}e)$ with $\mathtt{v}$ some program variable and e, e' arithmetical expressions over the integers that may contain $\mathtt{v}$. Furthermore, assume that $(e{==}e')$ evaluates to *true* if and only if e and e' represent the same value, and $(\mathtt{v}{:=}e)$ always evaluates to *true* with the effect that the value of e is assigned to $\mathtt{v}$. Then all atoms satisfy the equation schemes (49), and thus RPSCL applies.[2] Note that the stronger identification $a \mathbin{\diamond\!\!\wedge} a = a$ for all atoms a is not valid: if $\mathtt{v}$ has initial value 0 or 1, the statements
>
> $$((\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\wedge} (\mathtt{v}{:=}\mathtt{v}{+}1)) \mathbin{\diamond\!\!\wedge} (\mathtt{v}{==}2) \quad \text{and} \quad (\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\wedge} (\mathtt{v}{==}2)$$
>
> evaluate to different results. Finally, observe that for all initial values of $\mathtt{v}$ and for all $P \in \mathcal{S}_A$,
>
> $$\text{RPSCL} \vdash (\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\wedge} ((\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\vee} P) = (\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\wedge} (\mathtt{v}{:=}\mathtt{v}{+}1), \tag{53}$$
>
> which agrees with the example in (50).

We note that the set-up of our toy programming language suggests a sequential variant of *Dynamic Logic* (see, e.g. Harel, 1984) in which assignments can be used both as tests and as programs. Such a sequential variant could be appropriate for reasoning about side effects and RPSCL would be its underlying short-circuit logic. However, if we assume that each assignment $(\mathtt{v}{:=}e)$ evaluates to the Boolean value of e, RPSCL would not be the appropriate short-circuit logic. For example, if in (53) we take $P = (\mathtt{v}{==}0)$ and set the initial value of $\mathtt{v}$ to -2, then

$$(\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\wedge} ((\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\vee} (\mathtt{v}{==}0))$$

evaluates to *true*, while $(\mathtt{v}{:=}\mathtt{v}{+}1) \mathbin{\diamond\!\!\wedge} (\mathtt{v}{:=}\mathtt{v}{+}1)$ evaluates to *false*. Hence, under this interpretation of assignments, FSCL is the appropriate short-circuit logic.

5. Conclusions

In this paper we discussed *free short-circuit logic* (FSCL), following earlier research reported on in Bergstra et al. (2013), Staudt (2012), Bergstra and Ponse (2012). In FSCL, intermediate evaluation results are not memorised throughout the evaluation of a propositional statement, so evaluations of distinct occurrences of an atom may yield different truth values. The example on the condition a pedestrian evaluates before crossing a road with two-way traffic provides a clear motivation for this specific type of short-circuit evaluation. The use of dedicated names and notation for connectives that *prescribe* short-circuit evaluation is important in our approach (in the area of computer science, one finds a wide variety of names and notations for short-circuit conjunction, such as 'logical and' and 'conditional and'). The symbols $\mathbin{\circ\!\!\wedge}$ and $\mathbin{\circ\!\!\vee}$, as introduced in Bergstra et al. (1995) for four-valued logic and named (left first) sequential conjunction and sequential disjunction, provide a convenient solution in this case. Furthermore, this paper also contains an interesting exposition of the left sequential connectives for the case of three-valued logics.

We note that defining the short-circuit evaluation function *se* might have been avoided by defining the domain of the short-circuit evaluation model $\mathbb{M}_{se}$ directly from the constants and connectives of $\Sigma_{\mathrm{SCL}}(A)$. We believe, however, that the function *se* captures the operational nature of short-circuit evaluation in a simple and elegant manner and should therefore be made explicit.

A last comment on the ten equational axioms that we selected for our axiomatisation of FSCL (in Bergstra & Ponse, 2012; Bergstra et al., 2013; Staudt, 2012, a slightly different set of axioms is used). Although evaluation trees provide an elegant way to model short-circuit evaluation in the presence of side effects, the equational axioms of EqFSCL seem to grasp the nature of FSCL-identities in a more direct way, and each of these axioms embodies a simple idea. This is in particular the case for (F1) and (F3), and that is why we kept these axioms in our definition of this axiom set (although they can be derived from the remaining ones).

When it comes to reasoning about side effects, we subscribe to Parnas' view 2010:

> Most mainline methods disparage side effects as a bad programming practice. Yet even in well-structured, reliable software, many components do have side effects; side effects are very useful in practice. It is time to investigate methods that deal with side effects as the normal case.

We hope that this paper establishes a step in this direction.

Future work. Some specific questions are these:

- Can Theorem 3.2.2, i.e. EqFSCL $\vdash s = t \iff \mathbb{M}_{se} \models s = t$ be generalised to open terms? (Note that $\Rightarrow$ is proved in Theorem 2.1.4.)
- Is there a simpler proof of Theorem 3.2.2 than the one presented in this paper?
- Can the open problem from Section 4.2, i.e. EqFSCL $\vdash s = t \iff$ FSCL $\vdash s = t$, be solved? (Note that Theorem 4.2.3 solves this for closed terms.)

Furthermore, we aim to provide elegant and independent equational axiomatisations for some other variants of SCL defined in Bergstra and Ponse (2012), Bergstra et al. (2013), or proofs of their non-existence when hidden operators are not involved. And, last but not least, we aim to find fruitful applications for FSCL and the other SCLs we defined.

Notes

1. Also, for each $X \in \mathcal{T}_A$ it follows that $X = X[\triangle \mapsto Y]$ for any $Y \in \mathcal{T}_A$, but we do not consider $X[\triangle \mapsto Y]$ to be a 'decomposition' of X in this case.
2. Of course, not all equations valid in this setting follow from RPSCL, e.g. RPSCL $\nvdash$ `(1==1)` $=$ T.
3. We speak of 'basic forms' instead of normal forms in order to avoid intuitions from term rewriting: for example, the basic form associated with action a is $\mathrm{T} \lhd a \rhd \mathrm{F}$, whereas one would expect that the normal form of the latter is a.
4. Without loss of generality it can be assumed that substitutions happen first in equational proofs, that is, the rule (Substitution) in Table 1 may only be used when $s = t$ is an axiom in EqFSCL (see, e.g. Aceto, Chen, Fokkink, & Ingolfsdottir, 2008).

Acknowledgements

We thank an anonymous reviewer and Inge Bethke for useful suggestions and discussion.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Alban Ponse http://orcid.org/0000-0001-6061-5355

References

Aceto, L., Chen, T., Fokkink, W. J., & Ingolfsdottir, A. (2008). On the axiomatizability of priority. *Mathematical Structures in Computer Science, 18*(1), 5–28.

Bergstra, J. A., Bethke, I., & Rodenburg, P. H. (1995). A propositional logic with 4 values: True, false, divergent and meaningless. *Journal of Applied Non-Classical Logics, 5*(2), 199–218.

Bergstra, J. A., Heering, J., & Klint, P. (1990). Module algebra. *Journal of the ACM, 37*(2), 335–372.

Bergstra, J. A., & Ponse, A. (2011). Proposition algebra. *ACM Transactions on Computational Logic*, *12*(3), 36 p. Article 21.

Bergstra, J. A., & Ponse, A. (2012). Proposition algebra and short-circuit logic. In F. Arbab & M. Sirjani (Eds.), *Proceedings of the 4th International Conference on Fundamentals of Software Engineering (FSEN 2011, Tehran)*: Vol. 7141. *LNCS* (pp. 15–31). Berlin, Heidelberg: Springer-Verlag.

Bergstra, J. A., & Ponse, A. (2015). Evaluation trees for proposition algebra. Retrieved from arXiv:1504.08321v3 [cs.LO].

Bergstra, J.A., Ponse, A., & Staudt, D. J. C. (2013). *Short-circuit logic*. Retrieved from arXiv:1010.3674v4 [cs.LO, math.LO]. (First version appeared in 2010).

Black, P. E., & Windley, P. J. (1996). Inference rules for programming languages with side effects in expressions. In J. von Wright, J. Grundy, & J. Harrison (Eds.), *Theorem proving in higher order logics: 9th International Conference* (pp. 51–60). Berlin, Heidelberg: Springer-Verlag.

Black, P. E., & Windley, P. J. (1998). Formal verification of secure programs in the presence of side effects. In R. H. Sprague (Ed.), *Proceedings of the Thirty-first Hawaii International Conference on System Sciences (HICSS-31)* (Vol. III, pp. 327–334). Los Alamitos, CA: IEEE Computer Science Press.

Burris, S. N., & Sankappanavar, H. P. (2012). A course in universal algebra – The millennium edition. Retrieved from http://www.math.uwaterloo.ca/snburris/htdocs/ualg.html

Dijkstra, E. W. (1976). *A discipline of programming*. Englewood Cliffs, NJ: Prentice Hall Inc.

Gries, D. (1981). *The science of programming*. New York, NY: Springer-Verlag.

Harel, D. (1984). Dynamic logic. In D. Gabbay & F. Günthner (Eds.), *Handbook of philosophical logic* (Vol. II, pp. 497–604). Dordrecht: Reidel.

Hoare, C. A. R. (1985). A couple of novelties in the propositional calculus. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 31*(2), 173–178.

Kneuss, E., Kuncak, V., & Suter, Ph. (2014). Effect analysis for programs with callbacks. In E. Cohen & A. Rybalchenko (Eds.), *VSTTE 2013: Verified software: Theories, tools, experiments*: Vol. 8164. *LNCS* (pp. 48–67). Berlin, Heidelberg: Springer-Verlag.

McCarthy, J. (1963). A basis for a mathematical theory of computation. In P. Braffort & D. Hirschberg (Eds.), *Computer programming and formal systems*: Vol. 35. *Studies in logic and the foundations of mathematics* (pp. 33–70). Amsterdam: North-Holland.

McCune, W. (2008). The GUI: Prover9 and Mace4 with a Graphical User Interface. Prover9-Mace4-v05B.zip (March 14, 2008). Retrieved from https://www.cs.unm.edu/mccune/prover9/gui/v05.html

Norrish, M. (1997). An abstract dynamic semantics for C. Computer Laboratory, University of Cambridge, Technical Report. Retrieved from http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-421.pdf

Parnas, D. L. (2010). Really rethinking 'Formal Methods'. *Computer*, *43*(1), 28–34. IEEE Computer Society (Jan. 2010).

Staudt, D. J. C. (2012, May). Completeness for two left-sequential logics (MSc thesis Logic). University of Amsterdam. Retrieved from arXiv:1206.1936v1 [cs.LO].

Zimmermann, W. & Dold, A. (2003). A framework for modeling the semantics of expression evaluation with abstract state machines. In E. Börger, A. Gargantini, & E. Riccobene (Eds.), *ASM 2003*: Vol. 2589. *LNCS* (pp. 391–406). Berlin, Heidelberg: Springer-Verlag.

Appendix 1. Independence, normalisation, and CP and evaluation trees

A.1. Independence proofs

We prove independence of the axioms of $\mathrm{EqFSCL}^- = \mathrm{EqFSCL} \setminus \{(\mathrm{F1}), (\mathrm{F3})\}$. All independence models that we use for this purpose were generated by *Mace4*, see McCune (2008). The independence of axiom (F10) was shown in the proof of Theorem 2.1.9. For all models $\mathbb{M}$ defined below, we assume $[\![\mathsf{F}]\!]^{\mathbb{M}} = 0$ and $[\![\mathsf{T}]\!]^{\mathbb{M}} = 1$. Furthermore, observe that all refutations below use at most one atom.

A model $\mathbb{M}$ for $\mathrm{EqFSCL}^- \setminus \{(\mathrm{F2})\}$ with domain $\{0, 1\}$ that refutes $\mathsf{F} \mathbin{\mathring{\vee}} \mathsf{F} = \neg(\neg\mathsf{T} \mathbin{\mathring{\wedge}} \neg\mathsf{T})$ is this one:

$\neg$	
0	1
1	1

$\mathring{\wedge}$	0	1
0	0	0
1	0	1

$\mathring{\vee}$	0	1
0	0	0
1	1	0

A model $\mathbb{M}$ for $\mathrm{EqFSCL}^- \setminus \{(\mathrm{F4})\}$ with domain $\{0, 1\}$ that refutes $\mathsf{T} \mathbin{\mathring{\wedge}} \mathsf{F} = \mathsf{F}$ is the following:

$\neg$	
0	0
1	1

$\mathring{\wedge}$	0	1
0	0	0
1	1	1

$\mathring{\vee}$	0	1
0	0	0
1	1	1

A model $\mathbb{M}$ for $\mathrm{EqFSCL}^- \setminus \{(\mathrm{F5})\}$ with domain $\{0, 1\}$ that refutes $\mathsf{T} \mathbin{\mathring{\vee}} \mathsf{F} = \mathsf{T}$ is this one:

$\neg$	
0	0
1	0

$\mathring{\wedge}$	0	1
0	0	0
1	0	1

$\mathring{\vee}$	0	1
0	0	0
1	0	0

A model $\mathbb{M}$ for $\mathrm{EqFSCL}^- \setminus \{(\mathrm{F6})\}$ with domain $\{0, 1, 2\}$ and $[\![a]\!]^{\mathbb{M}} = 2$ that refutes $\mathsf{F} \mathbin{\mathring{\wedge}} a = \mathsf{F}$ is the following:

$\neg$	
0	1
1	0
2	2

$\mathring{\wedge}$	0	1	2
0	0	0	2
1	0	1	2
2	0	2	2

$\mathring{\vee}$	0	1	2
0	0	1	2
1	1	1	2
2	2	1	2

A model $\mathbb{M}$ for EqFSCL$^-$ \ {(F7)} with domain {0, 1, 2, 3} and $[\![a]\!]^{\mathbb{M}} = 2$ that refutes $(a \owedge \mathsf{F}) \owedge a = a \owedge (\mathsf{F} \owedge a)$ is this one:

$\neg$	
0	1
1	0
2	2
3	3

$\owedge$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	3	2	0	3
3	3	3	2	3

$\overline{\vee}$	0	1	2	3
0	0	1	2	3
1	1	1	1	1
2	2	3	1	3
3	3	3	2	3

A model $\mathbb{M}$ for EqFSCL$^-$ \ {(F8)} with domain {0, 1, 2, 3} and $[\![a]\!]^{\mathbb{M}} = 2$ that refutes $\neg a \owedge \mathsf{F} = a \owedge \mathsf{F}$ is the following:

$\neg$	
0	1
1	0
2	3
3	2

$\owedge$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	2	2	2	2
3	3	3	3	3

$\overline{\vee}$	0	1	2	3
0	0	1	2	3
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3

A model $\mathbb{M}$ for EqFSCL$^-$ \ {(F9)} with domain {0, 1, 2, 3, 4} and $[\![a]\!]^{\mathbb{M}} = 2$ that refutes $(a \owedge \mathsf{F}) \overline{\vee} a = (a \overline{\vee} \mathsf{T}) \owedge a$ is this one:

$\neg$	
0	1
1	0
2	2
3	4
4	3

$\owedge$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	3	2	2	3	2
3	3	3	3	3	3
4	3	4	4	3	4

$\overline{\vee}$	0	1	2	3	4
0	0	1	2	3	4
1	1	1	1	1	1
2	2	4	2	2	4
3	3	4	3	3	4
4	4	4	4	4	4

A.2 Correctness of the normalisation function

In order to prove that $f : \mathcal{S}_A \to SNF$ is indeed a normalisation function we need to prove that for all SCL-terms P, $f(P)$ terminates, $f(P) \in SNF$ and EqFSCL $\vdash f(P) = P$. To arrive at this result, we prove several intermediate results about the functions f^n and f^c in the order in which their definitions were presented in Section 2.2. For the sake of brevity, we will not explicitly prove that these functions terminate. To see that each function terminates consider that a termination proof would closely mimic the proof structure of the lemmas dealing with the grammatical categories of the images of these functions.

Lemma A.2.1: *For all* $P \in P^{\mathsf{F}}$ *and* $Q \in P^{\mathsf{T}}$, EqFSCL $\vdash P = P \owedge x$ *and* EqFSCL $\vdash Q = Q \overline{\vee} x$.

***Proof*:** We prove both claims simultaneously by induction. In the base case we have $\mathsf{F} = \mathsf{F} \owedge x$ by axiom (F6). The base case for the second claim follows from that for the first claim by duality.

For the induction we have $(a \overline{\vee} P_1) \owedge P_2 = (a \overline{\vee} P_1) \owedge (P_2 \owedge x)$ by the induction hypothesis and the result follows from (F7). For the second claim we again appeal to duality. □

The equality we showed as an example in Lemma 2.1.6 will prove useful in this appendix, as will the following equalities, which also deal with terms of the form $x \owedge \mathsf{F}$ and $x \overline{\vee} \mathsf{T}$. In the sequel, we refer to the dual of axiom (n) by $(n)'$.

Lemma A.2.2: *The following equations can all be derived from* EqFSCL.

(1) $(x \overline{\vee} (y \owedge \mathsf{F})) \owedge (z \owedge \mathsf{F}) = (\neg x \overline{\vee} (z \owedge \mathsf{F})) \owedge (y \owedge \mathsf{F})$,
(2) $(x \owedge (y \overline{\vee} \mathsf{T})) \overline{\vee} (z \owedge \mathsf{F}) = (x \overline{\vee} (z \owedge \mathsf{F})) \owedge (y \overline{\vee} \mathsf{T})$,
(3) $(x \overline{\vee} \mathsf{T}) \owedge \neg y = \neg((x \overline{\vee} \mathsf{T}) \owedge y)$,
(4) $(x \owedge (y \owedge (z \overline{\vee} \mathsf{T}))) \overline{\vee} (w \owedge (z \overline{\vee} \mathsf{T})) = ((x \owedge y) \overline{\vee} w) \owedge (z \overline{\vee} \mathsf{T})$,
(5) $(x \overline{\vee} ((y \overline{\vee} \mathsf{T}) \owedge (z \owedge \mathsf{F}))) \owedge ((w \overline{\vee} \mathsf{T}) \owedge (z \owedge \mathsf{F})) = ((x \owedge (w \overline{\vee} \mathsf{T})) \overline{\vee} (y \overline{\vee} \mathsf{T})) \owedge (z \owedge \mathsf{F})$,
(6) $(x \overline{\vee} ((y \overline{\vee} \mathsf{T}) \owedge (z \owedge \mathsf{F}))) \owedge (w \owedge \mathsf{F}) = ((\neg x \owedge (y \overline{\vee} \mathsf{T})) \overline{\vee} (w \owedge \mathsf{F})) \owedge (z \owedge \mathsf{F})$.

Table A1. Derivations for the proof of Lemma A.2.2.

(1)	$(x \veebar (y \wedge \mathsf{F})) \wedge (z \wedge \mathsf{F})$	
	$= (\neg x \veebar (z \wedge \mathsf{F})) \wedge ((y \wedge \mathsf{F}) \wedge (z \wedge \mathsf{F}))$	by Lemma 2.1.6
	$= (\neg x \veebar (z \wedge \mathsf{F})) \wedge (y \wedge \mathsf{F}),$	by (F6), (F7)
(2)	$(x \wedge (y \veebar \mathsf{T})) \veebar (z \wedge \mathsf{F})$	
	$= (x \veebar (z \wedge \mathsf{F})) \wedge ((y \veebar \mathsf{T}) \veebar (z \wedge \mathsf{F}))$	by (F10)
	$= (x \veebar (z \wedge \mathsf{F})) \wedge (y \veebar \mathsf{T}),$	by (F6)′, (F7)′
(3)	$(x \veebar \mathsf{T}) \wedge \neg y$	
	$= \neg((\neg x \wedge \mathsf{F}) \veebar y)$	by (F2)
	$= \neg((x \wedge \mathsf{F}) \veebar y)$	by (F8)
	$= \neg((x \veebar \mathsf{T}) \wedge y),$	by (F9)
(4)	$(x \wedge (y \wedge (z \veebar \mathsf{T}))) \veebar (w \wedge (z \veebar \mathsf{T}))$	
	$= ((x \wedge y) \wedge (z \veebar \mathsf{T})) \veebar (w \wedge (z \veebar \mathsf{T}))$	by (F7)
	$= ((x \wedge y) \veebar w) \wedge (z \veebar \mathsf{T}),$	by (F10)′
(5)	$(x \veebar ((y \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F}))) \wedge ((w \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F}))$	
	$= (x \veebar ((y \wedge \mathsf{F}) \veebar (z \wedge \mathsf{F}))) \wedge ((w \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F}))$	by (F9)
	$= ((x \veebar (y \wedge \mathsf{F})) \veebar (z \wedge \mathsf{F})) \wedge ((w \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F}))$	by (F7)′
	$= (\neg(x \veebar (y \wedge \mathsf{F})) \veebar (w \veebar \mathsf{T})) \wedge (z \wedge \mathsf{F})$	by Lemma 2.1.6
	$= ((\neg x \wedge (\neg y \veebar \mathsf{T})) \veebar (w \veebar \mathsf{T})) \wedge (z \wedge \mathsf{F})$	by (F2)′
	$= ((\neg x \wedge (y \veebar \mathsf{T})) \veebar (w \veebar \mathsf{T})) \wedge (z \wedge \mathsf{F})$	by (F8)′
	$= ((\neg x \wedge (y \veebar \mathsf{T})) \veebar (w \veebar (\mathsf{T} \veebar (y \veebar \mathsf{T})))) \wedge (z \wedge \mathsf{F})$	by (F6)′
	$= ((\neg x \wedge (y \veebar \mathsf{T})) \veebar ((w \veebar \mathsf{T}) \veebar (y \veebar \mathsf{T}))) \wedge (z \wedge \mathsf{F})$	by (F7)′
	$= ((x \wedge (w \veebar \mathsf{T})) \veebar (y \veebar \mathsf{T})) \wedge (z \wedge \mathsf{F}),$	by Lemma 2.1.6′
(6)	$(x \veebar ((y \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F}))) \wedge (w \wedge \mathsf{F})$	
	$= (\neg x \veebar (w \wedge \mathsf{F})) \wedge (((y \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F})) \wedge (w \wedge \mathsf{F}))$	by Lemma 2.1.6
	$= (\neg x \veebar (w \wedge \mathsf{F})) \wedge ((y \veebar \mathsf{T}) \wedge (z \wedge \mathsf{F}))$	by (F6), (F7)
	$= ((\neg x \veebar (w \wedge \mathsf{F})) \wedge (y \veebar \mathsf{T})) \wedge (z \wedge \mathsf{F})$	by (F7)
	$= ((\neg x \wedge (y \veebar \mathsf{T})) \veebar ((w \wedge \mathsf{F}) \wedge (y \veebar \mathsf{T}))) \wedge (z \wedge \mathsf{F})$	by (F10)′
	$= ((\neg x \wedge (y \veebar \mathsf{T})) \veebar (w \wedge \mathsf{F})) \wedge (z \wedge \mathsf{F}).$	by (F6), (F7)

***Proof*:** See Table A1. We note that these equations were checked with the theorem prover *Prover9* (McCune, 2008). □

Lemma A.2.3: *For all $P \in SNF$, if P is a* T*-term then $f^n(P)$ is an* F*-term, if it is an* F*-term then $f^n(P)$ is a* T*-term, if it is a* T-$*$*-term then so is $f^n(P)$, and*

$$\mathrm{EqFSCL} \vdash f^n(P) = \neg P.$$

***Proof*:** We first prove the claims for T-terms, by induction on P^{T}. In the base case $f^n(\mathsf{T}) = \mathsf{F}$ by (9), so $f^n(\mathsf{T})$ is an F-term. The claim that $\mathrm{EqFSCL} \vdash f^n(\mathsf{T}) = \neg\mathsf{T}$ is immediate by (F1). For the inductive case, we have that $f^n((a \wedge P^{\mathsf{T}}) \veebar Q^{\mathsf{T}}) = (a \veebar f^n(Q^{\mathsf{T}})) \wedge f^n(P^{\mathsf{T}})$ by (10), where we assume that $f^n(P^{\mathsf{T}})$

and $f^n(Q^{\mathsf{T}})$ are F-terms and that EqFSCL $\vdash f^n(P^{\mathsf{T}}) = \neg P^{\mathsf{T}}$ and EqFSCL $\vdash f^n(Q^{\mathsf{T}}) = \neg Q^{\mathsf{T}}$. It follows from the induction hypothesis that $f^n((a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{T}})$ is an F-term. Furthermore, noting that by the induction hypothesis we may assume that $f^n(P^{\mathsf{T}})$ and $f^n(Q^{\mathsf{T}})$ are F-terms, we have:

$$\begin{aligned}
f^n((a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{T}}) &= (a \ovee f^n(Q^{\mathsf{T}})) \owedge f^n(P^{\mathsf{T}}) && \text{by (10)}\\
&= (a \ovee (f^n(Q^{\mathsf{T}}) \owedge \mathsf{F})) \owedge (f^n(P^{\mathsf{T}}) \owedge \mathsf{F}) && \text{by Lemma A.2.1}\\
&= (\neg a \ovee (f^n(P^{\mathsf{T}}) \owedge \mathsf{F})) \owedge (f^n(Q^{\mathsf{T}}) \owedge \mathsf{F}) && \text{by Lemma A.2.2.1}\\
&= (\neg a \ovee f^n(P^{\mathsf{T}})) \owedge f^n(Q^{\mathsf{T}}) && \text{by Lemma A.2.1}\\
&= (\neg a \ovee \neg P^{\mathsf{T}}) \owedge \neg Q^{\mathsf{T}} && \text{by induction hypothesis}\\
&= \neg((a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{T}}). && \text{by (F2) and its dual}
\end{aligned}$$

For F-terms we prove our claims by induction on P^{F}. In the base case $f^n(\mathsf{F}) = \mathsf{T}$ by (11), so $f^n(\mathsf{F})$ is a T-term. The claim that EqFSCL $\vdash f^n(\mathsf{F}) = \neg\mathsf{F}$ is immediate by (F1)′. For the inductive case we have that $f^n((a \ovee P^{\mathsf{F}}) \owedge Q^{\mathsf{F}}) = (a \owedge f^n(Q^{\mathsf{F}})) \ovee f^n(P^{\mathsf{F}})$ by (12), where we assume that $f^n(P^{\mathsf{F}})$ and $f^n(Q^{\mathsf{F}})$ are T-terms and that EqFSCL $\vdash f^n(P^{\mathsf{F}}) = \neg P^{\mathsf{F}}$ and EqFSCL $\vdash f^n(Q^{\mathsf{F}}) = \neg Q^{\mathsf{F}}$. It follows from the induction hypothesis that $f^n((a \ovee P^{\mathsf{F}}) \owedge Q^{\mathsf{F}})$ is a T-term. Furthermore, noting that by the induction hypothesis we may assume that $f^n(P^{\mathsf{F}})$ and $f^n(Q^{\mathsf{F}})$ are T-terms, the proof of derivably equality is dual to that for $f^n((a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{T}})$.

To prove the lemma for T-$*$-terms, we first verify that the auxiliary function f_1^n returns a $*$-term and that for any $*$-term P, EqFSCL $\vdash f_1^n(P) = \neg P$. We show this by induction on the number of ℓ-terms in P. For the base cases, it is immediate by the above cases for T-terms and F-terms that $f_1^n(P)$ is a $*$-term. Furthermore, if P is an ℓ-term of the form $(a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{F}}$ we have:

$$\begin{aligned}
f_1^n((a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{F}}) &= (\neg a \owedge f^n(Q^{\mathsf{F}})) \ovee f^n(P^{\mathsf{T}}) && \text{by (14)}\\
&= (\neg a \owedge (f^n(Q^{\mathsf{F}}) \ovee \mathsf{T})) \ovee (f^n(P^{\mathsf{T}}) \owedge \mathsf{F}) && \text{by Lemma A.2.1}\\
&= (\neg a \ovee (f^n(P^{\mathsf{T}}) \owedge \mathsf{F})) \owedge (f^n(Q^{\mathsf{F}}) \ovee \mathsf{T}) && \text{by Lemma A.2.2.2}\\
&= (\neg a \ovee f^n(P^{\mathsf{T}})) \owedge f^n(Q^{\mathsf{F}}) && \text{by Lemma A.2.1}\\
&= (\neg a \ovee \neg P^{\mathsf{T}}) \owedge \neg Q^{\mathsf{F}} && \text{by induction hypothesis}\\
&= \neg((a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{F}}). && \text{by (F2) and its dual}
\end{aligned}$$

If P is an ℓ-term of the form $(\neg a \owedge P^{\mathsf{T}}) \ovee Q^{\mathsf{F}}$ the proof proceeds the same, substituting $\neg a$ for a and applying (15) and (F3) where needed. For the inductive step we assume that the result holds for all $*$-terms with fewer ℓ-terms than $P^* \owedge Q^d$ and $P^* \ovee Q^c$. By (16) and (17), each application of f_1^n changes the main connective (not occurring inside an ℓ-term) and hence the result is a $*$-term. Derivable equality is, given the induction hypothesis, an instance of (F2)′.

With this result, we can now see that $f^n(P^{\mathsf{T}} \owedge Q^*)$ is indeed a T-$*$-term. We note that, by the above, Lemma A.2.1 implies that $\neg P^{\mathsf{T}} = \neg P^{\mathsf{T}} \owedge \mathsf{F}$. Now we find that:

$$\begin{aligned}
f^n(P^{\mathsf{T}} \owedge Q^*) &= P^{\mathsf{T}} \owedge f_1^n(Q^*) && \text{by (13)}\\
&= P^{\mathsf{T}} \owedge \neg Q^* && \text{as shown above}\\
&= (P^{\mathsf{T}} \ovee \mathsf{T}) \owedge \neg Q^* && \text{by Lemma A.2.1}\\
&= \neg((P^{\mathsf{T}} \ovee \mathsf{T}) \owedge Q^*) && \text{by Lemma A.2.2.3}\\
&= \neg(P^{\mathsf{T}} \owedge Q^*). && \text{by Lemma A.2.1}
\end{aligned}$$

Hence for all $P \in SNF$, EqFSCL $\vdash f^n(P) = \neg P$. □

Lemma A.2.4: *For any* T*-term* P *and* $Q \in SNF$, $f^c(P,Q)$ *has the same grammatical category as* Q *and*

$$\mathrm{EqFSCL} \vdash f^c(P,Q) = P \curlywedge Q.$$

***Proof*:** By induction on the complexity of the T-term. In the base case, we see that $f^c(\mathsf{T},P) = P$ by (18), which is clearly of the same grammatical category as P. Derivable equality is an instance of (F4).

For the inductive step, we assume that the result holds for all T-terms of lesser complexity than $(a \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{T}}$. The claim about the grammatical category follows immediately from the induction hypothesis. For the claim about derivable equality, we make a case distinction on the grammatical category of the second argument. If the second argument is a T-term, we prove derivable equality as follows:

$$\begin{aligned}
&f^c((a \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{T}}, R^{\mathsf{T}}) \\
&\quad = (a \curlywedge f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \curlyvee f^c(Q^{\mathsf{T}}, R^{\mathsf{T}}) && \text{by (19)} \\
&\quad = (a \curlywedge (P^{\mathsf{T}} \curlywedge R^{\mathsf{T}})) \curlyvee (Q^{\mathsf{T}} \curlywedge R^{\mathsf{T}}) && \text{by induction hypothesis} \\
&\quad = (a \curlywedge (P^{\mathsf{T}} \curlywedge (R^{\mathsf{T}} \curlyvee \mathsf{T}))) \curlyvee (Q^{\mathsf{T}} \curlywedge (R^{\mathsf{T}} \curlyvee \mathsf{T})) && \text{by Lemma A.2.1} \\
&\quad = ((a \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{T}}) \curlywedge (R^{\mathsf{T}} \curlyvee \mathsf{T}) && \text{by Lemma A.2.2.4} \\
&\quad = ((a \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{T}}) \curlywedge R^{\mathsf{T}}. && \text{by Lemma A.2.1}
\end{aligned}$$

If the second argument is an F-term, we prove derivable equality as follows:

$$\begin{aligned}
&f^c((a \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{T}}, R^{\mathsf{F}}) \\
&\quad = (a \curlyvee f^c(Q^{\mathsf{T}}, R^{\mathsf{F}})) \curlywedge f^c(P^{\mathsf{T}}, R^{\mathsf{F}}) && \text{by (20)} \\
&\quad = (a \curlyvee (Q^{\mathsf{T}} \curlywedge R^{\mathsf{F}})) \curlywedge (P^{\mathsf{T}} \curlywedge R^{\mathsf{F}}) && \text{by induction hypothesis} \\
&\quad = (a \curlyvee ((Q^{\mathsf{T}} \curlyvee \mathsf{T}) \curlywedge (R^{\mathsf{F}} \curlywedge \mathsf{F}))) \curlywedge \\
&\qquad ((P^{\mathsf{T}} \curlyvee \mathsf{T}) \curlywedge (R^{\mathsf{F}} \curlywedge \mathsf{F})) && \text{by Lemma A.2.1} \\
&\quad = ((a \curlywedge (P^{\mathsf{T}} \curlyvee \mathsf{T})) \curlyvee (Q^{\mathsf{T}} \curlyvee \mathsf{T})) \curlywedge (R^{\mathsf{F}} \curlywedge \mathsf{F}) && \text{by Lemma A.2.2.5} \\
&\quad = ((a \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{T}}) \curlywedge R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}$$

If the second argument is T-$*$-term, the result follows by (21) from the case where the second argument is a T-term, and (F7). □

Lemma A.2.5: *For any* F*-term* P *and* $Q \in SNF$, $f^c(P,Q)$ *is an* F*-term and*

$$\mathrm{EqFSCL} \vdash f^c(P,Q) = P \curlywedge Q.$$

***Proof*:** The grammatical result is immediate by (22) and the claim about derivable equality follows from Lemma A.2.1, (F7) and (F6). □

Lemma A.2.6: *For any* T*-*$*$*-term* P *and* T*-term* Q, $f^c(P,Q)$ *has the same grammatical category as* P *and*

$$\mathrm{EqFSCL} \vdash f^c(P,Q) = P \curlywedge Q.$$

***Proof*:** By (23) and (F7) it suffices to prove the claims for f_1^c, i.e. that $f_1^c(P^*, Q^{\mathsf{T}})$ is a $*$-term and that $\mathrm{EqFSCL} \vdash f_1^c(P^*, Q^{\mathsf{T}}) = P^* \curlywedge Q^{\mathsf{T}}$. We prove this by induction on the number of ℓ-terms in P^*. In the base case, we deal with ℓ-terms and the grammatical claim follows from Lemma A.2.4. We prove derivable equality as follows, letting $\hat{a} \in \{a, \neg a\}$:

$$\begin{aligned}
f_1^c((\hat{a} \curlywedge P^{\mathsf{T}}) \curlyvee Q^{\mathsf{F}}, R^{\mathsf{T}}) &= (\hat{a} \curlywedge f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \curlyvee Q^{\mathsf{F}} && \text{by (24), (25)} \\
&= (\hat{a} \curlywedge (P^{\mathsf{T}} \curlywedge R^{\mathsf{T}})) \curlyvee Q^{\mathsf{F}} && \text{by Lemma A.2.4}
\end{aligned}$$

$$
\begin{aligned}
&= ((\hat{a} \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}} && \text{by (F7)}\\
&= ((\hat{a} \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\wedge} (R^{\mathsf{T}} \mathbin{\diamondsuit\!\!\vee} \mathsf{T})) \mathbin{\diamondsuit\!\!\vee} (Q^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}) && \text{by Lemma A.2.1}\\
&= ((\hat{a} \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} (Q^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} (R^{\mathsf{T}} \mathbin{\diamondsuit\!\!\vee} \mathsf{T}) && \text{by Lemma A.2.2.2}\\
&= ((\hat{a} \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{T}}. && \text{by Lemma A.2.1}
\end{aligned}
$$

For the induction step, we assume that the result holds for all $*$-terms with fewer ℓ-terms than $P^* \mathbin{\diamondsuit\!\!\wedge} Q^d$ and $P^* \mathbin{\diamondsuit\!\!\vee} Q^c$. In the case of conjunctions, the results follow from (26), the induction hypothesis, and (F7). In the case of disjunctions, the results follow immediately from (27), the induction hypothesis, Lemma A.2.1, and (F10)$'$. □

Lemma A.2.7: *For any* T-$*$-*term* P *and* F-*term* Q, $f^c(P, Q)$ *is an* F-*term and*

$$\mathrm{EqFSCL} \vdash f^c(P, Q) = P \mathbin{\diamondsuit\!\!\wedge} Q.$$

***Proof*:** By (28), Lemma A.2.4 and (F7) it suffices to prove that $f_2^c(P^*, Q^{\mathsf{F}})$ is an F-term and that $\mathrm{EqFSCL} \vdash f_2^c(P^*, Q^{\mathsf{F}}) = P^* \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{F}}$. We prove this by induction on the number of ℓ-terms in P^*. In the base case, we deal with ℓ-terms and the grammatical claim follows from Lemma A.2.4. We derive the remaining claim for ℓ-terms of the form $(a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}$ as:

$$
\begin{aligned}
f_2^c((a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}, R^{\mathsf{F}}) &= (a \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} f^c(P^{\mathsf{T}}, R^{\mathsf{F}}) && \text{by (29)}\\
&= (a \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} (P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}}) && \text{by Lemma A.2.4}\\
&= ((a \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}} && \text{by (F7)}\\
&= ((a \mathbin{\diamondsuit\!\!\vee} (Q^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} (P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\vee} \mathsf{T})) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}} && \text{by Lemma A.2.1}\\
&= ((a \mathbin{\diamondsuit\!\!\wedge} (P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\vee} \mathsf{T})) \mathbin{\diamondsuit\!\!\vee} (Q^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}} && \text{by Lemma A.2.2.2}\\
&= ((a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}
$$

For ℓ-terms of the form $(\neg a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}$, we derive:

$$
\begin{aligned}
&f_2^c((\neg a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}, R^{\mathsf{F}})\\
&\quad= (a \mathbin{\diamondsuit\!\!\vee} f^c(P^{\mathsf{T}}, R^{\mathsf{F}})) \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{F}} && \text{by (30)}\\
&\quad= (a \mathbin{\diamondsuit\!\!\vee} (P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}})) \mathbin{\diamondsuit\!\!\wedge} Q^{\mathsf{F}} && \text{by induction hypothesis}\\
&\quad= (a \mathbin{\diamondsuit\!\!\vee} ((P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\vee} \mathsf{T}) \mathbin{\diamondsuit\!\!\wedge} (R^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}))) \mathbin{\diamondsuit\!\!\wedge} (Q^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}) && \text{by Lemma A.2.1}\\
&\quad= ((\neg a \mathbin{\diamondsuit\!\!\wedge} (P^{\mathsf{T}} \mathbin{\diamondsuit\!\!\vee} \mathsf{T})) \mathbin{\diamondsuit\!\!\vee} (Q^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} (R^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F}) && \text{by Lemma A.2.2.6}\\
&\quad= ((\neg a \mathbin{\diamondsuit\!\!\wedge} P^{\mathsf{T}}) \mathbin{\diamondsuit\!\!\vee} Q^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}
$$

For the induction step, we assume that the result holds for all $*$-terms with fewer ℓ-terms than $P^* \mathbin{\diamondsuit\!\!\wedge} Q^d$ and $P^* \mathbin{\diamondsuit\!\!\vee} Q^c$. In the case of conjunctions, the results follow from (31), the induction hypothesis, and (F7). In the case of disjunctions, note that by Lemma A.2.3 and the proof of Lemma A.2.6, we have that $f^n(f_1^c(P^*, f^n(R^{\mathsf{F}})))$ is a $*$-term with same number of ℓ-terms as P^*. The grammatical result follows from this fact, (32), and the induction hypothesis. Furthermore, noting that by the same argument $f^n(f_1^c(P^*, f^n(R^{\mathsf{F}}))) = \neg(P^* \mathbin{\diamondsuit\!\!\wedge} \neg R^{\mathsf{F}})$, we derive:

$$
\begin{aligned}
f_2^c(P^* \mathbin{\diamondsuit\!\!\vee} Q^c, R^{\mathsf{F}}) &= f_2^c(f^n(f_1^c(P^*, f^n(R^{\mathsf{F}}))), f_2^c(Q^c, R^{\mathsf{F}})) && \text{by (32)}\\
&= f^n(f_1^c(P^*, f^n(R^{\mathsf{F}}))) \mathbin{\diamondsuit\!\!\wedge} (Q^c \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}}) && \text{by induction hypothesis}\\
&= \neg(P^* \mathbin{\diamondsuit\!\!\wedge} \neg R^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} (Q^c \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}}) && \text{as shown above}\\
&= (\neg P^* \mathbin{\diamondsuit\!\!\vee} R^{\mathsf{F}}) \mathbin{\diamondsuit\!\!\wedge} (Q^c \mathbin{\diamondsuit\!\!\wedge} R^{\mathsf{F}}) && \text{by (F3), (F2)}\\
&= (\neg P^* \mathbin{\diamondsuit\!\!\vee} (R^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})) \mathbin{\diamondsuit\!\!\wedge} (Q^c \mathbin{\diamondsuit\!\!\wedge} (R^{\mathsf{F}} \mathbin{\diamondsuit\!\!\wedge} \mathsf{F})) && \text{by Lemma A.2.1}
\end{aligned}
$$

$$= (P^* \curlyvee Q^c) \barwedge (R^{\mathsf{F}} \barwedge \mathsf{F}) \qquad \text{by Lemma 2.1.6}$$
$$= (P^* \curlyvee Q^c) \barwedge R^{\mathsf{F}}. \qquad \text{by Lemma A.2.1}$$

This completes the proof. □

Lemma A.2.8: *For any $P, Q \in SNF$, $f^c(P,Q)$ is in SNF and* EqFSCL $\vdash f^c(P,Q) = P \barwedge Q$.

***Proof*:** By the four preceding lemmas, it suffices to show that

$$f^c(P^{\mathsf{T}} \barwedge Q^*, R^{\mathsf{T}} \barwedge S^*)$$

is in *SNF* and that EqFSCL $\vdash f^c(P^{\mathsf{T}} \barwedge Q^*, R^{\mathsf{T}} \barwedge S^*) = (P^{\mathsf{T}} \barwedge Q^*) \barwedge (R^{\mathsf{T}} \barwedge S^*)$. By (F7) and (33), in turn, it suffices to prove that $f_3^c(P^*, Q^{\mathsf{T}} \barwedge R^*)$ is a $*$-term and that EqFSCL $\vdash f_3^c(P^*, Q^{\mathsf{T}} \barwedge R^*) = P^* \barwedge (Q^{\mathsf{T}} \barwedge R^*)$. We prove this by induction on the number of ℓ-terms in R^*. In the base case we have that $f_3^c(P^*, Q^{\mathsf{T}} \barwedge R^\ell) = f_1^c(P^*, Q^{\mathsf{T}}) \barwedge R^\ell$ by (34) and the lemma's statement follows from Lemma A.2.6 and (F7).

For conjunctions the lemma's statement follows from the induction hypothesis, (F7) and (35), and for disjunctions it follows from Lemma A.2.6, (F7) and (36). □

We can now easily prove Theorem 2.2.2:

Theorem A.2.9 (Normal forms): *For any $P \in \mathcal{S}_A$, $f(P)$ terminates, $f(P) \in SNF$ and*

$$\text{EqFSCL} \vdash f(P) = P.$$

***Proof*:** By induction on the structure of P. If P is an atom, the result follows from (3) and axioms (F4), (F5) and its dual. If P is T or F the result follows from by (4) or (5). For the induction we get the result from definitions (6)–(8), Lemmas A.2.3, A.2.8 and axiom (F2). □

A.3 CP and evaluation trees

We finally show that equivalence (44) holds: this is Theorem A.3.6 below and this text (excluding footnotes) is taken from Bergstra and Ponse (2015).

Let C_A be the set of closed terms over $\Sigma_{\mathrm{CP}}(A)$, and recall *se*'s definition on C_A from Section 4.1.

Definition A.3.1: **Basic forms over** A are defined by the following grammar

$$t ::= \mathsf{T} \mid \mathsf{F} \mid t \lhd a \rhd t \quad \text{for } a \in A.$$

We write BF_A for the set of basic forms over A.

The following lemma's exploit the structure of basic forms.[3]

Lemma A.3.2: *For each $P \in C_A$ there exists $Q \in BF_A$ such that* CP $\vdash P = Q$.

***Proof*:** First we establish an auxiliary result: if P, Q, R are basic forms, then there is a basic form S such that CP $\vdash P \lhd Q \rhd R = S$. This follows by structural induction on Q.

The lemma's statement follows by structural induction on P. The base cases $P \in \{\mathsf{T}, \mathsf{F}, a \mid a \in A\}$ are trivial, and if $P = P_1 \lhd P_2 \rhd P_3$ there exist by induction basic forms Q_i such that CP $\vdash P_i = Q_i$, hence CP $\vdash P_1 \lhd P_2 \rhd P_3 = Q_1 \lhd Q_2 \rhd Q_3$. Now apply the auxiliary result. □

Recall that the symbol $\equiv$ denotes 'syntactic equivalence'.

Lemma A.3.3: *For all basic forms P and Q, $se(P) = se(Q)$ implies $P \equiv Q$.*

***Proof*:** By structural induction on P. The base cases $P \in \{\mathsf{T}, \mathsf{F}\}$ are trivial. If $P \equiv P_1 \lhd a \rhd P_2$, then $Q \notin \{\mathsf{T}, \mathsf{F}\}$ and $Q \not\equiv Q_1 \lhd b \rhd Q_2$ if $b \neq a$, so $Q \equiv Q_1 \lhd a \rhd Q_2$ and $se(P_i) = se(Q_i)$. By induction, we find $P_i \equiv Q_i$, and hence $P \equiv Q$. □

Definition A.3.4: **Free valuation congruence**, notation $=_{se}$, is defined on C_A as follows:

$$P =_{se} Q \quad \Longleftrightarrow \quad se(P) = se(Q).$$

Lemma A.3.5: *Free valuation congruence is a congruence relation.*

***Proof*:** Let $P, Q, R \in C_A$ and assume $P =_{se} P'$, thus $se(P) = se(P')$. Then $se(P \lhd Q \rhd R) = se(Q)[\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(R)] = se(Q)[\mathsf{T} \mapsto se(P'), \mathsf{F} \mapsto se(R)] = se(P' \lhd Q \rhd R)$, and thus $P \lhd Q \rhd R =_{se} P' \lhd Q \rhd R$. The two remaining cases can be proved in a similar way. □

Theorem A.3.6 (Completeness of CP for closed terms): *For all $P, Q \in C_A$,*

$$\mathrm{CP} \vdash P = Q \quad \Longleftrightarrow \quad P =_{se} Q.$$

***Proof*:** We first prove $\Rightarrow$.[4] By Lemma A.3.5, $=_{se}$ is a congruence relation, and it easily follows that closed instances of CP-axioms are valid. In the case of axiom (CP4), this follows from

$$\begin{aligned}
&se(P \lhd (Q \lhd R \rhd S) \rhd U)\\
&\quad = se(Q \lhd R \rhd S)[\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(U)]\\
&\quad = \big(se(R)[\mathsf{T} \mapsto se(Q), \mathsf{F} \mapsto se(S)]\big)\ [\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(U)]\\
&\quad = se(R)[\mathsf{T} \mapsto se(Q)[\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(U)], \mathsf{F} \mapsto\ se(S)[\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(U)]]\\
&\quad = se(R)[\mathsf{T} \mapsto se(P \lhd Q \rhd U), \mathsf{F} \mapsto se(P \lhd S \rhd U)]\\
&\quad = se((P \lhd Q \rhd U) \lhd R \rhd (P \lhd S \rhd U)).
\end{aligned}$$

In order to prove $\Leftarrow$, let $P =_{se} Q$. According to Lemma A.3.2, there exist basic forms P' and Q' such that $\mathrm{CP} \vdash P = P'$ and $\mathrm{CP} \vdash Q = Q'$, so $\mathrm{CP} \vdash P' = Q'$. By ($\Rightarrow$) we find $P' =_{se} Q'$, so by Lemma A.3.3, $P' \equiv Q'$. Hence, $\mathrm{CP} \vdash P = P' = Q' = Q$. □