On the Axiomatizability of Contractive Short-Circuit Logic

Jeroen van Wier

June 8, 2016
Abstract

Short-circuit evaluation is the concept of sequentially evaluating propositional variables in such a way that the second argument of a connective is only evaluated if the first does not suffice to determine the value of the expression. The left-sequential logic that describes this phenomenon is called short-circuit logic (SCL). In this work we discuss a variant of SCL called contractive short-circuit logic (CSCL), in which subsequent evaluations of the same propositional variable can be contracted into one evaluation. We first provide versions of SCL without constants for true and false and discuss their uses. We then provide an equational axiomatization for CSCL in a setting where only one propositional variable is available. Lastly we show that this result is not preserved in the case that more propositional variables are available and prove that no finite equational axiomatization exists for CSCL in any setting with at least three different propositional variables. We conclude with some remarks about the setting where two propositional variables exist.
1.1 What is SCL

In almost every modern-day programming language conditional statements are a core part of the language. It is hard to imagine a language such as Java or C without if and while statements. While the conditions in these constructs might seem like terms of standard propositional logic at first glance, their behaviour often deviates from propositional logic. Consider a statement \( x \land y \) in the Java programming language, where \( x \) and \( y \) are propositional variables, henceforth called atoms. These atoms can be anything from equality tests to function calls. Often these atoms have side effects that influence the evaluations of other atoms. Because of this, evaluation order is key within the Java language. Like many languages, Java uses a left-sequential evaluation strategy, so in the term \( x \land y \), \( x \) is evaluated before \( y \). Consider for example \((i = 3) \land (i == 3)\), where a single = stands for assignment and a double = stands for equality test. While this statement will always be \textit{true} when evaluated with a left-sequential evaluation strategy, it might become \textit{false} when reversed to \((i == 3) \land (i = 3)\). Thus a lack of commutativity is a key difference with propositional logic.

This phenomenon is dubbed Short-Circuit Logic (SCL) by Bergstra, Ponse and Staudt in [BPS13], of which the first version was published in 2010. SCL can generally be considered the logic of programming languages, although not all languages use SCL. SCL and Propositional Logic differ in three key aspects: evaluation order, side effects and short-circuit behaviour. Short-circuit behaviour is the evaluation strategy that only evaluates the least amount of atoms possible to determine the truth value of a term. In our example of \( x \land y \), this means that \( y \) is only evaluated if \( x \) is \textit{true}, for otherwise the outcome is already established to be \textit{false}. Note that the evaluation order is still maintained, thus a statement of the form \( x \land \textit{false} \) will still evaluate \( x \).

1.2 Related Work

Short-circuit evaluation is a concept that is integrated into many programming languages and is a concept that many programmers are familiar with. It allows programmers to write shorter code and is often used as an optimization in compilers[GGN07]. However, when short-circuit evaluation was first discussed, computer scientist were wary of its differences.
with propositional logic\cite{Dij87}.

In \cite{BP11}, Bergstra and Ponse define Proposition Algebra as a way to reason about sequential evaluation based on the ternary connective by Hoare\cite{Hoa85}. In \cite{BPS13} they use proposition algebra to define Short-Circuit Logic. They distinguish multiple variants of SCL, of which we will discuss Contractive SCL. In \cite{Sta12}, Staudt provides an axiomatization for Free SCL, the least identifying SCL, and he also introduces evaluation trees for Propositional Algebra and SCL. These trees are further explored by Bergstra and Ponse in \cite{BP15}. In \cite{Wor11}, Wortel discusses the definition of side effects using Quantified Dynamic Logic, and discusses similarities between QDL and variants of SCL.

1.3 Thesis Overview

In this thesis we will discuss Contractive SCL (CSCL), a variant of SCL. We will introduce the concept of Atomic SCLs and provide comparisons between CSCL and its atomic counterpart. The difference between SCL and Atomic SCL is the absence of constants for true and false in Atomic SCL. We use this notion of an Atomic SCL to provide a finite axiomatization for CSCL in a setting where only one atom exists. This notion of finite axiomatizations does not extend to settings with three or more atoms and we provide a proof that no finite equational axiomatization for CSCL can exist in this setting. We start in Chapter 2 by providing preliminary related work and formal definitions of concepts discussed in the introduction. In Chapter 3 we formally introduce Atomic SCL and Atomic CSCL, and prove that Atomic CSCL identifies almost as much as CSCL. Chapter 4 presents an axiomatization for CSCL using propositional connectives for the setting with only one atom. In Chapter 5 we give a proof that CSCL cannot be finitely axiomatized using the propositional connectives in a setting with three or more atoms. We conclude in Chapter 6 with some final remarks and suggestions for future work.
2.1 Proposition Algebra

The evaluation order of a statement in any short-circuit logic is what distinguishes it from standard propositional logic. In order to accurately describe this evaluation order we will use proposition algebra [BP11], based on the ternary connective

\[ x \triangleleft y \triangleright z. \]

The statement \( x \triangleleft y \triangleright z \) can be semantically interpreted as

if \( y \) then \( x \) else \( z \).

In \( x \triangleleft y \triangleright z \), \( y \) is first evaluated, and if \( y \) resulted in \textit{true}, \( x \) is then evaluated and otherwise \( z \) is evaluated. In [Hoa85], Hoare shows that propositional logic can be characterized using the signature

\[ \Sigma_{\text{CP}}(A) = \{ T, F, \triangleleft, \triangleright, a | a \in A \}, \]

with \( T \) and \( F \) for \textit{true} and \textit{false} respectively and \( A \) a countable set of atoms. He also provides a set of axioms for this, of which Table 2.1 is the subset that defines the conditional as a primitive connective.

\[ \text{Definition 2.1. Given a countable set } A \text{ of atoms, the set } C_A \text{ of closed terms over } \Sigma_{\text{CP}}(A) \text{ is defined inductively by} \]

\[ t ::= T \mid F \mid a \mid t \triangleleft t \triangleright t \]

\[ x \triangleleft T \triangleright y = x \quad (\text{CP1}) \]
\[ x \triangleleft F \triangleright y = y \quad (\text{CP2}) \]
\[ T \triangleleft x \triangleright F = x \quad (\text{CP3}) \]
\[ x \triangleleft (y \triangleleft z \triangleright u) \triangleright v = (x \triangleleft y \triangleright v) \triangleleft z \triangleright (x \triangleleft u \triangleright v) \quad (\text{CP4}) \]

Table 2.1: The set CP of equational axioms for proposition algebra
where \( a \) ranges over \( A \).

We will henceforth write \( \text{Var} \) for the collection of all variables in open terms and \( A \) for a countable set of atoms.

### 2.2 Short Circuit Logic (SCL)

In the introduction we mentioned Java, which uses the &&, \&\& and ! connectives for the short-circuited versions of ‘and’, ‘or’ and ‘not’ respectively. To maintain consistency with previous research and increase readability we will from now on use the notations \( \land, \lor \) and \( \lnot \) for &&, \&\& and ! respectively. In order to reason about the \( \land, \lor \) and \( \lnot \) connectives in any context, we will first define these connectives with the conditional as primary connective.

Firstly, the \( \lnot \) can be defined by

\[
\lnot x = \begin{cases} F & \text{if } x = T \\ T & \text{if } x = F \end{cases}
\]

Secondly, we define \( \land \) by

\[
x \land y = \begin{cases} y & \text{if } x = F \\ x \land T & \text{if } x = T \land y \end{cases}
\]

Lastly, we define \( \lor \) using \( \lnot \) and \( \land \), similar to their counterparts in propositional logic, as

\[
x \lor y = \lnot (\lnot x \land \lnot y) = \begin{cases} T & \text{if } x = F \land y = T \\ F & \text{if } x = F \land y = F \lor x = T \land y = F \lor x = T \land y = T \end{cases}
\]

Using these connectives we can properly define a short-circuit logic with the use of the export operator \( \square \) of module algebra [BHK90]. Simply put, in \( S \square X \), with \( S \) a signature and \( X \) a set of equations or module, all signature elements not in \( S \) are declared auxiliary. In this case the conditional is declared an auxiliary operator.

**Definition 2.2.** A **short-circuit logic** is a logic that implies the consequences of the module expression

\[
\{ T, \lnot, \land, a \mid a \in A \} \square (\text{CP} + \{ \neg x = F < x \triangleright T \} + \{ x \land y = y < x \triangleright F \}).
\]

We further write \( \text{SCL} \) for this module expression, and we write \( \text{SCL} \vdash t = t' \) if \( t \) and \( t' \) are terms over the signature

\[
\Sigma_{\text{SCL}}(A) = \{ T, \lnot, \land, a \mid a \in A \}
\]

and

\[
\text{CP} + \{ \neg x = F < x \triangleright T \} + \{ x \land y = y < x \triangleright F \}) \vdash t = t'.
\]

This definition, as well as the definitions for the connectives, are taken from [BPS13].

**Definition 2.3.** Given a countable set \( A \) of atoms, the set \( \text{PS}_A \) of closed terms over \( \Sigma_{\text{SCL}}(A) \) is defined inductively by

\[
t ::= T \mid F \mid a \mid \lnot t \mid t \land t
\]

where \( a \) ranges over \( A \).
\[
\begin{align*}
F &= \neg T & \text{(SCL1)} \\
x \lor y &= \neg (\neg x \land \neg y) & \text{(SCL2)} \\
\neg x &= x & \text{(SCL3)} \\
T \land x &= x & \text{(SCL4)} \\
x \land T &= x & \text{(SCL5)} \\
F \land x &= F & \text{(SCL6)} \\
(x \land y) \land z &= x \land (y \land z) & \text{(SCL7)} \\
x \land F &= \neg x \land F & \text{(SCL8)} \\
(x \land F) \lor y &= (x \lor T) \land y & \text{(SCL9)} \\
(x \land (z \land F)) \land (y \lor (z \land F)) &= (x \land (z \land F)) \land (y \lor (z \land F)) & \text{(SCL10)} \\
\end{align*}
\]

Table 2.2: EqFSCL, a set of axioms for FSCL

Definition 2.2 only states the minimum requirements of a SCL. Because of this a number of variants of SCL exist, of which we will discuss four. The least identifying SCL is free short-circuit logic (FSCL), which makes no assumptions about the side effects of atoms.

**Definition 2.4.** FSCL (free short-circuit logic) is the short-circuit logic that implies no other consequences than those of the module expression $\Sigma_{SCL}(A)$.

The equations in Table 2.2 axiomatize FSCL, as is proven in [BPS13].

The most identifying SCL is static short-circuit logic (SSCL), which assumes that no atoms have any side effects. Because of this assumption, the order of evaluation of atoms no longer affects their produced truth values. This behaviour creates a strong similarity between SSCL and propositional logic.

**Definition 2.5.** SSCL (static short-circuit logic) is the short-circuit logic that implies no other consequences than those of the module expression $\Sigma_{SCL}(A)$.

The equations in Table 2.3 axiomatize SSCL, as is proven in [BPS13].

The last variant of SCL we will discuss is contractive short-circuit logic (CSCL), which identifies more than FSCL but less than SSCL. CSCL is the short circuit logic in which repeated subsequent evaluation of the same atom not only yields the same truth value, but also has no additional side effect after the first evaluation. We define $CP_{cr}(A)$ as an extension of CP with

\[
\begin{align*}
(x \triangleleft a \triangleright y) \triangleright a \triangleright z &= x \triangleleft a \triangleright z, & \text{(CPcr1)} \\
x \triangleleft a \triangleright (y \triangleleft a \triangleright z) &= x \triangleleft a \triangleright z. & \text{(CPcr2)}
\end{align*}
\]

where $a$ ranges over $A$, as presented in [BP11].
is defined recursively by Definition 2.8.

The operator \( \ominus \) is called post-conditional composition over \( a \). In the evaluation tree \( X \ominus a \bowtie Y \), the root is represented by \( a \), the left branch by \( X \) and the right branch by \( Y \). The depth of an evaluation tree \( X \) is defined recursively by \( d(T) = d(F) = 0 \) and \( d(Y \ominus a \bowtie Z) = 1 + \max(d(Y), d(Z)) \).

**Definition 2.8.** The replacement operator on \( T_A \), denoted \( X'[T \mapsto Y, F \mapsto Z] \)

is defined recursively by \( T'[T \mapsto Y, F \mapsto Z] = Y \), \( F'[T \mapsto Y, F \mapsto Z] = Z \), and

\[
(X' \ominus a \bowtie X'')[T \mapsto Y, F \mapsto Z] = X'[T \mapsto Y, F \mapsto Z] \ominus a \bowtie X''[T \mapsto Y, F \mapsto Z].
\]
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., $X[F \mapsto Z] = X[T \mapsto T, F \mapsto Z]$. Repeated replacements satisfy the following identity:

$$X[T \mapsto Y_1, F \mapsto Z_1][T \mapsto Y_2, F \mapsto Z_2] = X[T \mapsto Y_1][T \mapsto Y_2, F \mapsto Z_2], \quad F \mapsto Z_1[T \mapsto Y_2, F \mapsto Z_2].$$

We also note that functions bind stronger than replacement, thus for any function $f$ and any input $x$ for $f$

$$f(x)[T \mapsto X, F \mapsto Y] = (f(x))[T \mapsto X, F \mapsto Y].$$

Staudt also defines a function for translating statements from proposition algebra to evaluation trees.

**Definition 2.9.** The unary short-circuit evaluation function $se : C_A \to T_A$ is defined as follows, where $a \in A$:

$$se(T) = T, \quad se(F) = F, \quad se(a) = T \underline{\iff} a \underline{\iff} F,$$

$$se(P \land Q \land R) = se(Q)[T \mapsto se(P), F \mapsto se(R)].$$

For SCL we can also derive the cases

$$se(P \land Q) = se(P)[T \mapsto se(Q)],$$

$$se(\neg P) = se(P)[T \mapsto F, F \mapsto T].$$

**Definition 2.10.** Free valuation congruence, notation $\equiv_{se}$, is defined on $C_A$ as follows:

$$P \equiv_{se} Q \iff se(P) = se(Q)$$

In [BP15] the $\equiv_{se}$ relation is proven to be a congruence on terms over $\Sigma_{CP}(A)$ and it easily follows that it also is a congruence on $PS_A$.

In FSCL the trees produced by $se$ are interpretations of SCL terms. For CSCL [BP15] defines a function, the contractive short-circuit evaluation function, that contracts repeated atoms as much as possible.

**Definition 2.11.** Let $B, B' \in \{T, F\}$. The unary contractive short-circuit evaluation function $cse : C_A \to T_A$ yields contractive evaluation trees and is defined by

$$cse(P) = cr(se(P)),$$

where the auxiliary function $cr : T_A \to T_A$ is defined as follows ($a \in A$):

$$cr(T) = T,$$

$$cr(F) = F,$$

$$cr(X \underline{\iff} a \underline{\iff} Y) = cr(H_a(X)) \underline{\iff} a \underline{\iff} cr(J_a(Y)).$$

For $a \in A$, the auxiliary functions $H_a : T_A \to T_A$ and $J_a : T_A \to T_A$ are defined by

$$H_a(T) = T,$$

$$H_a(F) = F,$$

$$H_a(X \underline{\iff} b \underline{\iff} Y) = \begin{cases} H_a(X) & \text{if } b = a \\ X \underline{\iff} b \underline{\iff} Y & \text{otherwise} \end{cases}$$
The evaluation tree $se(P)$

\[
\begin{array}{c}
\text{F} \\
\text{T}
\end{array}
\]

The evaluation tree $cse(P)$

\[
\begin{array}{c}
\text{F} \\
\text{F}
\end{array}
\]

Figure 2.1: The difference between $se(P)$ and $cse(P)$ for $P = F < \alpha > a$.

and

\[
\begin{align*}
J_a(T) &= T, \\
J_a(F) &= F, \\
J_a(X \triangleq b \triangleright Y) &= \begin{cases} 
J_a(Y) & \text{if } b = a \\
X \triangleq b \triangleright Y & \text{otherwise}
\end{cases}
\end{align*}
\]

Definition 2.12. *Contractive valuation congruence*, notation $=_{cse}$, is defined on $C_A$ as follows:

\[
P =_{cse} Q \iff cse(P) = cse(Q)
\]

In [BP15] the $=_{cse}$ relation is proven to be a congruence on terms over $\Sigma_{CP}(A)$ and it easily follows that it also is a congruence on $PS_A$.

Example 2.13. Let $P = F < \alpha > a$. We demonstrate the difference between $se(P)$ and $cse(P)$ in Figure 2.1.

2.4 Substitutions

Evaluation trees provide a way to interpret SCL terms, but they can only interpret closed terms. In order to bridge the gap between open and closed terms we use substitutions.

Definition 2.14. A *substitution* is a mapping $\sigma$ from variables to terms. We write

\[
\sigma = \{ x_1 \mapsto t_1, \ldots, x_k \mapsto t_k \}
\]

to denote that the substitution $\sigma$ maps $x_i$ to $t_i$ for all $i \in \{1, \ldots, k\}$, and maps every other variable to itself. A term $\phi$ is a *substitution instance* of $\psi$ if there exists a substitution $\sigma$ such that

\[
\sigma(\psi) = \phi.
\]

Substitutions on terms are defined in the usual way.

Substitutions can instantiate an open term to a closed term, but they can also make the structure of an open term more specific, meaning that they can introduce more structure. We distinguish different types of substitutions to more accurately describe the impact of a substitution.
Definition 2.15. A substitution is called a **renaming substitution** if it is a permutation on the set of all variables.

Renaming substitutions only map variables to other variables in such a way that the substitution is invertible, which means every variable is the result of at most one renaming. An example of a renaming substitution is \( \{ y_1 \mapsto y_2, y_2 \mapsto y_1 \} \).

Definition 2.16. A substitution is called a **ground substitution** if it maps all variables to terms without variables. We use the shorthand notation

\[
\sigma = [x_1 \mapsto t_1, \ldots, x_k \mapsto t_k],
\]

**Note**: \( \sigma \) maps \( x_i \) to a closed term \( t_i \) for every \( i \in \{1, \ldots, k\} \).

Ground substitution are used to change an open term to a closed one. Simple examples of ground substitutions are \( \{x \mapsto a\} \) and \( \{y_1 \mapsto a\} \).

Definition 2.17. A substitution is called an **atomic substitution** if it maps all variables to terms over \( \varSigma_{SCL} \) without \( T \) or \( F \) in them. In a CP-setting, an atomic substitution maps all variables to terms representing terms over \( \varSigma_{SCL}(A) \) without \( T \) or \( F \) in them.

Atomic substitutions are specific to CP and SCL and provide an interesting case, since they do not influence what part of a term still gets evaluated after substitution. Take for example

\[
\phi = x \land a
\]

and

\[
\sigma = \{x \mapsto F\}.
\]

Note that \( \sigma \) is not an atomic substitution and influences the evaluation of \( \phi \), since \( a \) is never evaluated in \( \sigma(\phi) \). Atomic substitutions do not have the ability to do this, as is proven as part of Lemma 5.1. An example of an atomic substitution in a SCL-setting is \( \{y_1 \mapsto a \land b\} \) and in a CP-setting this substitution would be \( \{y_1 \mapsto b \land a \lor F\} \).

Substitution instances provide a way to make a term more specific. As an inverse of this we have substitution preimages.

Definition 2.18. Given a term \( \phi \) over \( \varSigma_{SCL}(A) \), the set \( SP(\phi) \) of **substitution preimages** of any term \( \phi \) is defined as

\[
\{ \psi \mid \phi \text{ is a substitution instance of } \psi \},
\]

modulo renaming.

Substitution preimages allow us to reason about what a term might have looked like before substitution. For example, let

\[
\phi = y_1 \land (y_2 \lor y_3),
\]

then

\[
SP(\phi) = \{y_1, y_1 \land y_2, y_1 \land (y_2 \lor y_3)\}.
\]

Due to the nature of substitution, any term can be the substitution instance of infinitely many terms, since there are infinitely many variables and any variable is a substitution preimage of any term. To circumvent this, \( SP \) only contains elements that are not a renaming substitution instance of each other. Because of this, \( SP(\phi) \) is finite for any term \( \phi \).
2.5 Contractibility

In order to reason about terms in a CSCL-setting, we distinguish between terms that can be contracted to a smaller term and terms that are already minimal with respect to the contraction of atoms. If a term is already minimal we will call it incontractible.

**Definition 2.19.** A closed term $\phi$ is **incontractible** if

$$cse(\phi) = se(\phi).$$

A closed term $\psi$ is **contractible** if $\psi$ is not incontractible.

Incontractibility is useful for distinguishing between pairs of terms that are equal under CSCL only and pairs that are equal under FSCL, since contractibility can only be introduced by CSCL. This notion will play a key part in Chapter 5.

**Lemma 2.20.** For any $\phi, \psi \in PS_A$, if $\phi =_{se} \psi$ and $\phi$ is incontractible, then so is $\psi$.

**Proof.** Since $\phi$ is incontractible, $cse(\phi) = se(\phi)$. Since $\phi =_{se} \psi$, $se(\phi) = se(\psi)$. Consequently,

$$cse(\psi) = cse(se(\psi))$$

$$= cse(se(\psi))$$

$$= cse(\phi)$$

$$= se(\phi)$$

$$= se(\psi)$$

Thus, $\psi$ is incontractible.

Another useful property of incontractibility is that the existence of an incontractible substitution instance is preserved under $SP$. This will be useful for reasoning about axioms that are substitution preimages of the terms they are applied to.

**Lemma 2.21.** If there exists a substitution $\sigma$ such that $\sigma(\phi)$ is incontractible, then for every $\psi \in SP(\phi)$ there exists a $\sigma_1$ such that $\sigma_1(\psi)$ is incontractible.

**Proof.** By definition, for every $\psi \in SP(\phi)$ there exists a $\sigma_2$ such that $\sigma_2(\psi) = \phi$. Thus $\sigma(\sigma_2(\psi))$ is incontractible.

\[ \square \]
3.1 Atomic CSCL (ACSCL)

Contraction in CSCL takes place on the atomic level only, implying that \( a \land a =_{	ext{cas}} a \) but \( (a \land b) \land (a \land b) \neq_{	ext{cas}} a \land b \). To accentuate this, we define a logic in which only atoms and connectives are present:

**Definition 3.1.** An Atomic Short-Circuit Logic (ASCL) is a logic that implies the consequences of the module expression

\[
\{ \neg, \land, a \mid a \in A \} \sqcup (\text{CP} + (\neg x = F \land x \land T) + (x \land y = y \land x \land F)).
\]

Given a countable set \( A \) of atoms and \( \Sigma_{\text{atm}}(A) = \{ \neg, \land, a \mid a \in A \} \), the set of closed terms \( PS_{\text{atm}}(A) \subset PS_A \) over \( \Sigma_{\text{atm}}(A) \) is inductively defined by

\[
t := a \mid t \land t \mid \neg t
\]

where \( a \) ranges over \( A \).

**Definition 3.2.** ACSCL(A) (atomic contractive short-circuit logic) is the ASCL that implies no other consequences than those of the module expression

\[
\{ \neg, \land, a \mid a \in A \} \sqcup (\text{CP}_{\text{at}}(A) + (\neg x = F \land x \land T) + (x \land y = y \land x \land F)).
\]

To improve readability we will use ACSCL to refer to ACSCL(A) when not discussing a specific \( A \). The only difference between ACSCL and CSCL is the absence of \( T \) and \( F \) in ACSCL. To express the effect of these constants within ACSCL, we introduce the function \( f_F \) as:

**Definition 3.3.** Given a set of atoms \( A \), the falsification function \( f_F : PS_{\text{atm}}(A) \to PS_{\text{atm}}(A) \) is defined as

\[
\begin{align*}
f_F(a) &= a \land \neg a, \\
f_F(\neg \phi) &= f_F(\phi), \\
f_F(\phi \land \psi) &= f_F(\phi) \land f_F(\psi),
\end{align*}
\]

where \( a \) ranges over \( A \).
Lemma 3.4. Given a set of atoms \( A \), for any \( \phi \in PS_{atm}(A) \)

\[ f_{\mathcal{F}}(\phi) = \text{cse} \, \phi \land \mathcal{F} \]

Proof. By induction on the complexity of \( \phi \). For the basic case \( \phi = a \),

\[ \text{cse}(f_{\mathcal{F}}(\phi)) = \text{cse}(f_{\mathcal{F}}(a)) = \text{cse}(a \land \neg a) = \mathcal{F} \neg a \mathcal{F} = \text{cse}(a \land \mathcal{F}). \]

For the inductive case \( \phi = \neg \psi \),

\[ f_{\mathcal{F}}(\phi) = \text{cse} \, f_{\mathcal{F}}(\neg \psi) = \text{cse} \, f_{\mathcal{F}}(\psi) \land \mathcal{F} = \text{cse} \, \psi \land \mathcal{F} \quad \text{Induction Hypothesis} \]

\[ = \text{cse} \, \neg \psi \land \mathcal{F} = \text{cse} \, \phi \land \mathcal{F} \quad (\text{SCL8}) \]

For the inductive case \( \phi = \psi \land \chi \),

\[ f_{\mathcal{F}}(\phi) = \text{cse} \, f_{\mathcal{F}}(\psi \land \chi) = \text{cse} \, \psi \land f_{\mathcal{F}}(\chi) = \text{cse} \, \psi \land (\chi \land \mathcal{F}) \quad \text{Induction Hypothesis} \]

\[ = \text{cse} \, (\psi \land \chi) \land \mathcal{F} \quad (\text{SCL7}) \]

\[ = \text{cse} \, \phi \land \mathcal{F} \]

\[ \square \]

It is obvious that not every term in \( PS_A \) can be expressed by a term in \( PS_{atm}(A) \), a simple example being \( T \in PS_A \) but \( T \not\in PS_{atm}(A) \). However, ACSCL provides an interesting case where any term in \( PS_A \) not semantically equal to \( T \) or \( F \), is equal to a term in \( PS_{atm}(A) \) modulo contraction.

Theorem 3.5. Given a countable non-empty set of atoms \( A \), for any \( \phi \in PS_A \), with \( \text{se}(\phi) \not\in \{ T, F \} \) there exists \( \psi \in PS_{atm} \) such that

\[ \phi = \text{cse} \, \psi \]

Proof. By induction on the complexity of \( \phi \). The basic case of \( \phi = a \) for any \( a \in A \) is trivial, since then \( \phi \in PS_{atm}(A) \) and \( \phi = \text{cse} \, \phi \). The induction case of \( \phi = \neg \chi \) can be proven by 2 simple cases:

1. If \( \text{se}(\chi) \not\in \{ T, F \} \), then by the induction hypothesis a \( \chi' \in PS_{atm}(A) \) exists such that \( \chi = \text{cse} \, \chi' \). Since \( = \text{cse} \) is a congruence relation, this implies \( \neg \chi = \text{cse} \, \neg \chi' \). Since \( \chi' \in PS_{atm}(A) \), by definition \( \neg \chi' \in PS_{atm}(A) \).

2. If \( \text{se}(\chi) \in \{ T, F \} \), then \( \text{se}(\phi) \in \{ T, F \} \), which contradicts our limitations.

In the last case, when \( \phi = \phi' \land \psi' \), four cases can be distinguished:
1. \( se(\phi') = T \), thus \( \phi = cse \psi' \). By the induction hypothesis a \( \psi'' \in PS_{atm}(A) \) exists such that \( \psi'' = cse \psi' = cse \phi \).

2. \( se(\psi') = T \), thus \( \phi = cse \phi' \). By the induction hypothesis a \( \phi'' \in PS_{atm}(A) \) exists such that \( \phi'' = cse \phi' = cse \phi \).

3. \( se(\phi') \notin \{T,F\} \) and \( se(\psi') \notin \{T,F\} \). By the induction hypothesis a \( \phi'' \in PS_{atm}(A) \) and a \( \psi'' \in PS_{atm}(A) \) exist such that \( \phi'' = cse \phi' \) and \( \psi'' = cse \psi' \). Since \( =cse \) is a congruence it is also true that \( \phi'' \land \psi'' = cse \phi'' \land \psi'' \).

4. \( se(\psi') = F \), thus \( \phi = cse \phi' \land F \). By the induction hypothesis a \( \phi'' \in PS_{atm}(A) \) exists such that \( \phi'' = cse \phi'' \). From this we can conclude that

\[
\begin{align*}
\phi &= cse \phi' \land F \\
&= cse \phi'' \land F \\
&= cse f_F(\phi'') \\
\end{align*}
\]

By Lemma 3.4

Note that, since \( se(\phi) \notin \{T,F\} \), it can never be the case that \( se(\phi') \in \{T,F\} \) and \( se(\psi') \in \{T,F\} \). Similarly it can never be the case that \( se(\phi') = F \), since then \( se(\phi' \land \psi') = F[T \mapsto se(\psi') = F] \).

3.2 Other ASCLs

ACSL provides an interesting case, in which only the truth values true and false cannot be expressed. For other SCLS this is not always the case. For example, consider the difference between FSCL and AFSC, the atomic counterpart of FSCL. The only terms containing \( T \) of \( F \) that can be expressed in AFSC are those in which the constant has no effect on the term. For example, \( T \land a \) can be expressed, since \( FSCL \vdash T \land a = a \), but any term containing \( x \land \neg a \) cannot be expressed in AFSC.

The other extreme of the ASCLs is ASSCL, the atomic version of SSCL. Since SSCL assumes no atoms have side effects, \( SSCL \vdash a \land \neg a = F \) and \( SSCL \vdash a \lor \neg a = T \). Since \( T \) and \( F \) can be directly expressed in ASSCL, any term containing \( T \) and \( F \) can also be expressed in ASSCL.
As discussed in the previous chapter, the contractive behaviour of CSCL takes place on the atomic level. This provides an interesting case when only one atom exists, say $A = \{a\}$. Any evaluation tree $X$ with $d(X) > 1$ can be contracted to a tree $Y$ with $d(Y) = 1$. Because of this property, CSCL$(A)$ only requires six normal forms for the case $A = \{a\}$: two for $T$ and $F$ and four more for $B\bar{a}C$ with $B, C \in \{T, F\}$.

4.1 The atomic setting

\begin{align*}
\neg\neg x &= x & \text{(DNS)} \\
\neg(x \land y) &= \neg x \lor \neg y & \text{(DM)} \\
(x \land y) \land z &= x \land (y \land z) & \text{(Assoc)} \\
a \land a &= a & \text{(C1 and D1)} \\
\neg a \land a &= a \land \neg a & \text{(C2 and D2)} \\
a \land (a \lor x) &= a & \text{(C3 and D3)} \\
a \land (\neg a \lor x) &= a \land x & \text{(C4 and D4)} \\
(a \land \neg a) \land x &= a \land \neg a & \text{(C5 and D5)} \\
(a \lor \neg a) \land a &= a & (a \land \neg a) \lor a = a & \text{(C6.1, D6.1)} \\
(a \lor \neg a) \land (a \lor \neg a) &= a \lor \neg a & \text{(C6.2, D6.2)}
\end{align*}

Table 4.1: The axiom set EqAC$\text{CSCL}(A)$ for $A = \{a\}$. 
We prove that EqACSCL(A) is a complete axiomatization of $=_{cse}$ for the case $A = \{a\}$, using normal forms in

$$NF_A = \{a, \neg a, a \lor \neg a, a \land \neg a\}.$$

Lemma 4.1. Let $A = \{a\}$, then
1. $\forall \phi \in NF_A \exists \psi \in NF_A : \text{EqACSCL}(A) \vdash \neg \phi = \psi.$
2. $\forall \phi, \psi \in NF_A \exists \chi \in NF_A : \text{EqACSCL}(A) \vdash \phi \land \psi = \chi.$

Proof. Clause 1 follows easily from DNS and DM (and their duals). Clause 2 follows from this case distinction:

$$\phi = a \quad \text{and} \quad \begin{cases} 
\psi = a : & \text{apply C1,} \\
\psi = \neg a : & \text{done,} \\
\psi = a \lor \neg a : & \text{apply C3,} \\
\psi = a \land \neg a : & \text{apply Assoc and C1 (resulting in } a \land \neg a),\end{cases}$$

$$\phi = \neg a \quad \text{and} \quad \begin{cases} 
\psi = a : & \text{apply C2,} \\
\psi = \neg a : & \text{apply DM and D1,} \\
\psi = a \lor \neg a : & \text{apply DM, D4, and D1 (resulting in } \neg a), \\
\psi = a \land \neg a : & \text{apply Assoc and C5 (resulting in } a \land \neg a),\end{cases}$$

$$\phi = a \lor \neg a \quad \text{and} \quad \begin{cases} 
\psi = a : & \text{apply C6.1,} \\
\psi = \neg a : & \text{apply DM, C2, and D.6.1} \\
\psi = a \lor \neg a : & \text{apply C.6.2} \\
\psi = a \land \neg a : & \text{apply Assoc and C.6.1}\end{cases}$$

$$\phi = a \land \neg a \quad \text{and} \psi \in NF_A : \text{apply C5.} \quad \square$$

Lemma 4.2. Let $A = \{a\}$. For each closed term $\phi$ over $\Sigma_{atm}(A)$, there exists $\psi \in NF_A$ such that $\text{EqACSCL}(A) \vdash \phi = \psi$.

Proof. By structural induction on $\phi$. If $\phi = a$, then $\phi \in NF_A$. If $\phi = \neg \phi_1$, then

$$\text{EqACSCL}(A) \vdash \phi = \neg \phi_1 = \neg \phi'_1 \quad \phi'_1 \in NF_A \text{ By Induction Hypothesis}$$

$$= \phi_2. \quad \phi_2 \in NF_A \text{ By Lemma 4.1}$$

If $\phi = \phi_1 \land \phi_2$, then

$$\text{EqACSCL}(A) \phi = \phi_1 \land \phi_2 = \phi'_1 \land \phi'_2 \quad \phi'_1, \phi'_2 \in NF_A \text{ By Induction Hypothesis}$$

$$= \phi_3. \quad \phi_3 \in NF_A \text{ By Lemma 4.1} \quad \square$$

Theorem 4.3. For the case $A = \{a\}$, EqACSCL(A) is a complete axiomatization of $=_{cse}$.

18
Proof. The soundness of EqACSCL(A) follows easily, since $=_\text{cse}$ is a congruence relation.

In order to prove completeness, assume $P =_\text{cse} Q$. By Lemma 4.2 there are $P', Q' \in NF_A$ with EqACSCL(A) $\vdash P = P'$ and EqACSCL(A) $\vdash Q = Q'$. By soundness $P =_\text{cse} P'$ and $Q = Q'$, thus $P' =_\text{cse} Q'$ and hence $P' = Q'$. This implies that EqACSCL(A) $\vdash P' = Q'$ and thus EqACSCL(A) $\vdash P = Q$. \qed

4.2 The general setting

The extension to an equational axiomatization for CSCL(A) for $A = \{a\}$ requires the addition of all other axioms of EqFSCL and these three axioms to EqACSCL(A):

\begin{align*}
a \lor T &= a \lor \neg a & a \land F &= a \land \neg a & (C7 \text{ and D7}) \\
(a \lor \neg a) \land x &= (a \land \neg a) \lor x & \text{(A8)}
\end{align*}

We only use axiom A8 to reduce $(a \lor \neg a) \land F$ and $(a \land \neg a) \lor T$ and as such these two instances would also suffice. We call the resulting set of axioms EqCSCL(A).

Extend $NF_A$ with $\{T, F\}$ and write $NF_A^+$ for the resulting set of normal forms. The analogues of Lemmas 4.1 and 4.2 for $NF_A^+$ follow easily, which implies the following result.

**Theorem 4.4.** For the case $A = \{a\}$, EqCSCL(A) is a complete axiomatization of $=_\text{cse}$.

Proof. The soundness of EqCSCL(A) follows easily, since $=_\text{cse}$ is a congruence relation.

In order to prove completeness, assume $P =_\text{cse} Q$. Then, for some $P', Q' \in NF_A^+$, EqCSCL(A) $\vdash P = P'$ and EqCSCL(A) $\vdash Q = Q'$. By soundness, $P =_\text{cse} P'$ and $Q =_\text{cse} Q'$, thus $P' =_\text{cse} Q'$ and hence $P' = Q'$. This implies that EqCSCL(A) $\vdash P' = Q'$ and thus EqCSCL(A) $\vdash P = Q$. \qed
Because any term can be reduced to one of six normal forms in EqCSCL(A) when \( A = \{a\} \), any structure introduced by the connectives in this term is removed. This property does however not extend to any EqCSCL(A) with \(|A| > 1\), as even with two atoms an incontractible term of any finite length can be made, for example \( a \land b \land \ldots \land a \land b \).

We use this structure to prove that no finite EqCSCL(A) exists for \(|A| \geq 3\).

### 5.1 Contractibility in the CP setting

In order to establish this proof we will first show that, in the world of CP, at least one of the right-hand and left-hand side of an equality must contain a contractible pair of atoms that is still reachable after any substitution.

**Lemma 5.1.** For all open terms \( t, u \) over \( \Sigma_{CP}(A) \), if there exists a ground atomic substitution \( \sigma_1 \) such that \( \sigma_1(t) \) is incontractible and a ground atomic substitution \( \sigma_2 \) such that \( \sigma_2(u) \) is incontractible, then

\[
CP_{\sigma_1}(A) \vdash t = u \quad \Rightarrow \quad CP \vdash t = u.
\]

**Proof.** The set PBF of **proto-basic forms** is defined inductively by

\[
t ::= T \mid F \mid t \triangleleft a \triangleright t \mid t \triangleleft x \triangleright t
\]

where \( a \) ranges over \( A \) and \( x \) ranges over \( Var \).

**Fact 1:** For every term \( t \) over \( \Sigma_{CP}(A) \) there exists a \( t' \in \text{PBF} \) such that \( CP \vdash t = t' \).

We prove Fact 1 by first showing that if \( t_1, t_2, t_3 \) are proto-basic forms, then a proto-basic form \( t_4 \) exists such that \( CP \vdash t_4 = t_1 \triangleleft t_2 \triangleright t_3 \). This can be proven by structural induction on \( t_2 \). If \( t_2 = T \) or \( t_2 = F \), then this follows immediately since \( t_4 = t_1 \) or \( t_4 = t_3 \) respectively. If \( t_2 = u_1 \triangleleft a \triangleright u_2 \) with \( a \in A \), then

\[
CP \vdash t_1 \triangleleft t_2 \triangleright t_3 = t_1 \triangleleft (u_1 \triangleleft a \triangleright u_2) \triangleright t_3 = (t_1 \triangleleft u_1 \triangleright t_2) \triangleleft a \triangleright (t_1 \triangleleft u_2 \triangleright t_2)
\]
By induction there exist $v_1, v_2 \in \text{PBF}$ such that $\text{CP} \vdash v_1 = (t_1 \triangleleft u_1 \triangleright t_2)$ and $\text{CP} \vdash v_2 = (t_1 \triangleleft u_2 \triangleright t_2)$, and thus $\text{CP} \vdash v_1 \triangleleft a \triangleright v_2 = (t_1 \triangleleft u_1 \triangleright t_2) \triangleleft a \triangleright (t_1 \triangleleft u_2 \triangleright t_2)$ and by the definition of PBF, $v_1 \triangleleft a \triangleright v_2 \in \text{PBF}$. The same argument can be made for $t_2 = u_1 \triangleleft x \triangleright u_2$ with $x \in \text{Var}$.

We can now prove Fact 1 by structural induction on $t$. If $t = T$ or $t = F$ then $t$ is already a basic form. If $t = a$ or $t = x$ then $\text{CP} \vdash t = T \triangleleft a \triangleright F$ or $\text{CP} \vdash t = T \triangleleft x \triangleright F$ respectively. If $t = t_1 \triangleleft t_2 \triangleright t_3$, then by induction there exist $t_1, t_2, t_3 \in \text{PBF}$ such that $\text{CP} \vdash t_1 \triangleleft t_2 \triangleright t_3$, thus $\text{CP} \vdash t = t_1 \triangleleft t_2 \triangleright t_3$. By the first result a $t'$ exists such that $\text{CP} \vdash t' = t_1 \triangleleft t_2 \triangleright t_3 = t$.

**Fact 2:** For all $t, u \in \text{PBF}$, if $\text{CP} \vdash t = u$, then $t = u$

We prove Fact 2 by structural induction on $t$ and distinguish cases for $u$. Consider the case where $t = T$. If $u = F$, $u = u_1 \triangleleft a \triangleright u_2$ or $u = u_1 \triangleleft x \triangleright u_2$, then $\text{CP} \not\vdash t = u$. If $u = T$, then $t = u$. A similar argument can be made for the case where $t = F$.

Consider the case where $t = t_1 \triangleleft a \triangleright t_2$. If $u \neq u_1 \triangleleft a \triangleright u_2$, then $\text{CP} \vdash t \neq u$. If $u = u_1 \triangleleft a \triangleright u_2$ and $\text{CP} \vdash t = u$, then $\text{CP} \vdash t_1 = u_1$ and $\text{CP} \vdash t_2 = u_2$. By induction this means $t_1 = u_1$ and $t_2 = u_2$, thus $t = u$. A similar argument can be made for $t = t_1 \triangleleft x \triangleright t_2$.

**Fact 3:** For any closed term $t \in \text{PBF}$, if $t$ is incontractible, then it contains no subterm of the form $u_1 \triangleleft a \triangleright (u_2 \triangleleft a \triangleright u_3)$ or $(u_1 \triangleleft a \triangleright u_2) \triangleleft a \triangleright u_3$.

We note that any closed term $t$ in PBF is generated by

$$t ::= T | F | t \triangleleft a \triangleright t.$$  

In [BPS13] it is shown that for any $\phi$ of this form, $se(\phi)$ can be obtained by replacing all $\triangleleft$ with $\triangleright$ and all $\triangleright$ with $\triangleleft$. Thus, if $t$ contains a subterm $u$ of the described forms, then $se(t)$ contains a subtree $X$ of the form $X_1 \triangleright a \triangleright X_2 \triangleright a \triangleright X_3$ or $(X_1 \triangleright a \triangleright X_2 \triangleright a \triangleright X_3)$ or $X \not\triangleright X_1 \triangleright X_2 \triangleright X_3$. Since $X \not\triangleright X$, $se(t) \not\triangleright se(t)$, which contradicts our initial restriction that $t$ is incontractible, thus $t$ cannot contain any $u$ of the form $u_1 \triangleleft a \triangleright (u_2 \triangleleft a \triangleright u_3)$ or $(u_1 \triangleleft a \triangleright u_2) \triangleleft a \triangleright u_3$.

**Fact 4:** Given any ground atomic substitution $\sigma$ and any variable $x$, $se(\sigma(x))$ contains both $T$ and $F$.

Let $t = \sigma(x)$ and let $\phi$ be the term over $\Sigma_{\text{SCL}}(A)$ that $t$ represents. Since $\sigma$ is atomic, $\phi$ cannot contain $T$ or $F$. Since $\sigma$ is ground, $\phi$ is closed, thus in the domain of $se$. We prove Fact 4 by induction on the structure of $\phi$. If $\phi = a$, then $se(\phi) = T \triangleright a \triangleright F$, thus contains $T$ and $F$.

If $\phi = \neg \psi$, then by the induction hypothesis $se(\psi)$ contains both $T$ and $F$. Since $se(\phi) = se(\psi)[T \mapsto F, F \mapsto T]$, $se(\phi)$ also contains both $T$ and $F$.

If $\phi = \psi_1 \land \psi_2$, then by the induction hypothesis $se(\psi_1)$ and $se(\psi_2)$ both contain $T$ and $F$. Since $se(\phi) = se(\psi_1)[T \mapsto se(\psi_2)]$ and $se(\psi_1)$ contains $T$, $se(\phi)$ contains $se(\psi_2)$ and thus contains $T$ and $F$.

**Fact 5:** Given any term $t \in \text{PBF}$, if a ground atomic substitution $\sigma$ exists such that $\sigma(t)$ is incontractible, then $t$ contains no subterm $u$ of the form $u_1 \triangleleft a \triangleright (u_2 \triangleleft a \triangleright u_3)$ or $(u_1 \triangleleft a \triangleright u_2) \triangleleft a \triangleright u_3$.

Assume $\sigma$ exists. By Fact 1 $t' \in \text{PBF}$ exists such that $\text{CP} \vdash \sigma(t) = t'$, which means $t'$ is also incontractible. By Fact 3, $t'$ does not contain a subterm of the form $u_1 \triangleleft a \triangleright (u_2 \triangleleft a \triangleright u_3)$ or $(u_1 \triangleleft a \triangleright u_2) \triangleleft a \triangleright u_3$. Suppose $t$ does contain a subterm $u$ of the described form then it must be unreachable in $\sigma(t)$, which means that $t$ must contain one or more subterms
of the form \( t_1 \triangleq x \triangleright t_2 \) such that \( se(\sigma(x)) \) contains only \( T \) or only \( F \) leaves and only \( t_2 \) or only \( t_1 \) respectively contains \( u \). However, since \( \sigma \) is ground and atomic and by Fact 4, no such \( x \) can exist, thus \( t \) cannot contain a \( u \) of the described form.

Using these facts we can prove the Lemma’s statement. By Fact 1 there exist \( t' \), \( u' \in \text{PBF} \) such that \( CP \vdash t = t' \) and \( CP \vdash u = u' \). Suppose \( CPcr(A) \vdash t = u \) and \( CP \nvdash t = u \). By Fact 5 neither \( t' \) nor \( u' \) contains a subterm of the form \( u_1 \triangleq a \triangleright (u_2 \triangleq a \triangleright u_3) \) or \( (u_1 \triangleq a \triangleright u_2) \triangleq a \triangleright u_3 \). Because of this, any proof for \( CPcr(A) \vdash t' = u' \) must use an expansion before a contraction and any expansion is undone by a contraction, undoing any application of \( CPcr1 \) or \( CPcr2 \) axioms. This means that \( CPcr(A) \nvdash t' = u' \) can be proven using only CP axioms, which means \( CP \vdash t' = u' \), which contradicts our assumption. This means that \( CPcr(A) \nvdash t' = u' \) or \( CP \vdash t' = u' \) and thus \( CPcr(A) \vdash t = u \rightarrow CP \vdash t = u \). \( \square \)

5.2 Axiomatizability in the SCL setting

**Lemma 5.2.** Consider the inductively defined family of terms

\[
P_1(x_1, x_2) = (\neg x_1 \lor x_1) \land y_1, \quad P_{n+1}(x_1, x_2) = (P_n(x_1, x_2) \lor y_{2n}) \land y_{2n+1}
\]

and

\[
Q_1 = y_1, \quad Q_{n+1} = (Q_n \lor y_{2n}) \land y_{2n+1}.
\]

For any \( n \in \mathbb{N}^+ \) and any ground substitution \( \sigma \),

\[
\sigma(a \lor P_n(x, a)) = cse \sigma(a \lor Q_n).
\]

**Proof.** By induction on \( n \). First observe that, by definition of the \( cse \) function,

\[
cse(\sigma(a \lor P_n(x, a))) = T \upmodels a \Delta Q(\sigma(P_n(x, a))) \quad \text{and} \quad cse(\sigma(a \lor Q_n)) = T \upmodels a \Delta Q(\sigma(Q_n))
\]

and thus it suffices to show that \( H_a(se(P_n(x, a))) = H_a(se(Q_n)) \). Consider the basic case of \( P_1(x, a) \) and \( Q_1 \), then

\[
H_a(se(P_1(x, a))) = H_a(se((-a \lor x) \land y_1))
\]

\[
= H_a(se(x) \upmodels a \Delta se(y_1))
\]

\[
= H_a(se(y_1))
\]

\[
= H_a(se(Q_1))
\]

For the inductive case, consider \( P_{n+1}(x, a) \) and \( Q_{n+1} \), then

\[
H_a(se(P_{n+1}(x, a))) = H_a(se((P_n(x, a) \lor y_{2n}) \land y_{2n+1}))
\]

\[
= H_a(se(P_n(x, a)) \upmodels [F \rightarrow se(y_{2n})] \upmodels [T \rightarrow se(y_{2n+1})])
\]

\[
= H_a(H_a(se(P_n(x, a) \upmodels [F \rightarrow se(y_{2n})] \upmodels [T \rightarrow se(y_{2n+1})])))
\]

\[
= H_a(H_a(se(Q_n)) \upmodels se(y_{2n}) \upmodels se(y_{2n+1}))
\]

\[
= H_a(se((Q_n \lor y_{2n}) \land y_{2n+1}))
\]

\[
= H_a(se(Q_{n+1}))
\]

\( \square \)
Note how contraction in $a \lor P_n(x, a)$ can only take place between the first $a$ and the deepest $a$ in the term and no contraction can occur in $a \lor Q_n$ before substitution. This is what will prove problematic for any finite axiomatization of CSCL($A$). Using the fact that CP axiomatizes FSCL and CP(cfr($A$)) axiomatizes CSCL($A$), we can extend Lemma 5.1 to the SCL setting.

**Lemma 5.3.** Given any two terms $\phi$ and $\psi$, if there exists a ground atomic substitution $\sigma_1$ such that $\sigma_1(\phi)$ is incontractible and there exists a ground atomic substitution $\sigma_2$ such that $\sigma_2(\psi)$ is incontractible then

$$\text{CSCL}(A) \vdash \phi \Rightarrow \text{FSCL} \vdash \phi = \psi$$

**Proof.** Let $t$ and $u$ be terms over $\Sigma_{\text{CP}}(A)$ such that $t$ defines $\phi$ and $u$ defines $\psi$. We note that $\sigma_1(\phi)$ and $\sigma_2(\psi)$ are also incontractible, since for any $\sigma$, $\sigma(\phi) =_{se} \sigma(t)$ and $\sigma(\psi) =_{se} \sigma(u)$. Assume CSCL($A$) $\vdash \phi = \psi$, then also CP(cfr($A$)) $\vdash t = u$. By Lemma 5.1 this means that CP $\vdash t = u$, and thus FSCL $\vdash \phi = \psi$. \qed

We show that the right-hand sides of the family of equalities in Lemma 5.2 fulfills the requirements of Lemma 5.3.

**Lemma 5.4.** Let $n \in \mathbb{N}^+$. For any $\psi \in SP(a \lor Q_n)$ there exists a ground atomic substitution $\sigma$ such that $\sigma(\psi)$ is incontractible.

**Proof.** Consider the ground atomic substitution

$$\sigma = [y_{3k} \mapsto b, y_{3k-1} \mapsto a, y_{3k-2} \mapsto c | k \in \mathbb{N}^+]$$

We first prove that $\sigma(Q_n)$ is incontractible. Observe that $\sigma(Q_1)$ is incontractible, since $\sigma(Q_1) = \sigma(y_1) = c$. $\sigma(Q_2)$ is also incontractible, since $\text{se}(\sigma(Q_2)) = (T \geq b \geq F) \geq c \geq ((T \geq b \geq F) \geq y) \geq a \geq F) = \text{cse}(\sigma(Q_2))$,

as illustrated in Figure 5.1. Observe that no $c$ node exists in the tree that has any direct leaves, $T$ or $F$, and that all $a$ nodes only have $F$ leaves. As such, $\sigma(Q_2) \lor c$ is still incontractible, since $\text{se}(\sigma(Q_2) \lor c) = \text{se}(\sigma(Q_2))[[F \mapsto \text{se}(c)]$. After replacement no $a$ nodes have leaves and $b$ nodes only have $T$ leaves. This means that $\sigma(Q_3) = (\sigma(Q_2) \lor c) \land a$ is also incontractible and $\text{se}(\sigma(Q_3)) = \text{se}(\sigma(Q_2))[[F \mapsto \text{se}(c)]][[T \mapsto \text{se}(a)] has no $c$ nodes with leaves and all $c$ nodes only have $F$ leaves. This can be continued, alternating $T$ and $F$ and cycling $c, a$ and $b$, to show $\sigma(Q_n)$ is always incontractible. By Lemma 2.21 this means that for every $\psi \in SP(Q_n)$ a substitution $\sigma_1$ exists such that $\sigma_1(\psi)$ is incontractible. Since the root atom of $\text{se}(\sigma(Q_n))$ is always $c$, we can conclude that for every $\chi \in SP(a \lor Q_n)$, $\sigma_1(\chi)$ is also incontractible. Lastly, since $a \lor Q_n$ does not contain $T$ or $F$ and $\sigma$ is ground and atomic, $\sigma_1$ must also be ground and atomic. \qed

Since FSCL $\not\vdash a \lor P_n(x, a) = a \lor Q_n$, the left side of the equality cannot fulfill the requirements of Lemma 5.3, but we show that any substitution preimage of the left side, that is not a renaming substitution instance of the left side, does fulfill these requirements.

**Lemma 5.5.** Let $n \in \mathbb{N}^+$. For any $\psi \in SP(a \lor P_n(x, a))$, if no renaming substitution $\sigma_r$ exists such that $\sigma_r(\psi) = a \lor P_n(x, a)$, then there exists a ground atomic substitution $\sigma$ such that $\sigma(\psi)$ is incontractible.

**Proof.** We start by observing that any $\psi \in SP(a \lor P_n(x, a))$ can be categorized into one or more of three categories:
closed under duality and symmetry, that is

Proof. Suppose no renaming substitution \( \sigma_r \) exists such that \( \sigma_r(\psi) = a \lor P_n(x, a) \).

2. A substitution \( \sigma_1 \) exists such that \( \sigma_1(\psi) = a \lor P_n(x, x_2) \).

3. A substitution \( \sigma_2 \) exists such that \( \sigma_2(\psi) = x_2 \lor P_n(x, a) \).

Suppose a \( \psi \in SP(a \lor P_n(x, a)) \) would not fit in any of these categories, then let \( \sigma_c \) be the substitution such that \( \sigma_c(\psi) = a \lor P_n(x, a) \). \( \sigma_c \) cannot be a renaming substitution, since it would fit category 1. Since \( (x_3 \lor P_n(x, x_2)) \in SP(a \lor P_n(x, x_2)) \), \( \psi \) would need to be more structurally specific than \( x_3 \lor P_n(x, x_2) \), which means a substitution \( \sigma_s \) exists such that \( \sigma_s((x_3 \lor P_n(x, x_2)) = \psi \). This substitution cannot contain \( \{x_3 \mapsto a\} \), since then \( \psi \) would fit in category 2 or 1. It can also not contain \( \{x_2 \mapsto a\} \), since then \( \psi \) would fit in category 3 or 1. The only remaining option is that \( \sigma_s \) contains \( \{y_k \mapsto \phi_1\} \), with \( \phi_1 = \neg \psi_1 \lor \chi_1 \) or \( \phi_1 = \psi_1 \land \chi_1 \). No such substitution can exist, since the number of connectives in \( \sigma_s(x_3 \lor P_n(x, x_2)) \) would be greater than in \( a \lor P_n(x, a) \), thus \( \sigma_c \) cannot exist.

Suppose no renaming substitution \( \sigma_r \) exists such that \( \sigma_r(\psi) = a \lor P_n(x, a) \). We note that a ground atomic substitution \( \sigma_3 \) exists such that \( \sigma_3(a \lor P_n(x, x_2)) \) is incontractible. An example of \( \sigma_3 \) is

\[
\sigma_3 = [x \mapsto a, x_2 \mapsto b, y_{3k-2} \mapsto c, y_{3k-1} \mapsto b, y_{3k} \mapsto a \mid k \in \mathbb{N}^+].
\]

This can be proven in the same manner as Lemma 5.4. We also note that a ground atomic substitution \( \sigma_4 \) exists such that \( \sigma_4(x_2 \lor P_n(x, a)) \). An example of \( \sigma_4 \) is

\[
\sigma_4 = [x \mapsto b, x_2 \mapsto b, y_{3k-2} \mapsto c, y_{3k-1} \mapsto a, y_{3k} \mapsto b \mid k \in \mathbb{N}^+] ,
\]

which can again be proven in the same way as Lemma 5.4. Since \( \psi \) would fit in category 2 or 3, and by Lemma 2.21, we can conclude that an atomic \( \sigma \) exists such that \( \sigma(\psi) \) is incontractible.

**Theorem 5.6.** For any finite set of atoms \( A \), with \( |A| \geq 3 \), let \( \text{EqCSCL}(A) \) be an extension of \( \text{EqFSCL} \) with \( E \) that axiomatizes \( \text{CSCL}(A) \), that is

\[ \text{EqCSCL}(A) \vdash \phi = \psi \Leftrightarrow \text{CSCL}(A) \vdash \phi = \psi, \]

then \( E \) is infinite and thus \( \text{EqCSCL}(A) \) is infinite.

**Proof.** Without loss of generality we can assume that \( \text{EqCSCL}(A) \), and thus also \( E \), is closed under duality and symmetry, that is

\[
(T = U) \in \text{EqCSCL}(A) \rightarrow (T^d = U^d) \in \text{EqCSCL}(A) \quad \text{where } P^d \text{ is the dual of } P.
\]

\[
(T = U) \in \text{EqCSCL}(A) \rightarrow (U = T) \in \text{EqCSCL}(A)
\]

24
We start by observing that, for any ground substitution \( \sigma \) and for any \( (T = U) \in E \), \( \sigma(T) =_{csc} \sigma(U) \) since CSCL(A) \( \vdash \sigma(T) = \sigma(U) \) by definition of \( E \).

**Assumption 1:** Suppose \( E \) is finite. Let \( n \in \mathbb{N}^+ \) be larger than the amount of connectives in any \( T \) for \( (T = U) \in E \). Let \( a, b, c \in A \). By Lemma 5.2, consider CSCL(A) \( \vdash a \lor P_n(x, a) = a \lor Q_n \). Observe that a substitution \( \sigma \) such that \( \sigma(a \lor P_n(x, a)) \) is contractible but \( \sigma(a \lor Q_n) \) is incontractible, for example

\[
\sigma = \{ x \mapsto b, y_{3k} \mapsto b, y_{3k-1} \mapsto a, y_{3k-2} \mapsto c \mid k \in \mathbb{N}^+ \}.
\]

By Lemma 2.20, FSCL \( \not\vdash a \lor P_n(x, a) = a \lor Q_n \).

We note that \( \sigma(P_n(x, a)) \neq_{csc} \sigma(Q_n) \) and thus CSCL(A) \( \not\vdash P_n(x, a) = Q_n \). The same is true for all pairs of left subterms of \( P_n(x, a) \) and \( Q_n \). Because of this we can assume that any equational proof for EqCSCL(A) \( \vdash a \lor P_n(x, a) = a \lor Q_n \) only uses symmetry, substitution and transitivity rules. Any proof for this equality can thus be written as a series of alternating equalities of substitutions and axioms, since EqCSCL(A) is closed under symmetry.

Let \( T_1, \ldots, T_{2m} \) be the equations that prove EqCSCL(A) \( \vdash a \lor P_n(x, a) = a \lor Q_n \). Let \( \sigma_f \) and \( \sigma_1 \) exist such that \( \sigma_f(T_1) = a \lor P_n(x, a) \) and \( \sigma_1(T_{2m}) = a \lor Q_n \). The proof then looks like

\[
\begin{align*}
a \lor P_n(x, a) &= \sigma_f(T_1) \\
&= \sigma_f(T_2) \quad (T_1 = T_2) \in \text{EqCSCL(A)} \\
&= \sigma_f(\sigma_1(T_3)) \quad T_2 = \sigma_1(T_3) \\
&= \sigma_f(\sigma_1(T_4)) \quad (T_3 = T_4) \in \text{EqCSCL(A)} \\
& \vdots \\
&= \sigma_f(\sigma_1(\sigma_2(\ldots \sigma_{m-1}(T_{2m}) \ldots ))) \\
&= \sigma_1(T_{2m}) \\
&= a \lor Q_n.
\end{align*}
\]

**Assumption 2:** Suppose \( \sigma_f \) is not a renaming substitution. Let \( \sigma_c \) be the composition of \( \sigma_1, \ldots, \sigma_{m-1} \). Observe that by the above derivation EqCSCL(A) \( \vdash T_1 = \sigma_c(T_{2m}) \) and that by Lemma 5.5 there exists an atomic substitution \( \sigma_2 \) such that \( \sigma_2(T_1) \) is incontractible, since \( \sigma_f(T_1) = a \lor P_n(x, a) \) and thus \( T_1 \in SP(a \lor P_n(x, a)) \). By Lemma 5.4 there also exists an atomic substitution \( \sigma_{c2} \) such that \( \sigma_{c2}(\sigma_c(T_{2m})) = a \lor Q_n \) and thus \( \sigma_{c}(T_{2m}) \in SP(a \lor Q_n) \). By Lemma 5.3 this means that FSCL \( \vdash T_1 = \sigma_c(T_{2m}) \), which contradicts our earlier finding that FSCL \( \not\vdash a \lor P_n(x, a) = a \lor Q_n \). This means Assumption 2 is wrong and \( \sigma_f \) must be a renaming substitution, thus the number of connectives in \( T_1 \) must equal the number of connectives in \( a \lor P_n(x, a) \). Hence the number of connectives in \( a \lor P_n(x, a) \) is larger than \( n \), which contradicts the fact that \( n \) is larger than the number of connectives of any \( T \), thus Assumption 1 must be wrong and therefore \( E \) and EqCSCL(A) must be infinite.

\[ \square \]

**Corollary 5.7.** For any countable set of atoms \( A \) with \( |A| \geq 3 \), let EqCSCL(A) be an extension of EqFSCL that axiomatizes CSCL(A), that is

\[ \text{EqCSCL}(A) \vdash \phi = \psi \iff \text{CSCL}(A) \vdash \phi = \psi, \]

then EqCSCL(A) is infinite.
Proof. Since the proof of Theorem 5.6 does not use an upper bound for \( A \), its reasoning is also valid when \( A \) is infinite.

\[ \Box \]

Corollary 5.8. For any countable set of atoms \( A \) with \(|A| \geq 3\), let \( \text{EqACSCL}(A) \) be a set of SCL equation that axiomatizes ACSCL(A), that is

\[ \text{EqACSCL}(A) \vdash \phi = \psi \iff \text{ACSCL}(A) \vdash \phi = \psi, \]

then \( \text{EqACSCL}(A) \) is infinite.

\[ \Box \]

Proof. Since the proof of Theorem 5.6 does not use the constant \( T \) or \( F \) anywhere, its reasoning is also valid for \( \text{EqACSCL}(A) \).
6.1 Conclusion

In [BPS13] Bergstra, Ponse and Staudt present SCL as a method of expressing the short-circuit evaluation that can be found in many modern-day programming languages. They also introduce CSCL and pose the open question whether their set of axiom schemes is complete. This thesis is a direct attempt at answering that question.

We have defined an atomic version of SCL and its contractive variant. While the restriction imposed by atomic SCL might be unrealistic with respect to most programming languages, it is reasonably realistic with respect to how these languages are often used. Even in AFSCSCL, which differs most with its non-atomic counterpart FSCL when compared to other variants, the addition of constants for true and false only affects the execution behaviour when used in a construct such as $x && false$ or $x || true$, which is not very common. We have shown that for some variants of SCL, such as contractive and static SCL, the removal of the constants for true and false has little to no effect, as most terms can be rewritten to terms without T or F.

We have also presented results that, in the case of $A = \{a\}$, a finite extension of the set of axiom schemes in [BPS13] exists that is complete with respect to $=_cse$. This is an interesting case because any evaluation tree can be reduced to a tree with a depth of 1 or 0. This behaviour is unique to this setting where $A = \{a\}$ and to SCLs that identify more than CSCL, such as SSCL.

Lastly we have shown that, in the case of $|A| \geq 3$, no finite axiomatization exists using only terms over $\Sigma_{SCL}(A)$, and thus no extension of the set of axiom schemes in [BPS13] exists that is complete with respect to $=_cse$. This result provides an answer to the main question of this research.

While no extensive research has been done into the applications for these results, we note that they may be used in compiler optimization for certain languages. Short-circuit evaluation has its origins in optimizing code efficiency and the concepts of CSCL might provide advancements. Consider for example the setting presented in Example 6.1. While we do not discuss an example of an actual programming language in which contraction could occur within programs, we note the similarities between the setting provided in Example 6.1 and a modern-day programming language such as Prolog.
Example 6.1. An example that demonstrates that the properties of CSCL can be exploited in a toy language setting.

- Consider the operators == and :=. An atom is of the form (Variable := Expression) or (Expression == Expression), where Variable is the name of a variable.

- An Expression is generated by the grammar

\[ e ::= n \mid \text{Variable} \mid e + e \mid e - e \mid e \ast e, \]

where \( n \) ranges over \( \mathbb{N} \), + symbolises addition, − subtraction, * multiplication.

- An atom (Variable := Expression) has the side effect of changing the value of Variable to the value yielded by the evaluation of the Expression, and yield true upon evaluation if the new value of the Variable is not equal to 0 and false otherwise. The Expression cannot contain the Variable that is being changed.

- An atom (Expression == Expression) has no side effect and yield true if both expressions yield the same value upon evaluation. Note that, since neither expression can have a side effect, the order of the expressions within the atom is not important.

6.2 Future Work

We mention some subjects for future research that might arise from this thesis. Firstly, a more thorough research into the applications of CSCL and SCL in general might prove useful. Furthermore this thesis provides axiomatizability results for \( |A| = 1 \) and \( |A| \geq 3 \). Since these results greatly differ, this raises the interesting question

Does CSCL(A) have a finite equational axiomatization in a setting where \( |A| = 2 \)?

This question is interesting since the approaches we used are unlikely to provide an answer to this question. Unlike the setting for \( A = \{a\} \), not all terms in the setting of \( A = \{a, b\} \) can be reduced to a finite set of terms, since an infinite number of incontractible terms exist, for example \( a \land b \land \ldots \land a \land b \). Terms in this setting also have the interesting property that their \( cse \) image can never contain a subtree of the form \( X / a \) (\( X / b \). Y), where \( X, Y \notin \{T, F\} \), since the root of \( X \) cannot be \( a \) nor \( b \). This suggests that no incontractible term of the form \( (a \land b \land \ldots) \lor x \) exists, thus excluding the possibility of an argument similar to that in Chapter 5.

Lastly, a possible subject for future research is the existence of a set of equational axioms over \( \Sigma_{SCL}(A) \) for RPSCL, another variant of SCL introduced in [BPS13]. We pose the conjecture that, for \( |A| \geq 3 \), a similar argument can be made as for CSCL(A), based on the family of terms

\[ \text{RPSCL}(A) \vdash a \lor P_n(x, a) = a \lor (a \lor Q_n). \]

The case of \( A = \{a\} \) would also prove interesting for RPSCL(A), but would differ significantly from CSCL, since in RPSCL(A) it is not true that any term can be reduced to one in a finite set of normal forms.
Acknowledgements

I would like to sincerely thank my thesis supervisor, Alban Ponse, for all the time he spent reviewing drafts and all the feedback he has provided. Without his helpful input I might have never found the axiomatizability results. His suggestion on notations and presentation, as well as his confidence, have proven to be invaluable during this research.
Bibliography


