Implementation of decision procedures for non-Fregean logic SCI

Adrian Siwiec

May 5, 2024

Test: ążźćłóęń

1 Sentential logic and the Fregean axiom

2 SCI

TODO: Just some free notes here for now, so that I don't forget later.

Whenever we wite $\varphi \not\equiv \psi$ we mean $\neg(\varphi \equiv \psi)$

Definition 2.1 (Size of formula)**.** Given a formula φ, let us define:

$$s(\varphi) = \begin{cases} 1, & \text{if } \varphi = p, \\ c(\psi) + 1, & \text{if } \exists \psi \in \texttt{FOR, such that } \varphi = \neg\psi, \\ c(\psi) + c(\vartheta) + 1, & \text{if } \exists \psi, \vartheta \in \texttt{FOR, such that } \varphi = \psi \equiv \vartheta, \text{ or } \varphi = \psi \rightarrow \vartheta. \end{cases}$$

Definition 2.2 (Complexity of a set of formulas)**.** Given a set of formulas Φ, 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 $\varphi \equiv \psi \prec \psi \equiv \varphi$.

2.1 ...

2.2 Axiomatization of SCI

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 1 and 2.

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 unchaged as well). If fun is replacing in phase 2 then e.g. after rule $\equiv_T^{\rightarrow}$ 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.

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

(a) Decomposition rules

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

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

(b) Reduction rules

$$(\not\equiv_1)\quad \frac{\Phi \cup \{p \not\equiv \psi\}}{\Phi \cup \{p \not\equiv \alpha, \alpha \equiv \psi\}} \qquad\qquad (\not\equiv_2)\quad \frac{\Phi \cup \{\varphi \not\equiv \psi\}}{\Phi \cup \{\alpha \equiv \varphi, \beta \equiv \psi, \alpha \not\equiv \beta\}}$$

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

$$(\equiv_{\rightarrow}^l)\quad \frac{\Phi \cup \{p \equiv (q \rightarrow \psi)\}}{\Phi \cup \{p \equiv (q \rightarrow \alpha), \alpha \equiv \psi\}} \qquad (\equiv_{\rightarrow}^r)\quad \frac{\Phi \cup \{p \equiv (\varphi \rightarrow q)\}}{\Phi \cup \{p \equiv (\alpha \rightarrow q), \alpha \equiv \varphi\}}$$

$$(\equiv_{\equiv}^l)\quad \frac{\Phi \cup \{p \equiv (q \equiv \psi)\}}{\Phi \cup \{p \equiv (q \equiv \alpha), \alpha \equiv \psi\}} \qquad (\equiv_{\equiv}^r)\quad \frac{\Phi \cup \{p \equiv (\varphi \equiv q)\}}{\Phi \cup \{p \equiv (\alpha \equiv q), \alpha \equiv \varphi\}}$$

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

(c) Identity rules. Note: in all identity rules the highlighted formula is not simple (i.e. φ and ψ are not variables)

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

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

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

$$(\equiv_{\top})\quad \frac{\Phi \cup \{p \equiv q\}}{\Phi \cup \{p, q, p \equiv q\} \mid \Phi \cup \{\neg p, \neg q, p \equiv q\}} \qquad (\equiv_{\top}^{\neg})\quad \frac{\Phi \cup \{p \equiv \neg q\}}{\Phi \cup \{p, \neg q, p \equiv \neg q\} \mid \Phi \cup \{\neg p, q, p \equiv \neg q\}}$$

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

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

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

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

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

3.2.3 Correctness and completeness of T^*

Theorem 1. 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.

$\square$

Theorem 2. 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 3. *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 4. *If rule R is a decomposition rule or an identity rule, Φ is its premise and $\Phi_1, ... \Phi_n$ are its conclusions, then $c(\Phi) > c(\Phi_i)$, for $1 \leq i \leq n$.*

Proof. TODO: Przerachować. Chyba zdaje się działać, ewentualnie trochę podkręcić s lub c. $\square$

Lemma 5. *If rule R is a reduction rule Φ is its premise and Φ_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 6. *The reduction rules cannot be applied indefinitely without lowering the complexity function.*

Proof. TODO: przepisać. To jest trywialne.

Based on lemma lemma 5, 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 4, 5 and 6 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 7. *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 8. *Given a set of simple formulas, phase 2 finishes.*

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

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

Theorem 9. *Open leaf X of a T^* tree satisfies the following conditions:*

(X1) X is a set of simple formulas.

(X2) If $\varphi \in \textbf{FOR}^X$ then φ is either a simple formula or is equal to $p \to q$, for some variables p, q.

(X3) If $\varphi \equiv \psi \in \mathit{FOR}^X$, then $\varphi \leq \psi$ or $\psi \equiv \varphi \in \mathit{FOR}^X$.

(X4) If $\varphi \not\equiv \psi \in \mathit{FOR}^X$, then $\varphi < \psi$ or $\psi \not\equiv \varphi \in \mathit{FOR}^X$.

(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 $p \equiv (q \to r) \in X$, 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 $i \in \{1, ..., 6\}$, where Φ_i are defined in figure 2.

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. $\qquad\square$

Lemma 10 (Lemat 5 z DD)**.** *Let X be an open leaf of a T^* tree and let $\neg\varphi \in \mathit{FOR}^X$. Then:*

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

Proof. Because of $X2$, if $\neg\varphi \in \mathit{FOR}^X$, 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 $\neg\varphi = p \not\equiv q \in \mathit{FOR}^X$, 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, $\neg\varphi \in X \subseteq X \cup \mathit{ID}^X$). We need to show that $\varphi \notin X \cup \mathit{ID}^X$, i.e. $p \equiv q \notin X \cup \mathit{ID}^X$. We have that $p \equiv q \notin X$, so all we need to show is that $p \equiv q \notin \mathit{ID}^X$.

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

Since $p \equiv q \notin X$, we have that $p \equiv q \notin \mathit{ID}_0^X$. $p \equiv q \notin \mathit{ID}_n^X$ for $n > 0$ based on the definition of ID_n^X. So, for the $p \equiv q$ to be in ID^X we need to have formulas $\theta_1, ..., \theta_k$, 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 θ_1 by p. In any case, $p \equiv q$ would be in X (possibly after $\mathtt{sym}_1^2$). 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 θ_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.

$2.2°$ $k > 1$.
???

$\qquad\square$

4 Implementation

4.1 Data structures

4.2 Optimizations

4.3 Size of produced proof trees

4.4 Time complexity

4.5 Real time