An independent axiomatization for free short-circuit logic

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

Publication date
2017

Document Version
Final published version

An independent axiomatization for free short-circuit logic

Alban Ponse Daan J.C. Staudt

Section Theory of Computer Science, Informatics Institute

Faculty of Science, University of Amsterdam

https://staff.science.uva.nl/a.ponse https://www.daanstaudt.nl

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 memorized 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 SCL and an independent axiomatization. Finally, we discuss evaluation strategies, some other SCLs, and side effects.

Keywords: Non-commutative conjunction, conditional composition, reactive valuation, sequential connective, short-circuit evaluation, side effect

Contents

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 [9, 10]), or the short-circuited connectives `&&` and `||` as used in programming languages such as C, Go, Java, and Perl.[1]

Short-circuit evaluation is sometimes motivated as a solution for dealing with a compound statement such as

```
(x ~= 0) && (y/x > 5)
```

(which can occur in the condition of an if-then-else or while construct) because this statement will always evaluate to a classical truth value. Note that in this example, the expressions `(x ~= 0)` and `(y/x > 5)` are considered to be propositional variables, or, as we will henceforth call these, "atoms". A perhaps more subtle motivation for short-circuit evaluation arises in the setting in which the evaluation of atoms 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) && (x < 3),
```

the result of which can be different from the short-circuit evaluation of `(x < 3) && (f(x) > 5)` if a *side effect* in the evaluation of `f(x)` changes the value of `x`. These two examples show that sequential conjunction is not a commutative operation. In Section 4 we briefly discuss side effects.

Following [1] 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 memorized 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 [2], is the compound statement a pedestrian evaluates before crossing a road with two-way traffic driving on the right:

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

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

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

prescribes the same short-circuit evaluation.

A natural question is "which logical laws axiomatize short-circuit evaluation?", and in this paper we we provide an answer by defining *short-circuit logic* (SCL). Restricting evaluations to the truth values *true* and *false*, different SCLs can be distinguished based on the extent to which

[1] Cf. `https://en.wikipedia.org/wiki/Short-circuit_evaluation` (Accessed 15 July 2017).

atomic evaluation results are memorized. We define and 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 FSCL and we provide a complete and independent axiomatization.

The paper is structured as follows: in Section 2, FSCL is defined and axiomatized, and normal forms are defined. In Section 3 we provide a detailed completeness proof. In Section 4 we consider evaluation strategies, 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 two appendices, containing detailed proofs on independence and normalization.

Note. Considerable parts of the text in the forthcoming sections stem from [17, 5].

2 Free short-circuit logic

We define evaluation trees, free short-circuit logic (FSCL), and provide an equational axiomatization (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*

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

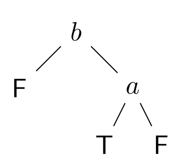
The operator $\_ \trianglelefteq a \trianglerighteq \_$ is called ***post-conditional composition over*** *a. In the evaluation tree $X \trianglelefteq a \trianglerighteq Y$, the root is represented by a, the left branch by X and the right branch by Y. The* ***depth*** *$d(..)$ of an evaluation tree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

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

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:

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

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

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

where $a \in A$, T is a constant for the truth value *true*, and F for *false*, and $\neg$ is negation. 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* $a \in A$*:*

$$\begin{aligned}
se(\mathsf{T}) &= \mathsf{T},\\
se(\mathsf{F}) &= \mathsf{F},\\
se(a) &= \mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F},\\
se(\neg P) &= se(P)[\mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}],\\
se(P \mathbin{{}^{\circ}\!\wedge} Q) &= se(P)[\mathsf{T} \mapsto se(Q)],\\
se(P \mathbin{{}^{\circ}\!\vee} Q) &= se(P)[\mathsf{F} \mapsto se(Q)].
\end{aligned}$$

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 $\neg b \mathbin{{}^{\circ}\!\wedge} a$:

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

which can be depicted as in (1). Also, $se(\neg(b \mathbin{{}^{\circ}\!\vee} \neg a)) = \mathsf{F} \trianglelefteq b \trianglerighteq (\mathsf{T} \trianglelefteq a \trianglerighteq \mathsf{F})$. 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 $se(P)$.

Definition 2.1.3. ***Free short-circuit logic****, notation* FSCL*, is the equational logic defined by identifying equal evaluation trees:*

$$\text{FSCL} \models P = Q$$

if, and only if, $se(P) = se(Q)$*.*

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

Table 1: EqFSCL, a set of axioms for FSCL

The axioms in Table 1 constitute an axiomatization of FSCL. Some comments:

- Axioms (F1) and (F2) can be seen as defining equations for F and $\mathbin{{}^{\circ}\!\!\vee}$. Axioms (F1)-(F3) imply sequential versions of De Morgan's laws, which implies a left-sequential version of the duality principle.
- 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 $P \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F}$ 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 $\neg P \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F}$.
- Axiom (F9) characterizes another property that concerns possible side effects: because the evaluation result of $P \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F}$ for each possible evaluation of the atoms in P is *false*, Q is always evaluated in $(P \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F}) \mathbin{{}^{\circ}\!\!\vee} Q$ and determines the evaluation result. For similar reason, Q is always evaluated in $(P \mathbin{{}^{\circ}\!\!\vee} \mathsf{T}) \mathbin{{}_{\circ}\!\!\wedge} Q$ and determines the evaluation result. Note that the evaluations of $P \mathbin{{}^{\circ}\!\!\vee} \mathsf{T}$ and $P \mathbin{{}_{\circ}\!\!\wedge} \mathsf{F}$ 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 $\mathbin{{}^{\circ}\!\!\vee}$ and (by duality) of $\mathbin{{}_{\circ}\!\!\wedge}$.

We define the *dual* P^d of $P \in \mathcal{S}_A$ as follows:

$$\mathsf{T}^d = \mathsf{F}, \qquad a^d = a \quad (\text{for } a \in A), \qquad (P \mathbin{{}_{\circ}\!\!\wedge} Q)^d = \neg P^d \mathbin{{}^{\circ}\!\!\vee} \neg Q^d,$$
$$\mathsf{F}^d = \mathsf{T}, \qquad (\neg P)^d = \neg(P^d), \qquad (P \mathbin{{}^{\circ}\!\!\vee} Q)^d = \neg P^d \mathbin{{}_{\circ}\!\!\wedge} \neg Q^d.$$

Setting $x^d = x$ for each variable x, the duality principle extends to equational axioms, e.g., $(x \mathbin{{}^{\circ}\!\!\vee} y) \mathbin{{}^{\circ}\!\!\vee} z = x \mathbin{{}^{\circ}\!\!\vee} (y \mathbin{{}^{\circ}\!\!\vee} z)$ is the dual of (F7), and we refer to this dual with the notation (F7)$'$.

The following lemma illustrates the use of these axioms and is used in our completeness proof for FSCL. We note that the lemma's identity was used as an EqFSCL-axiom in our earlier paper [3] and is now replaced by the current axiom (F8).

Lemma 2.1.4. EqFSCL $\vdash (x \mathbin{^{\circ}\!\vee} y) \mathbin{^{\circ}\!\wedge} (z \mathbin{^{\circ}\!\wedge} \mathsf{F}) = (\neg x \mathbin{^{\circ}\!\vee} (z \mathbin{^{\circ}\!\wedge} \mathsf{F})) \mathbin{^{\circ}\!\wedge} (y \mathbin{^{\circ}\!\wedge} (z \mathbin{^{\circ}\!\wedge} \mathsf{F}))$.

Proof.

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

□

Theorem 2.1.5 (Soundness)**.** *For all $P, Q \in \mathcal{S}_A$, if* EqFSCL $\vdash P = Q$ *then* FSCL $\models P = Q$.

Proof. It is immediately clear that identity, symmetry and transitivity are preserved. For congruence we show only that for all $P, Q, R \in \mathcal{S}_A$, FSCL $\models P = Q$ implies FSCL $\models R \mathbin{^{\circ}\!\wedge} P = R \mathbin{^{\circ}\!\wedge} Q$. The other cases proceed in a similar fashion. If FSCL $\models P = Q$, then $se(P) = se(Q)$, so

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

Therefore by definition of se, FSCL $\models R \mathbin{^{\circ}\!\wedge} P = R \mathbin{^{\circ}\!\wedge} Q$.

Verifying the validity of the axioms in EqFSCL is cumbersome, but not difficult. As an example we show this for (F3):

$$se(\neg\neg P) = se(P)[\mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}][\mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}] = se(P)$$

by a trivial structural induction on evaluation trees. □

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

Theorem 3.2.2 (Completeness)**.** For all $P, Q \in \mathcal{S}_A$, if FSCL $\models P = Q$ then EqFSCL $\vdash P = Q$.

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 accessible through the web interface *Son of BirdBrain II* [13], and in Appendix A.1 we explain how we used these tools for our purposes.

Definition 2.1.6. *Let* $\text{EqFSCL}^- = \text{EqFSCL} \setminus \{(\text{F1}), (\text{F3})\}$.

Proposition 2.1.7. $\text{EqFSCL}^- \setminus \{(\text{F8}), (\text{F10})\} \vdash (\text{F1}), (\text{F3})$.

Proof. Distilled from output of the theorem prover *Prover9* [13]. In order to derive axiom (F1) we start with some auxiliary results:

$$\begin{aligned}
\neg x \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F} &= (\neg x \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) \mathbin{{}^{\circ}\!\vee} \mathsf{F} && \text{by (F5)}\\
&= \neg(\neg(\neg x \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) && \text{by (F2)}\\
&= \neg((x \mathbin{{}^{\circ}\!\vee} \mathsf{F}) \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) && \text{by (F2)}\\
&= \neg(x \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}), && \text{by (F5)} \qquad \text{(Aux1)}
\end{aligned}$$

hence,

$$\neg\neg x \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F} \overset{\text{(Aux1)}}{=} \neg(\neg x \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) \overset{\text{(F2)}}{=} x \mathbin{{}^{\circ}\!\vee} \mathsf{F} \overset{\text{(F5)}}{=} x, \qquad \text{(Aux2)}$$

$$\neg\neg\mathsf{F} \overset{\text{(F4)}}{=} \neg(\mathsf{T} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) \overset{\text{(Aux1)}}{=} \neg\mathsf{T} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}, \qquad \text{(Aux3)}$$

$$\neg\mathsf{F} \overset{\text{(F6)}}{=} \neg(\mathsf{F} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) \overset{\text{(Aux1)}}{=} \neg\mathsf{F} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}. \qquad \text{(Aux4)}$$

Next,

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

hence,

$$\neg\mathsf{F} \overset{\text{(Aux5)}}{=} \neg(\neg\mathsf{T} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{F}) \overset{\text{(F2)}}{=} \mathsf{T} \mathbin{{}^{\circ}\!\vee} \mathsf{F} \overset{\text{(F5)}}{=} \mathsf{T}. \qquad \text{(Aux6)}$$

With these auxiliary results we derive axiom (F1):

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

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

$$\mathsf{F} \mathbin{{}^{\circ}\!\vee} \mathsf{T} \overset{\text{(F2)}}{=} \neg(\neg\mathsf{F} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{T}) \overset{\text{(Aux6)}}{=} \neg(\mathsf{T} \mathbin{{}^{\circ}\!\wedge} \neg\mathsf{T}) \overset{\text{(F4)}}{=} \neg\neg\mathsf{T} \overset{\text{(F1)}}{=} \neg\mathsf{F} \overset{\text{(Aux6)}}{=} \mathsf{T}, \qquad \text{(Aux7)}$$

and thus

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

□

Theorem 2.1.8. *The axioms of* EqFSCL$^-$ *are independent if A contains at least two atoms.*

Proof. With the tool *Mace4* [13], 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 A.1.

In order to prove the independence of axiom (F10), that is,

$$(x \leftand y) \leftor (z \leftand \mathsf{F}) = (x \leftor (z \leftand \mathsf{F})) \leftand (y \leftor (z \leftand \mathsf{F})),$$

assume $A \supseteq \{a, b\}$, and consider the model $\mathbb{M}$ with domain $\{0, 1, 2, 3\}$ in which the connectives are defined as follows:

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

$\leftand$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	0
3	3	3	3	3

$\leftor$	0	1	2	3
0	0	1	2	3
1	1	1	1	1
2	2	1	1	1
3	3	3	3	3

and for which the interpretation function ϕ is defined as follows:

$$\phi(\mathsf{F}) = 0, \quad \phi(\mathsf{T}) = 1, \quad \phi(a) = 2, \quad \phi(b) = 3.$$

Then all axioms from EqFSCL$^- \setminus \{(\text{F10})\}$ are valid in $\mathbb{M}$, while $\phi((a \leftand a) \leftor (b \leftand \mathsf{F})) = 3$ and $\phi((a \leftor (b \leftand \mathsf{F})) \leftand (a \leftor (b \leftand \mathsf{F}))) = 1$. □

2.2 Normal forms

To aid in the forthcoming proof of Theorem 3.2.2 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,

$$se(P \leftor \mathsf{T})$$

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

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

Similarly, for any $\mathcal{S}_A$-term P, $se(P \leftand \mathsf{F})$ 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:*

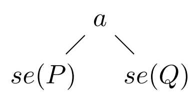
$$\begin{aligned}
P &::= P^{\mathsf{T}} \mid P^{\mathsf{F}} \mid P^{\mathsf{T}} \leftand P^* && (\textit{SNF-terms})\\
P^{\mathsf{T}} &::= \mathsf{T} \mid (a \leftand P^{\mathsf{T}}) \leftor P^{\mathsf{T}} && (\mathsf{T}\textit{-terms})\\
P^{\mathsf{F}} &::= \mathsf{F} \mid (a \leftor P^{\mathsf{F}}) \leftand P^{\mathsf{F}} && (\mathsf{F}\textit{-terms})\\
P^* &::= P^c \mid P^d && (*\textit{-terms})\\
P^c &::= P^\ell \mid P^* \leftand P^d\\
P^d &::= P^\ell \mid P^* \leftor P^c\\
P^\ell &::= (a \leftand P^{\mathsf{T}}) \leftor P^{\mathsf{F}} \mid (\neg a \leftand P^{\mathsf{T}}) \leftor P^{\mathsf{F}} && (\ell\textit{-terms})
\end{aligned}$$

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 $P^{\mathsf{T}} \leftand P^{*}$ 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 $\leftor$ is right-associative in T-terms, e.g.,

$$(a \leftand \mathsf{T}) \leftor ((b \leftand \mathsf{T}) \leftor \mathsf{T}) \quad \text{is a } \mathsf{T}\text{-term, but } ((a \leftand \mathsf{T}) \leftor (b \leftand \mathsf{T})) \leftor \mathsf{T} \text{ is not,}$$

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



Before we discuss the T-$*$-terms — the third type of our *SNF* normal forms — we consider the $*$-terms, which are $\leftand$-$\leftor$-combinations of ℓ-terms with the restriction that $\leftand$ and $\leftor$ 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,

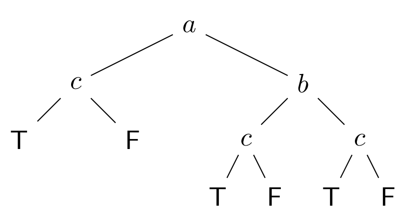
$$(P^{\ell} \leftand Q^{\ell}) \leftand R^{\ell}$$

is a $*$-term (it is in P^{c}-form), while $P^{\ell} \leftand (Q^{\ell} \leftand R^{\ell})$ 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

$$[a \leftor (b \leftor \mathsf{T})] \leftand c$$

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 $a \leftand (b \leftand \mathsf{F})$ and the $*$-term that encodes c, thus as

$$[(a \leftor \mathsf{F}) \leftand ((b \leftor \mathsf{F}) \leftand \mathsf{F})] \leftor [(c \leftand \mathsf{T}) \leftor \mathsf{F}].$$

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 normalization function

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

We will define f recursively using the functions

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

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 normalization function $f : \mathcal{S}_A \to \mathit{SNF}$ is defined recursively, using f^n and f^c, as follows.

$$f(a) = \mathsf{T} \leftand ((a \leftand \mathsf{T}) \leftor \mathsf{F}) \tag{2}$$
$$f(\mathsf{T}) = \mathsf{T} \tag{3}$$
$$f(\mathsf{F}) = \mathsf{F} \tag{4}$$
$$f(\neg P) = f^n(f(P)) \tag{5}$$
$$f(P \leftand Q) = f^c(f(P), f(Q)) \tag{6}$$
$$f(P \leftor Q) = f^n(f^c(f^n(f(P)), f^n(f(Q)))). \tag{7}$$

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. Analyzing 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 $f^n : \mathit{SNF} \to \mathit{SNF}$ 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.

$$f^n(\mathsf{T}) = \mathsf{F} \tag{8}$$
$$f^n((a \leftand P^\mathsf{T}) \leftor Q^\mathsf{T}) = (a \leftor f^n(Q^\mathsf{T})) \leftand f^n(P^\mathsf{T}) \tag{9}$$

$$f^n(\mathsf{F}) = \mathsf{T} \tag{10}$$
$$f^n((a \leftor P^\mathsf{F}) \leftand Q^\mathsf{F}) = (a \leftand f^n(Q^\mathsf{F})) \leftor f^n(P^\mathsf{F}) \tag{11}$$

$$f^n(P^\mathsf{T} \leftand Q^*) = P^\mathsf{T} \leftand f^n_1(Q^*) \tag{12}$$

$$f^n_1((a \leftand P^\mathsf{T}) \leftor Q^\mathsf{F}) = (\neg a \leftand f^n(Q^\mathsf{F})) \leftor f^n(P^\mathsf{T}) \tag{13}$$
$$f^n_1((\neg a \leftand P^\mathsf{T}) \leftor Q^\mathsf{F}) = (a \leftand f^n(Q^\mathsf{F})) \leftor f^n(P^\mathsf{T}) \tag{14}$$
$$f^n_1(P^* \leftand Q^d) = f^n_1(P^*) \leftor f^n_1(Q^d) \tag{15}$$
$$f^n_1(P^* \leftor Q^c) = f^n_1(P^*) \leftand f^n_1(Q^c). \tag{16}$$

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. $f^c(P^{\mathsf{T}} \leftand P^*, Q)$

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.

$$f^c(\mathsf{T}, P) = P \tag{17}$$
$$f^c((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}, R^{\mathsf{T}}) = (a \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \leftor f^c(Q^{\mathsf{T}}, R^{\mathsf{T}}) \tag{18}$$
$$f^c((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}, R^{\mathsf{F}}) = (a \leftor f^c(Q^{\mathsf{T}}, R^{\mathsf{F}})) \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{F}}) \tag{19}$$
$$f^c((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}, R^{\mathsf{T}} \leftand S^*) = f^c((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}, R^{\mathsf{T}}) \leftand S^*. \tag{20}$$

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.

$$f^c(P^{\mathsf{F}}, Q) = P^{\mathsf{F}} \tag{21}$$

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

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 $f^c_1 : P^* \times P^{\mathsf{T}} \to P^*$ to turn conjunctions of a $*$-term with a T-term into $*$-terms. We define f^c_1 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^c_1 are indeed $*$-terms.

$$f^c(P^{\mathsf{T}} \leftand Q^*, R^{\mathsf{T}}) = P^{\mathsf{T}} \leftand f^c_1(Q^*, R^{\mathsf{T}}) \tag{22}$$

$$f^c_1((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{T}}) = (a \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \leftor Q^{\mathsf{F}} \tag{23}$$
$$f^c_1((\neg a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{T}}) = (\neg a \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \leftor Q^{\mathsf{F}} \tag{24}$$
$$f^c_1(P^* \leftand Q^d, R^{\mathsf{T}}) = P^* \leftand f^c_1(Q^d, R^{\mathsf{T}}) \tag{25}$$
$$f^c_1(P^* \leftor Q^c, R^{\mathsf{T}}) = f^c_1(P^*, R^{\mathsf{T}}) \leftor f^c_1(Q^c, R^{\mathsf{T}}). \tag{26}$$

For case 3.2 we need to define $f^c(P^{\mathsf{T}} \leftand Q^*, R^{\mathsf{F}})$, 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 $f^c_2 : P^* \times P^{\mathsf{F}} \to P^{\mathsf{F}}$ that we define recursively by a case distinction on its first argument. Observe that the right-hand sides of the clauses defining f^c_2 are all F-terms.

$$f^c(P^{\mathsf{T}} \leftand Q^*, R^{\mathsf{F}}) = f^c(P^{\mathsf{T}}, f^c_2(Q^*, R^{\mathsf{F}})) \tag{27}$$

$$f^c_2((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{F}}) = (a \leftor Q^{\mathsf{F}}) \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{F}}) \tag{28}$$
$$f^c_2((\neg a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{F}}) = (a \leftor f^c(P^{\mathsf{T}}, R^{\mathsf{F}})) \leftand Q^{\mathsf{F}} \tag{29}$$
$$f^c_2(P^* \leftand Q^d, R^{\mathsf{F}}) = f^c_2(P^*, f^c_2(Q^d, R^{\mathsf{F}})) \tag{30}$$
$$f^c_2(P^* \leftor Q^c, R^{\mathsf{F}}) = f^c_2(f^n(f^c_1(P^*, f^n(R^{\mathsf{F}}))), f^c_2(Q^c, R^{\mathsf{F}})). \tag{31}$$

For case 3.3 we need to define $f^c(P^{\mathsf{T}} \leftand Q^*, R^{\mathsf{T}} \leftand S^*)$. We use the auxiliary function $f_3^c : P^* \times (P^{\mathsf{T}} \leftand P^*) \to P^*$ 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.

$$f^c(P^{\mathsf{T}} \leftand Q^*, R^{\mathsf{T}} \leftand S^*) = P^{\mathsf{T}} \leftand f_3^c(Q^*, R^{\mathsf{T}} \leftand S^*) \tag{32}$$

$$f_3^c(P^*, Q^{\mathsf{T}} \leftand R^{\ell}) = f_1^c(P^*, Q^{\mathsf{T}}) \leftand R^{\ell} \tag{33}$$

$$f_3^c(P^*, Q^{\mathsf{T}} \leftand (R^* \leftand S^d)) = f_3^c(P^*, Q^{\mathsf{T}} \leftand R^*) \leftand S^d \tag{34}$$

$$f_3^c(P^*, Q^{\mathsf{T}} \leftand (R^* \leftor S^c)) = f_1^c(P^*, Q^{\mathsf{T}}) \leftand (R^* \leftor S^c). \tag{35}$$

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

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

In Appendix A.2 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 normalization 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 analyze 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 $se(P \leftand Q)$ is assembled from $se(P)$ and $se(Q)$ and similarly for $se(P \leftor Q)$. 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 $se(P \leftand Q)$ 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

$$\triangle[\triangle \mapsto X]. \text{ [2]}$$

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

[2] 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.

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*.*

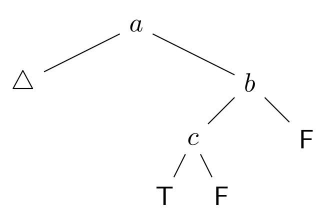
Proof. By induction on the structure of P. A proof of the first two statements is trivial. For the third statement, if P is an ℓ-term, we find that by definition of the grammar of P that one branch from the root of $se(P)$ will only contain T and not F, and for the other branch this is the other way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

For the induction we have to consider both $se(P_1 \leftand P_2)$ and $se(P_1 \leftor P_2)$. Consider $se(P_1 \leftand P_2)$, 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 $se(P_1 \leftand P_2)$ contains both T and F. The case $se(P_1 \leftor P_2)$ can be dealt with in a similar way. □

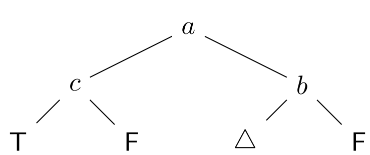
Decompositions of the se-image of $*$-terms turn out to be crucial in our approach. As an example, the se-image of the $*$-term

$$(P^\ell \leftor Q^\ell) \leftand R^\ell \quad \text{with} \quad P^\ell = ((a \leftand \mathsf{T}) \leftor \mathsf{F}),\ Q^\ell = ((b \leftand \mathsf{T}) \leftor \mathsf{F}),\ R^\ell = ((c \leftand \mathsf{T}) \leftor \mathsf{F})$$

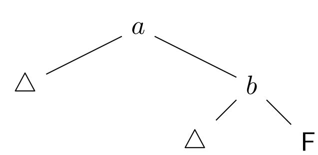
can be decomposed as $X_1[\triangle \mapsto Y]$ with $X_1 \in \mathcal{T}_{A,\triangle}$ as follows:



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:



and



Observe that the first two decompositions have the property that Y is a subtree of X_1 and X_2, respectively. Furthermore, observe that $X_3 = se(P^\ell \leftor Q^\ell)[\mathsf{T} \mapsto \triangle]$, 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ and $se(P \mathbin{{}^{\circ}\!\vee} Q)$.

We start with $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $\mathbin{{}_{\circ}\!\wedge}$ we have

$$se(P \mathbin{{}_{\circ}\!\wedge} Q) = se(P)[\mathsf{T} \mapsto se(Q)]$$

we also have

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

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 $P \mathbin{{}_{\circ}\!\wedge} Q$.

The case for $se(P \mathbin{{}^{\circ}\!\vee} Q)$ 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 analyze 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 $P \mathbin{{}_{\circ}\!\wedge} Q$ and $P \mathbin{{}^{\circ}\!\vee} Q$. We start with the case for $se(P \mathbin{{}_{\circ}\!\wedge} Q)$. Towards a contradiction, suppose that for some $*$-terms P and Q,

$$se(P \mathbin{{}_{\circ}\!\wedge} Q) = X[\triangle \mapsto Y] \tag{36}$$

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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ are subtrees in Y after being substituted in X:

- Because X does not contain T and F, Lemma 3.1.1 and (36) 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

$$se(Q) \text{ is a } \textit{proper} \text{ subtree of } Y. \tag{37}$$

This implies that *each* occurrence of $se(Q)$ in $se(P \mathbin{{}^{\circ}\!\wedge} Q)$ 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 (37) this implies that $se(Q)$ is a proper subtree of itself, which is a contradiction.

Because each occurrence of $se(Q)$ in $se(P \mathbin{{}^{\circ}\!\wedge} Q) = X[\triangle \mapsto Y]$ is an occurrence in Y (after being substituted) and because $se(P \mathbin{{}^{\circ}\!\wedge} Q) = se(P)[\mathsf{T} \mapsto se(Q)]$, 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 $se(P \mathbin{{}^{\circ}\!\wedge} Q)$.

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

We now arrive at two crucial definitions concerning decompositions. When considering $*$-terms, we already know that $se(P \mathbin{{}^{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}^{\circ}\!\wedge} Q)$. We also ensure that $se(P \mathbin{{}^{\circ}\!\vee} Q)$ does not have a decomposition of that kind, so that we can distinguish $se(P \mathbin{{}^{\circ}\!\wedge} Q)$ from $se(P \mathbin{{}^{\circ}\!\vee} Q)$. Similarly, we need to define another kind of decomposition so that $se(P \mathbin{{}^{\circ}\!\vee} Q)$ can only be decomposed as

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

and that $se(P \mathbin{{}^{\circ}\!\wedge} Q)$ 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 $P \mathbin{{}_{\circ}\!\wedge} Q$, $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ has no cdd. Similarly, for any $*$-term $P \mathbin{{}^{\circ}\!\vee} Q$, $se(P \mathbin{{}^{\circ}\!\vee} Q)$ has no ccd.*

Proof. We first treat the case for $P \mathbin{{}_{\circ}\!\wedge} Q$, so $P \in P^*$ and $Q \in P^d$. Towards a contradiction, suppose that (Y, Z) is a cdd of $se(P \mathbin{{}_{\circ}\!\wedge} Q)$. Let Z' be the witness of Lemma 3.1.2 for P. Observe that $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$. A proof for the case $se(P \mathbin{{}^{\circ}\!\vee} Q)$ is symmetric. $\square$

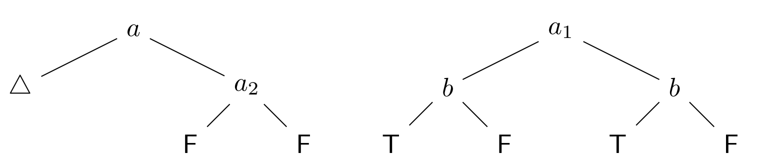
However, the ccd and cdd are not necessarily the decompositions we are looking for, because, for example, $se((P \mathbin{{}_{\circ}\!\wedge} Q) \mathbin{{}_{\circ}\!\wedge} R)$ has a ccd

$$(se(P)[\mathsf{T} \mapsto \triangle], se(Q \mathbin{{}_{\circ}\!\wedge} R)),$$

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

$$(se(P \mathbin{{}_{\circ}\!\wedge} Q)[\mathsf{T} \mapsto \triangle], se(R)).$$

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 $se(P^\ell \mathbin{{}_{\circ}\!\wedge} Q^\ell)$ with P^ℓ and Q^ℓ these ℓ-terms:

$$P^\ell = (a \mathbin{{}_{\circ}\!\wedge} ((a_1 \mathbin{{}_{\circ}\!\wedge} \mathsf{T}) \mathbin{{}^{\circ}\!\vee} \mathsf{T})) \mathbin{{}^{\circ}\!\vee} ((a_2 \mathbin{{}^{\circ}\!\vee} \mathsf{F}) \mathbin{{}_{\circ}\!\wedge} \mathsf{F}), \quad Q^\ell = (b \mathbin{{}_{\circ}\!\wedge} \mathsf{T}) \mathbin{{}^{\circ}\!\vee} \mathsf{F}.$$

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* $P \mathbin{{}^{\circ}\!\!\wedge} Q$*, i.e., with* $P \in P^*$ *and* $Q \in P^d$*,* $se(P \mathbin{{}^{\circ}\!\!\wedge} Q)$ *has the unique cd*

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

and no dd. For any $*$*-term* $P \mathbin{{}^{\circ}\!\!\vee} Q$*, i.e., with* $P \in P^*$ *and* $Q \in P^c$*,* $se(P \mathbin{{}^{\circ}\!\!\vee} Q)$ *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 $P \mathbin{{}^{\circ}\!\!\wedge} Q$ and $P \mathbin{{}^{\circ}\!\!\vee} Q$.

In the basis we have to consider, for ℓ-terms P^ℓ and Q^ℓ, the terms $P^\ell \mathbin{{}^{\circ}\!\!\wedge} Q^\ell$ and $P^\ell \mathbin{{}^{\circ}\!\!\vee} Q^\ell$. 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 $se(P^\ell \mathbin{{}^{\circ}\!\!\wedge} Q^\ell)$. Furthermore observe that the smallest subtree in $se(P^\ell \mathbin{{}^{\circ}\!\!\wedge} Q^\ell)$ 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 $se(P^\ell \mathbin{{}^{\circ}\!\!\wedge} Q^\ell)$. Now for the dd. It suffices to show that there is no cdd of $se(P^\ell \mathbin{{}^{\circ}\!\!\wedge} Q^\ell)$ 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 $P \mathbin{{}^{\circ}\!\!\wedge} Q$ and $P \mathbin{{}^{\circ}\!\!\vee} Q$. We will first treat the case for $P \mathbin{{}^{\circ}\!\!\wedge} Q$ and show that $(se(P)[\mathsf{T} \mapsto \triangle], se(Q))$ is the unique cd of $se(P \mathbin{{}^{\circ}\!\!\wedge} Q)$. 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 $Y[\triangle \mapsto Z] = se(P \mathbin{{}^{\circ}\!\!\wedge} Q)$ 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 $se(P \mathbin{{}^{\circ}\!\!\wedge} Q)$.
- 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

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

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 $se(P \leftand Q)$.

– 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 $se(P \leftand Q)$.

Now for the dd. By Lemma 3.1.5 there is no cdd of $se(P \leftand Q)$, so there is neither a dd of $se(P \leftand Q)$. A proof for the case $se(P \leftor Q)$ 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 $se(P^\mathsf{T} \leftand Q^*)$. To this end we define a T-$*$-decomposition, and as with cd's and dd's we first define a candidate T-$*$-decomposition.

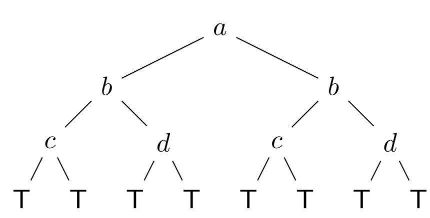
Definition 3.1.8. *The pair $(Y, Z) \in \mathcal{T}_{A,\triangle} \times \mathcal{T}_A$ is a **candidate** T-$*$**-decomposition (ctsd)** of $X \in \mathcal{T}_A$, if*

- $X = Y[\triangle \mapsto Z]$,
- *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 $se(P^\mathsf{T} \leftand Q^*)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P)[\mathsf{T} \mapsto se(Q)] = se(P \mathbin{{}_{\circ}\!\wedge} Q)$ 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 $se(P \mathbin{{}_{\circ}\!\wedge} Q)$. □

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.

$$g^{\mathsf{T}}(X) = \begin{cases} \mathsf{T} & \text{if } X = \mathsf{T}, \\ (a \mathbin{{}_{\circ}\!\wedge} g^{\mathsf{T}}(Y)) \mathbin{{}^{\circ}\!\vee} g^{\mathsf{T}}(Z) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z. \end{cases} \tag{38}$$

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

$$g^{\ell}(X) = \begin{cases} (a \mathbin{{}_{\circ}\!\wedge} g^{\mathsf{T}}(Y)) \mathbin{{}^{\circ}\!\vee} g^{\mathsf{F}}(Z) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z \text{ for some } a \in A \\ & \text{and } Y \text{ only has } \mathsf{T}\text{-leaves}, \\ (\neg a \mathbin{{}_{\circ}\!\wedge} g^{\mathsf{T}}(Z)) \mathbin{{}^{\circ}\!\vee} g^{\mathsf{F}}(Y) & \text{if } X = Y \trianglelefteq a \trianglerighteq Z \text{ for some } a \in A \\ & \text{and } Z \text{ only has } \mathsf{T}\text{-leaves}. \end{cases} \tag{40}$$

$$g^*(X) = \begin{cases} g^*(cd_1(X)[\triangle \mapsto \mathsf{T}]) \leftand g^*(cd_2(X)) & \text{if } X \text{ has a cd,} \\ g^*(dd_1(X)[\triangle \mapsto \mathsf{F}]) \leftor g^*(dd_2(X)) & \text{if } X \text{ has a dd,} \\ g^\ell(X) & \text{otherwise.} \end{cases} \tag{41}$$

$$g(X) = \begin{cases} g^{\mathsf{T}}(X) & \text{if } X \text{ has only } \mathsf{T}\text{-leaves,} \\ g^{\mathsf{F}}(X) & \text{if } X \text{ has only } \mathsf{F}\text{-leaves,} \\ g^{\mathsf{T}}(tsd_1(X)[\triangle \mapsto \mathsf{T}]) \leftand g^*(tsd_2(X)) & \text{otherwise.} \end{cases} \tag{42}$$

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 (38) that $g^{\mathsf{T}}(se(P)) \equiv g^{\mathsf{T}}(\mathsf{T}) \equiv \mathsf{T} \equiv P$. For the inductive case we have $P \equiv (a \leftand Q^{\mathsf{T}}) \leftor R^{\mathsf{T}}$ and

$$\begin{aligned} g^{\mathsf{T}}(se(P)) &\equiv g^{\mathsf{T}}(se(Q^{\mathsf{T}}) \trianglelefteq a \trianglerighteq se(R^{\mathsf{T}})) && \text{by definition of } se \\ &\equiv (a \leftand g^{\mathsf{T}}(se(Q^{\mathsf{T}}))) \leftor g^{\mathsf{T}}(se(R^{\mathsf{T}})) && \text{by (38)} \\ &\equiv (a \leftand Q^{\mathsf{T}}) \leftor R^{\mathsf{T}} && \text{by induction hypothesis} \\ &\equiv P. \end{aligned}$$

In a similar way it follows by (39) 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 $P \equiv (a \leftand Q^{\mathsf{T}}) \leftor R^{\mathsf{F}}$ or $P \equiv (\neg a \leftand Q^{\mathsf{T}}) \leftor R^{\mathsf{F}}$. In the first case we have

$$\begin{aligned} g^\ell(se(P)) &\equiv g^\ell(se(Q^{\mathsf{T}}) \trianglelefteq a \trianglerighteq se(R^{\mathsf{F}})) && \text{by definition of } se \\ &\equiv (a \leftand g^{\mathsf{T}}(se(Q^{\mathsf{T}}))) \leftor g^{\mathsf{F}}(se(R^{\mathsf{F}})) && \text{by (40), first case} \\ &\equiv (a \leftand Q^{\mathsf{T}}) \leftor R^{\mathsf{F}} && \text{as shown above} \\ &\equiv P. \end{aligned}$$

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 (41) and the second identity was shown above. For the induction we have either $P \equiv Q \leftand R$ or $P \equiv Q \leftor R$. In the first case note that by Theorem 3.1.7, $se(P)$ has a unique cd and no dd. So we have

$$\begin{aligned} g^*(se(P)) &\equiv g^*(cd_1(se(P))[\triangle \mapsto \mathsf{T}]) \leftand g_*(cd_2(se(P))) && \text{by (41)} \\ &\equiv g^*(se(Q)) \leftand g^*(se(R)) && \text{by Theorem 3.1.7} \\ &\equiv Q \leftand R && \text{by induction hypothesis} \\ &\equiv P. \end{aligned}$$

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

$$\begin{aligned} g^*(se(P)) &\equiv g^*(dd_1(se(P))[\triangle \mapsto \mathsf{F}]) \leftor g_*(dd_2(se(P))) && \text{by (41)} \\ &\equiv g^*(se(Q)) \leftor g^*(se(R)) && \text{by Theorem 3.1.7} \\ &\equiv Q \leftor R && \text{by induction hypothesis} \\ &\equiv P. \end{aligned}$$

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 (42) 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 (42) 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 $P \equiv Q \wedge R$,

$$\begin{aligned}
g(se(P)) &\equiv g^{\mathsf{T}}(tsd_1(se(P))[\triangle \mapsto \mathsf{T}]) \wedge g^*(tsd_2(se(P))) && \text{by (42)}\\
&\equiv g^{\mathsf{T}}(se(Q)) \wedge g^*(se(R)) && \text{by Theorem 3.1.10}\\
&\equiv Q \wedge R && \text{as shown above}\\
&\equiv P,
\end{aligned}$$

which completes the proof. □

Theorem 3.2.2 (Completeness)**.** *For all* $P, Q \in \mathcal{S}_A$*, if* FSCL $\models P = Q$ *then* EqFSCL $\vdash P = Q$*.*

Proof. Suppose FSCL $\models P = Q$, 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 2.1.5, $se(P) = se(P')$ 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 Evaluation strategies and side effects

Starting from short-circuit evaluation, we consider various evaluation strategies leading to SCLs that identify more sequential propositional statements than FSCL does (Section 4.1). Then we briefly discuss *side effects* (Section 4.2).

4.1 Evaluation strategies

In the case of free short-circuit logic, we assume that atomic evaluations are not memorized during the sequential evaluation of a compound propositional statement. In earlier work we also defined some variants of short-circuit logic in which atomic evaluations are memorized up to a certain extent, with sequential propositional logic as the extreme case: in the sequential evaluation of a compound statement, all atomic evaluations are memorized throughout the evaluation. Before explaining the notion of an evaluation strategy in more detail, we give an alternative characterization of the connectives $\wedge$ and $\vee$.

In 1985, Hoare [12] introduced the *conditional*, a ternary connective with notation

$$x \triangleleft y \triangleright z.$$

A more common expression for the conditional $x \triangleleft y \triangleright z$ is

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

which emphasizes 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

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

Table 2: The set CP of axioms for proposition algebra

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 $x \triangleleft y \triangleright z$ seems indispensable. In [12], Hoare provides an equational axiomatization of propositional logic that only uses the conditional and comments 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 [12]). This axiomatization consists of eleven axioms and includes the four axioms in Table 2, and some more axioms, for example

$$x \triangleleft y \triangleright (z \triangleleft y \triangleright w) = x \triangleleft y \triangleright w.$$

The extension of the definition of evaluation trees for the interpretation of the conditional is as expected:

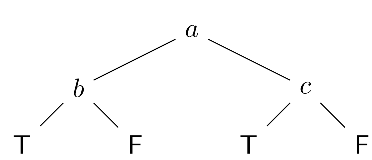
$$se(P \triangleleft Q \triangleright R) = se(Q)[\mathsf{T} \mapsto se(P), \mathsf{F} \mapsto se(R)].$$

The four axioms in Table 2, referred to as CP (for Conditional Propositions), establish a complete axiomatization for *free valuation congruence*, that is, for equality of evaluation trees. A simple and concise proof of this fact is recorded in [4].

With the conditional connective and the constants T and F, the sequential connectives and negation are definable:

$$x \mathbin{{}_{\circ}\!\wedge} y = y \triangleleft x \triangleright \mathsf{F}, \tag{43}$$
$$x \mathbin{{}^{\circ}\!\vee} y = \mathsf{T} \triangleleft x \triangleright y, \tag{44}$$
$$\neg x = \mathsf{F} \triangleleft x \triangleright \mathsf{T}. \tag{45}$$

Thus, the axioms in Table 2 plus the defining equations above, say CP($\mathbin{{}_{\circ}\!\wedge}, \mathbin{{}^{\circ}\!\vee}, \neg$), axiomatize equality of evaluation trees for expressions over this enriched signature. This does not provide clear intuitions about the evaluation trees that can be expressed *without* use of the conditional, although it is relatively easy to show that a simple evaluation tree like $se(b \triangleleft a \triangleright c)$, that is



cannot be expressed in FSCL, thus as the se-image of an $\mathcal{S}_A$-term. So, although one can check easily for two $\mathcal{S}_A$-terms P and Q whether FSCL $\models P = Q$, this approach does not provide insight as to which expressions are identified. Clearly, the axiomatization of FSCL in Table 1 does provide such insight.

In [2] a set-up is provided for defining short-circuit logics with help of the conditional by restricting the enriched language of $\mathrm{CP}(\leftand, \leftor, \neg)$ to the signature of $\mathcal{S}_A$. The conditional connective is declared as a *hidden operator*.[3] In this set-up, variants of FSCL that identify more sequential propositions can be easily defined. As an example, adding to $\mathrm{CP}(\leftand, \leftor, \neg)$ the two equation schemes

$$(x \lhd a \rhd y) \lhd a \rhd z = (x \lhd a \rhd x) \lhd a \rhd z \quad \text{and} \quad x \lhd a \rhd (y \lhd a \rhd z) = x \lhd a \rhd (z \lhd a \rhd z) \tag{46}$$

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,

$$\mathrm{RPSCL} \models a \leftand (a \leftor x) = a \leftand a.$$

For RPSCL there exist natural examples (below, we sketch one briefly), and furthermore it is proven in [6] that RPSCL has no finite equational axiomatization without hidden operations. Evaluation trees for RPSCL are defined by a transformation of *se*-trees according to the axiom schemes (46), see [4, 6]. For another example, adding to $\mathrm{CP}(\leftand, \leftor, \neg)$ the two axioms

$$x \lhd y \rhd (z \lhd u \rhd (v \lhd y \rhd w)) = x \lhd y \rhd (z \lhd u \rhd w) \quad \text{and} \quad \mathsf{F} \lhd x \rhd \mathsf{F} = \mathsf{F} \tag{47}$$

defines *static* SCL (see [2]) when restricted to the signature of $\mathcal{S}_A$, which is a sequential form of propositional logic. Note that the first axiom and those of CP (in Table 2) imply the axioms schemes (46).

Another evaluation strategy. 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 $x \fulland y$ and $x \fullor y$ 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 axiomatization is provided in [17]. This axiomatization also comprises axioms (F1) and (F3), and a typical axiom is

$$x \fulland \mathsf{F} = \mathsf{F} \fulland x. \tag{48}$$

With the tool *Prover9* [13] it follows that (F1) is derivable, and with the tool *Mace4* [13] it follows that the remaining axioms in [17] are independent (even if $|A| = 1$). Furthermore, both (F1) and (F3) become derivable if the axiom $x \fulland \mathsf{T} = x$ is replaced by $x \fullor \mathsf{F} = x$, and the remaining axioms are again independent (even if $|A| = 1$).

As is also noted in [17], the 'full sequential connectives' can be defined in terms of $\leftand$ and $\leftor$, and the constants T and F:

$$x \fulland y = (x \leftor (y \leftand \mathsf{F})) \leftand y \quad \text{and} \quad x \fullor y = (x \leftand (y \leftor \mathsf{T})) \leftor y.$$

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 (48) in EqFSCL.

[3]In fact, the defining equation (44) can be omitted, because the translation of axiom (F2) can be derived in $\mathrm{CP}(\leftand, \leftor, \neg)$ (this is a simple exercise).

FSCL without T and F. 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 axiomatization of this restricted form of free valuation 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.

4.2 Side effects

Although side effects seem to be well understood in programming, see e.g., [7, 8, 15, 14], 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 `(1 ~= 1) && (...)` 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 [4, Ex.7.1]:

> Assume atoms are of the form (e = e') and (v := e) with v some program variable and e, e' arithmetical expressions over the integers that may contain v. Furthermore, assume that (e = e') evaluates to *true* if and only if e and e' represent the same value, and (v := e) always evaluates to *true* with the effect that the value of e is assigned to v. Then these atoms satisfy the axioms of RPSCL, that is CP + (43)-(46).[4] Furthermore, if v has initial value 0 or 1, the conditions
>
> $$(\texttt{(v := v+1)} \mathbin{{}^{\circ}\!\wedge} \texttt{(v := v+1)}) \mathbin{{}^{\circ}\!\wedge} \texttt{(v = 2)} \quad \text{and} \quad \texttt{(v := v+1)} \mathbin{{}^{\circ}\!\wedge} \texttt{(v = 2)}$$
>
> evaluate to different results. Next, observe that for all initial values of v and for all $P \in \mathcal{S}_A$,
>
> $$\text{RPSCL} \models \texttt{(v := v+1)} \mathbin{{}^{\circ}\!\wedge} (\texttt{(v := v+1)} \mathbin{{}^{\circ}\!\vee} P) = \texttt{(v := v+1)} \mathbin{{}^{\circ}\!\wedge} \texttt{(v := v+1)}. \quad (49)$$

We note that the set-up of our toy programming language suggests a sequential variant of *Dynamic Logic* (see, e.g., [11]) in which assignments can be used both as tests and as programs. Such a sequential variant could be appropriate for reasoning about side effects. However, in an expression-evaluated language, an atom (v := e) evaluates to the Boolean value of e, and hence FSCL would be the appropriate SCL. For example, if we consider the condition in (49) as one in an expression-evaluated programming language such as Perl, we find that

$$\text{FSCL} \not\models \texttt{(v := v+1)} \mathbin{{}^{\circ}\!\wedge} (\texttt{(v := v+1)} \mathbin{{}^{\circ}\!\vee} \mathsf{T}) = \texttt{(v := v+1)} \mathbin{{}^{\circ}\!\wedge} \texttt{(v := v+1)}$$

(set the initial value of v to -2 and recall that 0 evaluates to *false*).

5 Conclusion

In this paper we discuss *free short-circuit logic* (FSCL), following earlier research reported on in [5, 17, 3]. In FSCL, intermediate evaluation results are not memorized 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 [1] for four-valued logic and named (left first) sequential conjunction and sequential disjunction, provide a convenient solution in this case.

A last comment on the ten equational axioms that we selected for our axiomatization of FSCL (in [5, 17, 3] 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, equational axioms seem to grasp the nature of FSCL-identities in a more direct way, and each of these axioms

[4]Of course, not all equations that are valid in this setting follow from CP + (43)-(46), e.g., $\not\vdash$ (1 = 1) = T.

embodies a simple idea. This is in particular the case for (F1) and (F3), and showing that these two axioms can be derived from the remaining ones might be part of a suitable introduction to the nature of FSCL. We note that Lemma 2.1.4 was checked with the tool *Prover9* [13].

When it comes to reasoning about side effects, we subscribe to Parnas' view [16]:

> *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.

Concerning future work, it remains a challenging question to find a more simple completeness proof for the axiomatization of FSCL (Theorem 3.2.2). Furthermore, we aim to provide elegant and independent equational axiomatizations for some other variants of SCL defined in [2], or proofs of their non-existence (without hidden operations [6]). And, last but not least, we aim to find fruitful applications for FSCL and the other SCLs we defined.

References

[1] 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.

[2] Bergstra, J.A. and Ponse, A. (2011). Proposition algebra. *ACM Transactions on Computational Logic*, Vol. 12, No. 3, Article 21 (36 pages).

[3] Bergstra, J.A. and Ponse, A. (2012). Proposition algebra and short-circuit logic. In F. Arbab and M. Sirjani (eds.), *Proceedings of the 4th International Conference on Fundamentals of Software Engineering (FSEN 2011)*, LNCS 7141, pages 15-31, Springer-Verlag.

[4] Bergstra, J.A. and Ponse, A. (2015). Evaluation trees for proposition algebra. Available at `http://arxiv.org/abs/1504.0832` [cs.LO].

[5] Bergstra, J.A., Ponse, A., and Staudt, D.J.C. (2013). Short-circuit logic. Available at `arXiv:1010.3674v4` [cs.LO,math.LO]. (First version appeared in 2010.)

[6] Bethke, I. and Ponse, A. (2017). Short-circuit logics without finite basis. In progress.

[7] Black, P.E. and Windley, P.J. (1996). Inference rules for programming languages with side effects in expressions. In: J. von Wright, J. Grundy and J. Harrison (eds.), *Theorem Proving in Higher Order Logics: 9th International Conference*, pages 51-60. Springer-Verlag.

[8] Black, P.E. and Windley, P.J. (1998). Formal verification of secure programs in the presence of side effects. In Jr. R. H. Sprague (ed.), *Proceedings of the Thirty-first Hawaii International Conference on System Sciences (HICSS-31)*, Volume III, pages 327-334. IEEE Computer Science Press, 1998.

[9] Dijkstra, E.W. (1976). *A Discipline of Programming.* Prentice Hall, Inc.

[10] Gries, D. (1981). *The Science of Programming.* Springer-Verlag.

[11] Harel, D. (1984). Dynamic Logic. In: D. Gabbay and F. Günthner (eds.), *Handbook of Philosophical Logic*, Volume II, pages 497-604.

[12] 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.

[13] McCune, W.W. (Accessed June 2017). *Son of BirdBrain II.* Web interface for the tools *Prover9* and *Mace4*, `http://www.cs.unm.edu/~mccune/sobb2/`.

[14] Kneuss, E., Kuncak, V., and Suter, Ph. (2014). Effect analysis for programs with callbacks. In: Cohen, E. and Rybalchenko, A. (eds), *VSTTE 2013: Verified Software: Theories, Tools, Experiments*, LNCS 8164, Springer, 48-67.

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

[16] Parnas, D.L. (2010). Really Rethinking 'Formal Methods'. *Computer*, 43(1):28-34, IEEE Computer Society.

[17] Staudt, D.J.C. (2012). Completeness for Two Left-Sequential Logics. MSc. thesis Logic, University of Amsterdam (May 2012). Available at `arXiv:1206.1936v1` [cs.LO].

A Appendix

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 the tool *Mace4*, accessible through the web interface *Son of BirdBrain II* [13]. The independence of axiom (F10) was shown in the proof of Theorem 2.1.8.

First we prove the independence of axiom (F2) by providing a model for $\mathrm{EqFSCL}^- \setminus \{(\mathrm{F2})\}$ that refutes the (F2)-instance

$$\mathsf{F} \mathbin{\overset{\circ}{\vee}} \mathsf{F} = \neg(\neg\mathsf{F} \mathbin{\underset{\circ}{\wedge}} \neg\mathsf{F}).$$

We briefly describe how we used [13] to obtain this model. Having chosen "Boolean Algebra" as the area in `https://www.cs.unm.edu/~mccune/sobb2/`, the input uploaded in the box "Additional Hypotheses" is the following (representing $\mathrm{EqFSCL}^- \setminus \{(\mathrm{F2})\}$):

```
1 ^ x = x.
x v 0 = x.
0 ^ x = 0.
(x ^ y) ^ z = x ^ (y ^ z).
x' ^ 0 = x ^ 0.
(x ^ 0) v y = (x v 1) ^ y.
(x ^ y) v (z ^ 0) = (x v (z ^ 0)) ^ (y v (z ^ 0)).
```

So, the constants T and F are represented by 1 and 0, left-sequential conjunction and disjunction are represented by `^` and `v`, and negation is represented by a postfix function `x'`. Axiom (F2), represented by the clause "`x v y = (x' ^ y')'.`", is uploaded in the box "Alternate Conclusion".

Below we copy the complete output of *Mace4*, obtained after hitting the button "OK, prove it!". In the model obtained, the number 2 in

`interpretation( 2, ...`

defines its domain as the set of natural numbers smaller than 2. The constants 1 and 0 (thus, T and F) are interpreted as the same values. The representation of the truth tables is self-explanatory. Note that in this model two constants c1 and c2 are used that both are interpreted as 0, so we can set c1 = c2 = F (as we did above).

```
Searching for a counterexample ...

Success! The formulas

   formulas(mace4_clauses).
   1 ^ x = x.
   x v 0 = x.
   0 ^ x = 0.
   (x ^ y) ^ z = x ^ (y ^ z).
   x' ^ 0 = x ^ 0.
   (x ^ 0) v y = (x v 1) ^ y.
   (x ^ y) v (z ^ 0) = (x v (z ^ 0)) ^ (y v (z ^ 0)).
   (c1' ^ c2')' != c1 v c2.
   end_of_list.

have the following model.

    ============================== MODEL =================================

    interpretation( 2, [number=1, seconds=0], [

            function(c1, [ 0 ]),

            function(c2, [ 0 ]),

            function('(_), [ 1, 1 ]),

            function(^(_,_), [
        0, 0,
        0, 1 ]),

            function(v(_,_), [
        0, 0,
        1, 0 ])
    ]).

    ============================== end of model ==========================
```

Next, we show the independence of each of the axioms (F4)-(F9) in EqFSCL$^-$. For all models defined below, we assume an interpretation function ϕ with $\phi(\mathsf{F}) = 0$ and $\phi(\mathsf{T}) = 1$. Furthermore, observe that all refutations below use at most one atom.

A model for EqFSCL$^- \setminus \{(\text{F4})\}$ with domain $\{0, 1\}$ that refutes $\mathsf{T} \wedge\!\!\!\circ\ \mathsf{F} = \mathsf{F}$ is the following:

$\neg$	
0	0
1	1

$\wedge\!\!\!\circ$	0	1
0	0	0
1	1	1

$\vee\!\!\!\circ$	0	1
0	0	0
1	1	1

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

$\neg$	
0	0
1	0

$\mathbin{{}^{\circ}\!\!\wedge}$	0	1
0	0	0
1	0	1

$\mathbin{{}^{\circ}\!\!\vee}$	0	1
0	0	0
1	0	0

A model for EqFSCL$^-$ $\setminus$ {(F6)} with domain $\{0,1,2\}$ that refutes $\mathsf{F} \mathbin{{}^{\circ}\!\!\wedge} a = \mathsf{F}$ in the case that $\phi(a) = 2$ is the following:

$\neg$	
0	1
1	0
2	2

$\mathbin{{}^{\circ}\!\!\wedge}$	0	1	2
0	0	0	2
1	0	1	2
2	0	2	2

$\mathbin{{}^{\circ}\!\!\vee}$	0	1	2
0	0	1	2
1	1	1	2
2	2	1	2

A model for EqFSCL$^-$ $\setminus$ {(F7)} with domain $\{0,1,2,3\}$ that refutes $(a \mathbin{{}^{\circ}\!\!\wedge} \mathsf{F}) \mathbin{{}^{\circ}\!\!\wedge} a = a \mathbin{{}^{\circ}\!\!\wedge} (\mathsf{F} \mathbin{{}^{\circ}\!\!\wedge} a)$ if $\phi(a) = 2$ is this one:

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

$\mathbin{{}^{\circ}\!\!\wedge}$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	3	2	0	3
3	3	3	2	3

$\mathbin{{}^{\circ}\!\!\vee}$	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 for EqFSCL$^-$ $\setminus$ {(F8)} with domain $\{0,1,2,3\}$ that refutes $\neg a \mathbin{{}^{\circ}\!\!\wedge} \mathsf{F} = a \mathbin{{}^{\circ}\!\!\wedge} \mathsf{F}$ if $\phi(a) = 2$ is the following:

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

$\mathbin{{}^{\circ}\!\!\wedge}$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	2	2	2	2
3	3	3	3	3

$\mathbin{{}^{\circ}\!\!\vee}$	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 for EqFSCL$^-$ $\setminus$ {(F9)} with domain $\{0,1,2,3,4\}$ that refutes $(a \mathbin{{}^{\circ}\!\!\wedge} \mathsf{F}) \mathbin{{}^{\circ}\!\!\vee} a = (a \mathbin{{}^{\circ}\!\!\vee} \mathsf{T}) \mathbin{{}^{\circ}\!\!\wedge} a$ if $\phi(a) = 2$ is this one:

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

$\mathbin{{}^{\circ}\!\!\wedge}$	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

$\mathbin{{}^{\circ}\!\!\vee}$	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 normalization function

In order to prove that $f : \mathcal{S}_A \to \mathit{SNF}$ is indeed a normalization function we need to prove that for all SCL-terms P, $f(P)$ terminates, $f(P) \in \mathit{SNF}$ 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 $\vdash P = P \mathbin{{}^{\circ}\!\!\wedge} x$ *and* EqFSCL $\vdash Q = Q \mathbin{{}^{\circ}\!\!\vee} x$*.*

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

For the induction we have $(a \mathbin{{}^{\circ}\!\!\vee} P_1) \mathbin{{}^{\circ}\!\!\wedge} P_2 = (a \mathbin{{}^{\circ}\!\!\vee} P_1) \mathbin{{}^{\circ}\!\!\wedge} (P_2 \mathbin{{}^{\circ}\!\!\wedge} x)$ 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 2.1.4 will prove useful in this appendix, as will the following equalities, which also deal with terms of the form $x \leftand \mathsf{F}$ and $x \leftor \mathsf{T}$.

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

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

Proof. We note that these equations were checked with the theorem prover *Prover9* [13].

1. $(x \leftor (y \leftand \mathsf{F})) \leftand (z \leftand \mathsf{F})$
   - $= (\neg x \leftor (z \leftand \mathsf{F})) \leftand ((y \leftand \mathsf{F}) \leftand (z \leftand \mathsf{F}))$ — by Lemma 2.1.4
   - $= (\neg x \leftor (z \leftand \mathsf{F})) \leftand (y \leftand \mathsf{F})$, — by (F6) and (F7)

2. $(x \leftand (y \leftor \mathsf{T})) \leftor (z \leftand \mathsf{F})$
   - $= (x \leftor (z \leftand \mathsf{F})) \leftand ((y \leftor \mathsf{T}) \leftor (z \leftand \mathsf{F}))$ — by (F10)
   - $= (x \leftor (z \leftand \mathsf{F})) \leftand (y \leftor \mathsf{T})$, — by (F6)′ and (F7)′

3. $(x \leftor \mathsf{T}) \leftand \neg y$
   - $= \neg((\neg x \leftand \mathsf{F}) \leftor y)$ — by (F2)
   - $= \neg((x \leftand \mathsf{F}) \leftor y)$ — by (F8)
   - $= \neg((x \leftor \mathsf{T}) \leftand y)$, — by (F9)

4. $(x \leftand (y \leftand (z \leftor \mathsf{T}))) \leftor (w \leftand (z \leftor \mathsf{T}))$
   - $= ((x \leftand y) \leftand (z \leftor \mathsf{T})) \leftor (w \leftand (z \leftor \mathsf{T}))$ — by (F7)
   - $= ((x \leftand y) \leftor w) \leftand (z \leftor \mathsf{T})$, — by (F10)′

5. $(x \leftor ((y \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F}))) \leftand ((w \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F}))$
   - $= (x \leftor ((y \leftand \mathsf{F}) \leftor (z \leftand \mathsf{F}))) \leftand ((w \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F}))$ — by (F9)
   - $= ((x \leftor (y \leftand \mathsf{F})) \leftor (z \leftand \mathsf{F})) \leftand ((w \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F}))$ — by (F7)′
   - $= (\neg(x \leftor (y \leftand \mathsf{F})) \leftor (w \leftor \mathsf{T})) \leftand (z \leftand \mathsf{F})$ — by Lemma 2.1.4
   - $= ((\neg x \leftand (\neg y \leftor \mathsf{T})) \leftor (w \leftor \mathsf{T})) \leftand (z \leftand \mathsf{F})$ — by (F2)′
   - $= ((\neg x \leftand (y \leftor \mathsf{T})) \leftor (w \leftor \mathsf{T})) \leftand (z \leftand \mathsf{F})$ — by (F8)′
   - $= ((\neg x \leftand (y \leftor \mathsf{T})) \leftor (w \leftor (\mathsf{T} \leftor (y \leftor \mathsf{T})))) \leftand (z \leftand \mathsf{F})$ — by (F6)′
   - $= ((\neg x \leftand (y \leftor \mathsf{T})) \leftor ((w \leftor \mathsf{T}) \leftor (y \leftor \mathsf{T}))) \leftand (z \leftand \mathsf{F})$ — by (F7)′
   - $= ((x \leftand (w \leftor \mathsf{T})) \leftor (y \leftor \mathsf{T})) \leftand (z \leftand \mathsf{F})$, — by Lemma 2.1.4′

6. $(x \leftor ((y \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F}))) \leftand (w \leftand \mathsf{F})$
   - $= (\neg x \leftor (w \leftand \mathsf{F})) \leftand (((y \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F})) \leftand (w \leftand \mathsf{F}))$ — by Lemma 2.1.4
   - $= (\neg x \leftor (w \leftand \mathsf{F})) \leftand ((y \leftor \mathsf{T}) \leftand (z \leftand \mathsf{F}))$ — by (F6) and (F7)
   - $= ((\neg x \leftor (w \leftand \mathsf{F})) \leftand (y \leftor \mathsf{T})) \leftand (z \leftand \mathsf{F})$ — by (F7)
   - $= ((\neg x \leftand (y \leftor \mathsf{T})) \leftor ((w \leftand \mathsf{F}) \leftand (y \leftor \mathsf{T}))) \leftand (z \leftand \mathsf{F})$ — by (F10)′
   - $= ((\neg x \leftand (y \leftor \mathsf{T})) \leftor (w \leftand \mathsf{F})) \leftand (z \leftand \mathsf{F})$. — by (F6) and (F7)

□

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*

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

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 (8), so $f^n(\mathsf{T})$ is an F-term. The claim that $\text{EqFSCL} \vdash f^n(\mathsf{T}) = \neg\mathsf{T}$ is immediate by (F1). For the inductive case we have that $f^n((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}) = (a \leftor f^n(Q^{\mathsf{T}})) \leftand f^n(P^{\mathsf{T}})$ by (9), where we assume that $f^n(P^{\mathsf{T}})$ and $f^n(Q^{\mathsf{T}})$ are F-terms and that $\text{EqFSCL} \vdash f^n(P^{\mathsf{T}}) = \neg P^{\mathsf{T}}$ and $\text{EqFSCL} \vdash f^n(Q^{\mathsf{T}}) = \neg Q^{\mathsf{T}}$. It follows from the induction hypothesis that $f^n((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}})$ 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:

$$\begin{aligned}
f^n((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}) &= (a \leftor f^n(Q^{\mathsf{T}})) \leftand f^n(P^{\mathsf{T}}) && \text{by (9)}\\
&= (a \leftor (f^n(Q^{\mathsf{T}}) \leftand \mathsf{F})) \leftand (f^n(P^{\mathsf{T}}) \leftand \mathsf{F}) && \text{by Lemma A.2.1}\\
&= (\neg a \leftor (f^n(P^{\mathsf{T}}) \leftand \mathsf{F})) \leftand (f^n(Q^{\mathsf{T}}) \leftand \mathsf{F}) && \text{by Lemma A.2.2.1}\\
&= (\neg a \leftor f^n(P^{\mathsf{T}})) \leftand f^n(Q^{\mathsf{T}}) && \text{by Lemma A.2.1}\\
&= (\neg a \leftor \neg P^{\mathsf{T}}) \leftand \neg Q^{\mathsf{T}} && \text{by induction hypothesis}\\
&= \neg((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}}). && \text{by (F2) and its dual}
\end{aligned}$$

For F-terms we prove our claims by induction on P^{F}. In the base case $f^n(\mathsf{F}) = \mathsf{T}$ by (10), so $f^n(\mathsf{F})$ is a T-term. The claim that $\text{EqFSCL} \vdash f^n(\mathsf{F}) = \neg\mathsf{F}$ is immediate by the dual of (F1). For the inductive case we have that $f^n((a \leftor P^{\mathsf{F}}) \leftand Q^{\mathsf{F}}) = (a \leftand f^n(Q^{\mathsf{F}})) \leftor f^n(P^{\mathsf{F}})$ by (11), where we assume that $f^n(P^{\mathsf{F}})$ and $f^n(Q^{\mathsf{F}})$ are T-terms and that $\text{EqFSCL} \vdash f^n(P^{\mathsf{F}}) = \neg P^{\mathsf{F}}$ and $\text{EqFSCL} \vdash f^n(Q^{\mathsf{F}}) = \neg Q^{\mathsf{F}}$. It follows from the induction hypothesis that $f^n((a \leftor P^{\mathsf{F}}) \leftand Q^{\mathsf{F}})$ 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 $f^n((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{T}})$.

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, $\text{EqFSCL} \vdash f_1^n(P) = \neg P$. 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 $(a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}$ we have:

$$\begin{aligned}
f_1^n((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}) &= (\neg a \leftand f^n(Q^{\mathsf{F}})) \leftor f^n(P^{\mathsf{T}}) && \text{by (13)}\\
&= (\neg a \leftand (f^n(Q^{\mathsf{F}}) \leftor \mathsf{T})) \leftor (f^n(P^{\mathsf{T}}) \leftand \mathsf{F}) && \text{by Lemma A.2.1}\\
&= (\neg a \leftor (f^n(P^{\mathsf{T}}) \leftand \mathsf{F})) \leftand (f^n(Q^{\mathsf{F}}) \leftor \mathsf{T}) && \text{by Lemma A.2.2.2}\\
&= (\neg a \leftor f^n(P^{\mathsf{T}})) \leftand f^n(Q^{\mathsf{F}}) && \text{by Lemma A.2.1}\\
&= (\neg a \leftor \neg P^{\mathsf{T}}) \leftand \neg Q^{\mathsf{F}} && \text{by induction hypothesis}\\
&= \neg((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}). && \text{by (F2) and its dual}
\end{aligned}$$

If P is an ℓ-term of the form $(\neg a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}$ the proof proceeds the same, substituting $\neg a$ for a and applying (14) and (F3) where needed. For the inductive step we assume that the result holds for all $*$-terms with fewer ℓ-terms than $P^* \leftand Q^d$ and $P^* \leftor Q^c$. By (15) and (16), 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 (the dual of) (F2).

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

$$\begin{aligned}
f^n(P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} Q^*) &= P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} f_1^n(Q^*) && \text{by (12)}\\
&= P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} \neg Q^* && \text{as shown above}\\
&= (P^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T}) \mathbin{{}^{\circ}\!\wedge} \neg Q^* && \text{by Lemma A.2.1}\\
&= \neg((P^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T}) \mathbin{{}^{\circ}\!\wedge} Q^*) && \text{by Lemma A.2.2.3}\\
&= \neg(P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} Q^*). && \text{by Lemma A.2.1}
\end{aligned}$$

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

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

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

Proof. By induction on the complexity of the T-term. In the base case we see that $f^c(\mathsf{T}, P) = P$ by (17), 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 $(a \mathbin{{}^{\circ}\!\wedge} P^{\mathsf{T}}) \mathbin{{}^{\circ}\!\vee} Q^{\mathsf{T}}$. 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:

$$\begin{aligned}
&f^c((a \mathbin{{}^{\circ}\!\wedge} P^{\mathsf{T}}) \mathbin{{}^{\circ}\!\vee} Q^{\mathsf{T}}, R^{\mathsf{T}})\\
&\quad= (a \mathbin{{}^{\circ}\!\wedge} f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \mathbin{{}^{\circ}\!\vee} f^c(Q^{\mathsf{T}}, R^{\mathsf{T}}) && \text{by (18)}\\
&\quad= (a \mathbin{{}^{\circ}\!\wedge} (P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} R^{\mathsf{T}})) \mathbin{{}^{\circ}\!\vee} (Q^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} R^{\mathsf{T}}) && \text{by induction hypothesis}\\
&\quad= (a \mathbin{{}^{\circ}\!\wedge} (P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} (R^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T}))) \mathbin{{}^{\circ}\!\vee} (Q^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} (R^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T})) && \text{by Lemma A.2.1}\\
&\quad= ((a \mathbin{{}^{\circ}\!\wedge} P^{\mathsf{T}}) \mathbin{{}^{\circ}\!\vee} Q^{\mathsf{T}}) \mathbin{{}^{\circ}\!\wedge} (R^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T}) && \text{by Lemma A.2.2.4}\\
&\quad= ((a \mathbin{{}^{\circ}\!\wedge} P^{\mathsf{T}}) \mathbin{{}^{\circ}\!\vee} Q^{\mathsf{T}}) \mathbin{{}^{\circ}\!\wedge} R^{\mathsf{T}}. && \text{by Lemma A.2.1}
\end{aligned}$$

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

$$\begin{aligned}
&f^c((a \mathbin{{}^{\circ}\!\wedge} P^{\mathsf{T}}) \mathbin{{}^{\circ}\!\vee} Q^{\mathsf{T}}, R^{\mathsf{F}})\\
&\quad= (a \mathbin{{}^{\circ}\!\vee} f^c(Q^{\mathsf{T}}, R^{\mathsf{F}})) \mathbin{{}^{\circ}\!\wedge} f^c(P^{\mathsf{T}}, R^{\mathsf{F}}) && \text{by (19)}\\
&\quad= (a \mathbin{{}^{\circ}\!\vee} (Q^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} R^{\mathsf{F}})) \mathbin{{}^{\circ}\!\wedge} (P^{\mathsf{T}} \mathbin{{}^{\circ}\!\wedge} R^{\mathsf{F}}) && \text{by induction hypothesis}\\
&\quad= (a \mathbin{{}^{\circ}\!\vee} ((Q^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T}) \mathbin{{}^{\circ}\!\wedge} (R^{\mathsf{F}} \mathbin{{}^{\circ}\!\wedge} \mathsf{F}))) \mathbin{{}^{\circ}\!\wedge}\\
&\qquad\ ((P^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T}) \mathbin{{}^{\circ}\!\wedge} (R^{\mathsf{F}} \mathbin{{}^{\circ}\!\wedge} \mathsf{F})) && \text{by Lemma A.2.1}\\
&\quad= ((a \mathbin{{}^{\circ}\!\wedge} (P^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T})) \mathbin{{}^{\circ}\!\vee} (Q^{\mathsf{T}} \mathbin{{}^{\circ}\!\vee} \mathsf{T})) \mathbin{{}^{\circ}\!\wedge} (R^{\mathsf{F}} \mathbin{{}^{\circ}\!\wedge} \mathsf{F}) && \text{by Lemma A.2.2.5}\\
&\quad= ((a \mathbin{{}^{\circ}\!\wedge} P^{\mathsf{T}}) \mathbin{{}^{\circ}\!\vee} Q^{\mathsf{T}}) \mathbin{{}^{\circ}\!\wedge} R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}$$

If the second argument is T-$*$-term, the result follows by (20) from the case where the second argument is a T-term, and (F7). □

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

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

Proof. The grammatical result is immediate by (21) 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*

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

Proof. By (22) 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 EqFSCL $\vdash f_1^c(P^*, Q^{\mathsf{T}}) = P^* \leftand Q^{\mathsf{T}}$. 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 $\hat{a} \in \{a, \neg a\}$:

$$\begin{aligned}
f_1^c((\hat{a} \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{T}}) &= (\hat{a} \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{T}})) \leftor Q^{\mathsf{F}} && \text{by (23), (24)}\\
&= (\hat{a} \leftand (P^{\mathsf{T}} \leftand R^{\mathsf{T}})) \leftor Q^{\mathsf{F}} && \text{by Lemma A.2.4}\\
&= ((\hat{a} \leftand P^{\mathsf{T}}) \leftand R^{\mathsf{T}}) \leftor Q^{\mathsf{F}} && \text{by (F7)}\\
&= ((\hat{a} \leftand P^{\mathsf{T}}) \leftand (R^{\mathsf{T}} \leftor \mathsf{T})) \leftor (Q^{\mathsf{F}} \leftand \mathsf{F}) && \text{by Lemma A.2.1}\\
&= ((\hat{a} \leftand P^{\mathsf{T}}) \leftor (Q^{\mathsf{F}} \leftand \mathsf{F})) \leftand (R^{\mathsf{T}} \leftor \mathsf{T}) && \text{by Lemma A.2.2.2}\\
&= ((\hat{a} \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}) \leftand R^{\mathsf{T}}. && \text{by Lemma A.2.1}
\end{aligned}$$

For the induction step we assume that the result holds for all $*$-terms with fewer ℓ-terms than $P^* \leftand Q^d$ and $P^* \leftor Q^c$. In the case of conjunctions the results follow from (25), the induction hypothesis, and (F7). In the case of disjunctions the results follow immediately from (26), the induction hypothesis, Lemma A.2.1, and the dual of (F10). □

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

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

Proof. By (27), Lemma A.2.4 and (F7) it suffices to prove that $f_2^c(P^*, Q^{\mathsf{F}})$ is an F-term and that EqFSCL $\vdash f_2^c(P^*, Q^{\mathsf{F}}) = P^* \leftand Q^{\mathsf{F}}$. 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 $(a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}$ as:

$$\begin{aligned}
f_2^c((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{F}}) &= (a \leftor Q^{\mathsf{F}}) \leftand f^c(P^{\mathsf{T}}, R^{\mathsf{F}}) && \text{by (28)}\\
&= (a \leftor Q^{\mathsf{F}}) \leftand (P^{\mathsf{T}} \leftand R^{\mathsf{F}}) && \text{by Lemma A.2.4}\\
&= ((a \leftor Q^{\mathsf{F}}) \leftand P^{\mathsf{T}}) \leftand R^{\mathsf{F}} && \text{by (F7)}\\
&= ((a \leftor (Q^{\mathsf{F}} \leftand \mathsf{F})) \leftand (P^{\mathsf{T}} \leftor \mathsf{T})) \leftand R^{\mathsf{F}} && \text{by Lemma A.2.1}\\
&= ((a \leftand (P^{\mathsf{T}} \leftor \mathsf{T})) \leftor (Q^{\mathsf{F}} \leftand \mathsf{F})) \leftand R^{\mathsf{F}} && \text{by Lemma A.2.2.2}\\
&= ((a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}) \leftand R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}$$

For ℓ-terms of the form $(\neg a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}$ we derive:

$$\begin{aligned}
&f_2^c((\neg a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}, R^{\mathsf{F}})\\
&\quad= (a \leftor f^c(P^{\mathsf{T}}, R^{\mathsf{F}})) \leftand Q^{\mathsf{F}} && \text{by (29)}\\
&\quad= (a \leftor (P^{\mathsf{T}} \leftand R^{\mathsf{F}})) \leftand Q^{\mathsf{F}} && \text{by induction hypothesis}\\
&\quad= (a \leftor ((P^{\mathsf{T}} \leftor \mathsf{T}) \leftand (R^{\mathsf{F}} \leftand \mathsf{F}))) \leftand (Q^{\mathsf{F}} \leftand \mathsf{F}) && \text{by Lemma A.2.1}\\
&\quad= ((\neg a \leftand (P^{\mathsf{T}} \leftor \mathsf{T})) \leftor (Q^{\mathsf{F}} \leftand \mathsf{F})) \leftand (R^{\mathsf{F}} \leftand \mathsf{F}) && \text{by Lemma A.2.2.6}\\
&\quad= ((\neg a \leftand P^{\mathsf{T}}) \leftor Q^{\mathsf{F}}) \leftand R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}$$

For the induction step we assume that the result holds for all $*$-terms with fewer ℓ-terms than $P^* \leftand Q^d$ and $P^* \leftor Q^c$. In the case of conjunctions the results follow from (30), 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 $f^n(f^c_1(P^*, f^n(R^{\mathsf{F}})))$ is a $*$-term with same number of ℓ-terms as P^*. The grammatical result follows from this fact, (31), and the induction hypothesis. Furthermore, noting that by the same argument $f^n(f^c_1(P^*, f^n(R^{\mathsf{F}}))) = \neg(P^* \leftand \neg R^{\mathsf{F}})$, we derive:

$$
\begin{aligned}
f^c_2(P^* \leftor Q^c, R^{\mathsf{F}}) &= f^c_2(f^n(f^c_1(P^*, f^n(R^{\mathsf{F}}))), f^c_2(Q^c, R^{\mathsf{F}})) && \text{by (31)}\\
&= f^n(f^c_1(P^*, f^n(R^{\mathsf{F}}))) \leftand (Q^c \leftand R^{\mathsf{F}}) && \text{by induction hypothesis}\\
&= \neg(P^* \leftand \neg R^{\mathsf{F}}) \leftand (Q^c \leftand R^{\mathsf{F}}) && \text{as shown above}\\
&= (\neg P^* \leftor R^{\mathsf{F}}) \leftand (Q^c \leftand R^{\mathsf{F}}) && \text{by (F3) and (F2)}\\
&= (\neg P^* \leftor (R^{\mathsf{F}} \leftand \mathsf{F})) \leftand (Q^c \leftand (R^{\mathsf{F}} \leftand \mathsf{F})) && \text{by Lemma A.2.1}\\
&= (P^* \leftor Q^c) \leftand (R^{\mathsf{F}} \leftand \mathsf{F}) && \text{by Lemma 2.1.4}\\
&= (P^* \leftor Q^c) \leftand R^{\mathsf{F}}. && \text{by Lemma A.2.1}
\end{aligned}
$$

This completes the proof. □

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

Proof. By the four preceding lemmas it suffices to show that

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

is in *SNF* and that $\text{EqFSCL} \vdash f^c(P^{\mathsf{T}} \leftand Q^*, R^{\mathsf{T}} \leftand S^*) = (P^{\mathsf{T}} \leftand Q^*) \leftand (R^{\mathsf{T}} \leftand S^*)$. By (F7) and (32), in turn, it suffices to prove that $f^c_3(P^*, Q^{\mathsf{T}} \leftand R^*)$ is a $*$-term and that $\text{EqFSCL} \vdash f^c_3(P^*, Q^{\mathsf{T}} \leftand R^*) = P^* \leftand (Q^{\mathsf{T}} \leftand R^*)$. We prove this by induction on the number of ℓ-terms in R^*. In the base case we have that $f^c_3(P^*, Q^{\mathsf{T}} \leftand R^\ell) = f^c_1(P^*, Q^{\mathsf{T}}) \leftand R^\ell$ by (33) 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 (34), and for disjunctions it follows from Lemma A.2.6, (F7) and (35). □

We can now easily prove Theorem 2.2.2:

Theorem 2.2.2 (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 (2) and axioms (F4), (F5) and its dual. If P is T or F the result follows from by (3) or (4). For the induction we get the result from definitions (5)-(7), Lemma A.2.3, Lemma A.2.8, and axiom (F2). □