

# Decision procedures for a non-Fregean logic SCI

Adrian Siwiec

August 7, 2024

Test: ążżćłóęń

## 1 Sentential logic and the Fregean axiom

## 2 SCI

### 2.1 Basic notions

**Definition 2.1** (Alphabet of SCI). The alphabet of SCI is as follows:

- $V = \{p, q, r, \dots\}$  – the countable infinite set of atomic formulas,
- $\neg$  – the unary operator “not”,
- $\rightarrow, \equiv$  – the binary operators “implication” and “identity”.

**Definition 2.2** (Language of SCI). The language of SCI is a set of formulas **FOR** defined with the grammar:

$$\text{FOR} \ni \varphi ::= p \mid \neg\varphi \mid (\varphi \rightarrow \varphi) \mid (\varphi \equiv \varphi)$$

where  $p \in V$  is an atomic formula.

From now on whenever we write  $p, q, r, \dots$  we will mean the atomic formulas. We will omit brackets when obvious. We will write  $\varphi \not\equiv \varphi$  as a shorthand for  $\neg(\varphi \equiv \varphi)$ .

The set of identities **ID** is a set of formulas  $\varphi \equiv \psi$  where  $\varphi, \psi \in \text{FOR}$ . Formulas  $\varphi \equiv \varphi$  are the trivial identities.

**Definition 2.3** (Subformulas). For a formula  $\varphi \in \text{FOR}$  let us define the set of subformulas of  $\varphi$  as:

$$\text{SUB}(\varphi) = \begin{cases} \{p\}, & \text{if } \varphi = p \in V, \\ \{\varphi\} \cup \text{SUB}(\psi), & \text{if } \varphi = \neg\psi, \\ \{\varphi\} \cup \text{SUB}(\psi) \cup \text{SUB}(\vartheta), & \text{if } \varphi = \psi \rightarrow \vartheta, \text{ or } \varphi = \psi \equiv \vartheta \end{cases}$$

By  $\varphi(\psi/\vartheta)$  we will denote the formula  $\varphi$  with all occurrences of its subformula  $\psi$  substituted with  $\vartheta$ .

**Definition 2.4** (Simple formulas). The formula  $\varphi$  is called a simple formula if it is equal to one of:

$$p, \neg p, p \equiv q, p \not\equiv q, p \equiv \neg q, p \equiv (q \rightarrow r), p \equiv (q \equiv r)$$

for  $p, q, r \in V$ .

**Definition 2.5** (Size of formula). Given a formula  $\varphi$ , let us define its size  $s(\varphi)$ :

$$s(\varphi) = \begin{cases} 1, & \text{if } \varphi = p, \\ s(\psi) + 1, & \text{if } \varphi = \neg\psi, \\ s(\psi) + s(\vartheta) + 1, & \text{if } \varphi = \psi \equiv \vartheta, \text{ or } \varphi = \psi \rightarrow \vartheta. \end{cases}$$

TODO: przepisać tak żeby to miało sens. Since  $V$  is a countable set we can assume it has a full ordering  $<$ . Let us extend it by saying that for each  $p \in V$ :  $p < “\neg” < “(” < “)” < “\rightarrow” < “\equiv”$  and then have  $\prec$  be a lexicographical ordering of formulas in **FOR** with  $<$ .

**Definition 2.6** (Axiomatization of SCI). SCI is axiomatized with the following axioms:

- (Ax1)  $\varphi \rightarrow (\psi \rightarrow \varphi)$
- (Ax2)  $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$

- (Ax3)  $(\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$
- (Ax4)  $\varphi \equiv \varphi$ ,
- (Ax5)  $(\varphi \equiv \psi) \rightarrow (\varphi \rightarrow \psi)$ ,
- (Ax6)  $(\varphi \equiv \psi) \rightarrow (\neg\varphi \equiv \neg\psi)$ ,
- (Ax7)  $(\varphi \equiv \psi) \rightarrow ((\chi \equiv \theta) \rightarrow ((\varphi \rightarrow \chi) \equiv (\psi \rightarrow \theta)))$ ,
- (Ax8)  $(\varphi \equiv \psi) \rightarrow ((\chi \equiv \theta) \rightarrow ((\varphi \equiv \chi) \equiv (\psi \equiv \theta)))$ .

The only inference rule is the *modus ponens* rule:

$$MP : \frac{\varphi, \varphi \rightarrow \psi}{\psi}$$

**Definition 2.7** (A theorem). A formula  $\varphi$  is a *theorem* if there exists a finite set of formulas  $\varphi_1, \dots, \varphi_n$  ( $n \geq 1$ ), such that  $\varphi = \varphi_n$ , and for all  $i \in \{1, \dots, n\}$  the formula  $\varphi_i$  is either an axiom of SCI, or it is inferred from formulas  $\varphi_j, \varphi_k$  ( $j, k < i$ ) via the *modus ponens* rule. We'll mark it as  $\vdash \varphi$  and say that it is *provable*.

Let us give an example of a provable formula with its full proof:  
TODO

**Definition 2.8** (Decidability). A logic is *decidable* if there exists an effective method to determine whether a given formula of this logic is a theorem.

Let us now give some semantic definitions:

**Definition 2.9** (Model). An SCI model is a structure  $M = (U, D, \neg, \rightarrow, \equiv)$  where:

- $U \neq \emptyset$  is the *universe* of  $M$ ,
- $\emptyset \neq D \subsetneq U$  is the *set of designated values*,
- $\neg : U \rightarrow U$  is the algebraic counterpart of the operator not, such that  $\forall a \in U$ :  $\neg a \in D$  iff  $a \notin D$ ,
- $\rightarrow : U \times U \rightarrow U$  is the algebraic counterpart of the operator of implication, such that  $\forall a, b \in U$ :  $a \rightarrow b \in D$  iff  $a \notin D$  or  $b \in D$ ,
- $\equiv : U \times U \rightarrow U$  is the algebraic counterpart of the operator of identity, such that  $\forall a, b \in U$ :  $a \equiv b$  iff  $a = b$  (that is, iff  $a$  and  $b$  denote the same element of the universe  $U$ ).

**Definition 2.10** (Valuation). Given a model  $M = (U, D, \neg, \rightarrow, \equiv)$ , a valuation in this model is a function  $V : \text{FOR} \rightarrow U$ , such that for all  $\varphi, \psi \in \text{FOR}$ :

- $V(\neg\varphi) = \neg V(\varphi)$
- $V(\varphi \rightarrow \psi) = V(\varphi) \rightarrow V(\psi)$
- $V(\varphi \equiv \psi) = V(\varphi) \equiv V(\psi)$

If  $V(\varphi) = a$  we will call  $a$  the *denotation* of  $\varphi$ .

**Definition 2.11** (Satisfiability of a formula). Given a model  $M = (U, D)(U, D, \neg, \rightarrow, \equiv)$  and a valuation  $V$ , a formula  $\varphi$  is *satisfied in this valuation* iff its valuation belongs to  $D$ . If a formula  $\varphi$  is satisfied we will mark it by  $M, V \models \varphi$ .

**Definition 2.12** (Truth of a formula). Given a model  $M = (U, D)(U, D, \neg, \rightarrow, \equiv)$ , a formula is *true in this model* iff it is satisfied in all valuations of this model. We'll mark it by  $M \models \varphi$ .

**Definition 2.13** (Validity of a formula). A formula is *valid* iff it is true in all models. We'll mark it by  $\models \varphi$ .

Let us see some examples: TODO

**Theorem 1** (Correctness and fullness of SCI). *For any formula  $\varphi$  the following two are equivalent:*

- $\varphi$  is *provable*,
- $\varphi$  is *valid*.

The proof of the correctness and fullness theorem can be found in TODO(Suszko Bloom Th 1.9).

**Theorem 2** (Decidability of SCI). *SCI is decidable.*

The proof of the decidability theorem can be found in TODO(Suszko Bloom Corollary 2.4)

---

**Definition 2.14** (Complexity of a set of formulas). Given a set of formulas  $\Phi$ , let us define:

$$c(\Phi) = \sum_{\phi \in \Phi} 3^{s(\phi)}$$

Let us assume that we are given a full ordering of formulas  $\prec$ , such that:

- if  $\varphi \neq \psi$ , then  $\varphi \prec \psi$  or  $\psi \prec \varphi$ ,
- if  $s(\varphi) < s(\psi)$ , then  $\varphi \prec \psi$ ,
- if  $\varphi \prec \psi$ , then  $\varphi \equiv \psi \prec \psi \equiv \varphi$ .

## 2.2 Axiomatization of SCI

The axioms of SCI are as follows:

The only inference rule is the rule MP:

$$\frac{\varphi \rightarrow \psi, \varphi}{\psi} \quad (\text{MP})$$

## 3 Satisfiability in SCI

### 3.1 SCI is decidable

### 3.2 Decision procedures

#### 3.2.1 $T$ decision procedure

#### 3.2.2 $T^*$ decision procedure

The  $T^*$  decision procedure has two main groups of deduction rules, see figures ?? and ??.

The rules are

TODO: differences compared to “Dedukcyjne Dylematy” (DD):

- Phase 1 rules replace formulas, phase 2 rules only add formulas.
- No need to say: “w wyniku zastosowania reguły co najmniej jedna z konkluzji zawiera formułę, która nie występuje w  $X \cup \{\varphi\}$ ”. In the first phase simply apply all the rules as long as you can. In second phase apply closure rules only once to any given formula.
- There are TWO *fun* rules: one for each phase. Phase 1 fun rule replaces formulas, phase 2 fun rule only adds formulas (leaves the original unchanged as well). If *fun* is replacing in phase 2 then e.g. after rule  $\equiv_T^-$  the rule *fun* is applied removing one of the conclusion formulas. Besides, the examples in DD seem to assume the *fun* works this way anyways.

#### 3.2.3 Correctness and completeness of $T^*$

**Theorem 3.**  *$T^*$  is correct.*

*Proof.* TODO: this is just a sketch, rewrite. We want to show that if  $T^*$  says the formula is closed, it is indeed closed.

1. All the rules of  $T^*$  are strongly correct. TODO at least the non-obvious ones.
2. We're given a closed tree.
3. Going up the tree, on every step we have a closed set of formulas, because every rule is strongly correct.
4. The root is closed.

□

$$(\neg) \quad \frac{\Phi \cup \{\neg\neg\varphi\}}{\Phi \cup \{\varphi\}} \quad (\rightarrow) \quad \frac{\Phi \cup \{\varphi \rightarrow \psi\}}{\Phi \cup \{\neg\varphi\} | \Phi \cup \{\psi\}} \quad (\neg \rightarrow) \quad \frac{\Phi \cup \{\neg(\varphi \rightarrow \psi)\}}{\Phi \cup \{\varphi, \neg\psi\}}$$

(a) Decomposition rules

$$(\text{sym}_1^1) \quad \frac{\Phi \cup \{\varphi \equiv \psi\}}{\Phi \cup \{\psi \equiv \varphi\}} \text{ if } \varphi > \psi \quad (\text{sym}_2^1) \quad \frac{\Phi \cup \{\varphi \not\equiv \psi\}}{\Phi \cup \{\psi \not\equiv \varphi\}} \text{ if } \varphi > \psi$$

$$(\text{fun}^1) \quad \frac{\Phi \cup \Phi(\varphi) \cup \{p \equiv \varphi\}}{\Phi \cup \Phi(\varphi/p) \cup \{p \equiv \varphi\}} \text{ if } p < \varphi$$

(b) Reduction rules

$$(\neq_1) \quad \frac{\Phi \cup \{p \not\equiv \psi\}}{\Phi \cup \{p \not\equiv \alpha, \alpha \equiv \psi\}} \quad (\neq) \quad \frac{\Phi \cup \{\varphi \not\equiv \psi\}}{\Phi \cup \{\alpha \equiv \varphi, \beta \equiv \psi, \alpha \not\equiv \beta\}}$$

$$(\neg) \quad \frac{\Phi \cup \{p \equiv \neg\psi\}}{\Phi \cup \{p \equiv \neg\alpha, \alpha \equiv \psi\}} \quad (\rightarrow) \quad \frac{\Phi \cup \{p \equiv (\varphi \rightarrow \psi)\}}{\Phi \cup \{p \equiv (\alpha \rightarrow \beta), \alpha \equiv \varphi, \beta \equiv \psi\}}$$

$$(\rightarrow) \quad \frac{\Phi \cup \{p \equiv (q \rightarrow \psi)\}}{\Phi \cup \{p \equiv (q \rightarrow \alpha), \alpha \equiv \psi\}} \quad (r_{\rightarrow}) \quad \frac{\Phi \cup \{p \equiv (\varphi \rightarrow q)\}}{\Phi \cup \{p \equiv (\alpha \rightarrow q), \alpha \equiv \varphi\}}$$

$$(\equiv) \quad \frac{\Phi \cup \{p \equiv (q \equiv \psi)\}}{\Phi \cup \{p \equiv (q \equiv \alpha), \alpha \equiv \psi\}} \quad (r_{\equiv}) \quad \frac{\Phi \cup \{p \equiv (\varphi \equiv q)\}}{\Phi \cup \{p \equiv (\alpha \equiv q), \alpha \equiv \varphi\}}$$

$$(\equiv) \quad \frac{\Phi \cup \{p \equiv (\varphi \equiv \psi)\}}{\Phi \cup \{p \equiv (\alpha \equiv \beta), \alpha \equiv \varphi, \beta \equiv \psi\}} \quad (\equiv) \quad \frac{\Phi \cup \{\varphi \equiv \psi\}}{\Phi \cup \{\alpha \equiv \beta, \alpha \equiv \varphi, \beta \equiv \psi\}}$$

(c) Identity rules. Note: in all identity rules the highlighted formula is not simple (i.e.  $\varphi$  and  $\psi$  are not variables)

Figure 1: Deduction rules of the first phase of  $T^*$ .

$$(\text{sym}_1^2) \quad \frac{\Phi \cup \{\varphi \equiv \psi\}}{\Phi \cup \{\psi \equiv \varphi, \varphi \equiv \psi\}} \text{ if } \varphi > \psi \quad (\text{sym}_2^2) \quad \frac{\Phi \cup \{\varphi \not\equiv \psi\}}{\Phi \cup \{\psi \not\equiv \varphi, \varphi \not\equiv \psi\}} \text{ if } \varphi > \psi$$

$$(\text{fun}^2) \quad \frac{\Phi \cup \Phi(\varphi) \cup \{p \equiv \varphi\}}{\Phi \cup \Phi(\varphi/p) \cup \Phi(\varphi) \cup \{p \equiv \varphi\}} \text{ if } p < \varphi$$

$$(\top) \quad \frac{\Phi \cup \{p \equiv q\}}{\Phi \cup \{p, q, p \equiv q\} | \Phi \cup \{\neg p, \neg q, p \equiv q\}} \quad (\neg \top) \quad \frac{\Phi \cup \{p \equiv \neg q\}}{\Phi \cup \{p, \neg q, p \equiv \neg q\} | \Phi \cup \{\neg p, q, p \equiv \neg q\}}$$

$$(\neg \neg) \quad \frac{\Phi \cup \{p \equiv (q \rightarrow r)\}}{\Phi \cup \{p, \neg q, \theta\} | \Phi \cup \{p, r, \theta\} | \Phi \cup \{\neg p, q, \neg r, \theta\}}, \text{ where } \theta = (p \equiv (q \rightarrow r))$$

$$(\equiv \equiv) \quad \frac{\Phi \cup \{p \equiv (q \equiv r)\}}{\Phi \cup \Phi_1 | \Phi \cup \Phi_2 | \Phi \cup \Phi_3 | \Phi \cup \Phi_4 | \Phi \cup \Phi_5 | \Phi \cup \Phi_6}, \text{ where}$$

$$\begin{aligned} \Phi_1 &= \{p, q, r, q \equiv r, p \equiv (q \equiv r)\} \\ \Phi_2 &= \{p, \neg q, \neg r, q \equiv r, p \equiv (q \equiv r)\} \\ \Phi_3 &= \{\neg p, q, r, q \not\equiv r, p \equiv (q \equiv r)\} \\ \Phi_4 &= \{\neg p, q, \neg r, q \not\equiv r, p \equiv (q \equiv r)\} \\ \Phi_5 &= \{\neg p, \neg q, r, q \not\equiv r, p \equiv (q \equiv r)\} \\ \Phi_6 &= \{\neg p, \neg q, \neg r, q \not\equiv r, p \equiv (q \equiv r)\} \end{aligned}$$

$$(\perp) \quad \frac{\Phi \cup \{p \not\equiv q\}}{\Phi \cup \{p, q, \theta\} | \Phi \cup \{p, \neg q, \theta\} | \Phi \cup \{\neg p, q, \theta\} | \Phi \cup \{\neg p, \neg q, \theta\}}, \text{ where } \theta = (p \not\equiv q)$$

Figure 2: Deduction rules of the second phase of  $T^*$ .

**Theorem 4.**  $T^*$  is complete.

We want to show, that if the formula is closed, the tree produced by  $T^*$  is closed, or by contraposition, that if the tree produced by  $T^*$  is open, the given formula is open.

First, let us show that the first phase finishes.

**Theorem 5.** The first phase of  $T^*$  finishes and produces finite trees.

*Proof.* Let us first notice that all decomposition and identity rules of the first phase reduce the complexity of the formulas.

**Lemma 6.** If rule  $R$  is a decomposition rule or an identity rule,  $\Phi$  is its premise and  $\Phi_1, \dots, \Phi_n$  are its conclusions, then  $c(\Phi) > c(\Phi_i)$ , for  $1 \leq i \leq n$ .

*Proof.* TODO: Przerachować. Chyba здаје się działać, ewentualnie trochę podkreślić  $s$  lub  $c$ .  $\square$

**Lemma 7.** If rule  $R$  is a reduction rule  $\Phi$  is its premise and  $\Phi_1$  is its conclusion, then  $c(\Phi) \geq c(\Phi_1)$ .

*Proof.* Trivial. All the reduction rule replace a formula with another formula that is smaller in our given order  $\prec$ , so they will never increase the complexity function.  $\square$

**Lemma 8.** The reduction rules cannot be applied indefinitely without lowering the complexity function.

*Proof.* TODO: przepisać. To jest trywialne.

Based on lemma ??, applying the reduction rule will never increase the complexity function. If applying the rule will give conclusions with the same complexity, the conclusion will contain some formulas that are lower in  $\prec$  than the appropriate formulas in the premise, while all the other formulas are equal.  $\square$

As a consequence of lemmas ??, ?? and ?? it is clear that, for each node of the produced tree, after a finite number of rules applied, either a closed node or a node with lower complexity will be reached. Since the complexity of the node is always positive and finite it is impossible that the complexity will be lowered indefinitely, so the first phase of the  $T^*$  procedure will finish.  $\square$

**Theorem 9.** All leaves produced by the first phase are either closed or contain only simple formulas.

*Proof.* Simple. Assume there is a non-simple formula in an open leaf. One of decomposition, reduction or equality rules would apply to it, which is a contradiction with the fact that the first phase finished.  $\square$

**Theorem 10.** Given a set of simple formulas, phase 2 finishes.

*Proof.* TODO: trywialne. Wszystkie formuły  $z \equiv w$  w nazwie wykonają się co najwyżej raz, po każdej z nich  $\text{sym}^*$  i  $\text{fun}$  też co najwyżej raz.  $\square$

Let us call a tree that is produced by the  $T^*$  procedure a  $T^*$  tree.

**Theorem 11.** Open leaf  $X$  of a  $T^*$  tree satisfies the following conditions:

- (X1)  $X$  is a set of simple formulas.
- (X2) If  $\varphi \in \text{FOR}^X$  then  $\varphi$  is either a simple formula or is equal to  $p \rightarrow q$ , for some variables  $p, q$ .
- (X3) If  $\varphi \equiv \psi \in \text{FOR}^X$ , then  $\varphi \leq \psi$  or  $\psi \equiv \varphi \in \text{FOR}^X$ .
- (X4) If  $\varphi \not\equiv \psi \in \text{FOR}^X$ , then  $\varphi < \psi$  or  $\psi \not\equiv \varphi \in \text{FOR}^X$ .
- (X5) If  $p \equiv q \in X$ , then  $p, q \in X$ , or  $\neg p, \neg q \in X$ .
- (X6) If  $p \equiv \neg q \in X$ , then  $p, \neg q \in X$ , or  $\neg p, q \in X$ .
- (X7) If  $p \equiv (q \rightarrow r) \in X$ , then  $p, \neg q \in X$ , or  $p, r \in X$ , or  $\neg p, q, \neg r \in X$ .
- (X8) If  $p \equiv (q \equiv r) \in X$ , then  $\Phi_i \subseteq X$ , for  $i \in \{1, \dots, 6\}$ , where  $\Phi_i$  are defined in figure ??.

*Proof.* (X1) Phase 1 ends with nodes containing only simple formulas. Phase 2 doesn't add any non-simple formulas.

(X2), ... (X8) Trivial, directly following phase 2 rules.  $\square$

**Lemma 12** (Lemat 5 z DD). *Let  $X$  be an open leaf of a  $T^*$  tree and let  $\neg\varphi \in \text{FOR}^X$ . Then:*

$$\varphi \in X \cup \text{ID}^X \text{ iff } \neg\varphi \notin X \cup \text{ID}^X.$$

*Proof.* Because of  $X2$ , if  $\neg\varphi \in \text{FOR}^X$ , then  $\varphi = p$  or  $\varphi = (p \equiv q)$  for some variables  $p, q$ .

1°  $\varphi = p$ . ... (Jak w DD.)

2°  $\varphi = (p \equiv q)$ .

Since  $\neg\varphi = p \not\equiv q \in \text{FOR}^X$ , we have that  $p \not\equiv q \in X$ , because  $p \not\equiv q$  cannot be a subformula of a simple formula. From  $p \not\equiv q \in X$  it follows that  $p \neq q$  and  $p \equiv q \notin X$ , otherwise  $X$  would be an axiom set and therefore closed.

So, we have:  $p \not\equiv q \in X$  (so,  $\neg\varphi \in X \subseteq X \cup \text{ID}^X$ ). We need to show that  $\varphi \notin X \cup \text{ID}^X$ , i.e.  $p \equiv q \notin X \cup \text{ID}^X$ . We have that  $p \equiv q \notin X$ , so all we need to show is that  $p \equiv q \notin \text{ID}^X$ .

Let us assume otherwise, that  $p \equiv q \in \text{ID}^X$ .

Since  $p \equiv q \notin X$ , we have that  $p \equiv q \notin \text{ID}_0^X$ .  $p \equiv q \notin \text{ID}_n^X$  for  $n > 0$  based on the definition of  $\text{ID}_n^X$ . So, for the  $p \equiv q$  to be in  $\text{ID}^X$  we need to have formulas  $\theta_1, \dots, \theta_k$ , for some  $k \geq 1$ , such that:  $p \equiv \theta_1 \in X$ , or  $\theta_1 \equiv p \in X$ , ...,  $\theta_k \equiv q$ , or  $q \equiv \theta_k$ .

2.1°  $k = 1$ .

TODO: przepisać żeby było jaśniej.

We have that  $p \equiv \theta_1 \in X$ , or  $\theta_1 \equiv p \in X$  and  $\theta_1 \equiv q \in X$ , or  $q \equiv \theta_1 \in X$ . One of the formulas in  $\{p, q, \theta_1\}$  will be smallest in the ordering  $\prec$ . If the formula  $p$  is the smallest, the **fun** rule would produce  $q \equiv p \in X$  (or  $p \equiv q \in X$ ) from  $q \equiv \theta_1 \in X$  (or  $\theta_1 \equiv q \in X$ ) by replacing  $\theta_1$  by  $p$ . In any case,  $p \equiv q$  would be in  $X$  (possibly after  $\text{sym}_1^2$ ). Thus  $X$  would be an axiom set ( $p \not\equiv q, p \equiv q \in X$ ), a contradiction. A contradiction happens in a similar way if the formula  $q$  is the smallest of the three. If the formula  $\theta_1$  is the smallest one, from  $p \not\equiv q \in X$  will be produced either  $\theta_1 \not\equiv q \in X$  or  $p \not\equiv \theta_1 \in X$  (by **fun** similarly to above), giving another contradiction.

2.2°  $k > 1$ .

???

□

## 4 Implementation

### 4.1 Data structures

### 4.2 Optimizations

### 4.3 Size of produced proof trees

### 4.4 Time complexity

### 4.5 Real time