Andrei Kouznetsov

DEDUCTION CHAINS AND DC-LIKE DECISION PROCEDURE FOR GUARDED LOGIC

Abstract

The notion of Deduction Chain (DC) was suggested by K.Schütte to prove the completeness of first order logic (FOL). There is a question whether it is possible to provide the decision procedure for the guarded fragment of FOL (GF) on the basis of DC-rules. We propose here the decision procedure, or DCL algorithm, for the GF of FOL which rules are close to the ones of DC. DCL algorithm can be viewed as a "mirror" of the tableau algorithm with the use of the so called blocking technique suggested in [4].

1. Guarded Fragment

The guarded fragment (GF) represents the special fragment of first-order logic (FOL), it was introduced by Andrèka, van Benthem, and Németi [1.1], [1.2], and the main ideas go back to Németi [5]. Let σ be a first-order relational structure (in which FO-formulas may contain symbols of arbitrary arity and constant symbols, but no function symbols of positive arity). The definition of the formula in GF is given as follows:

- $Rx \in GF$ for $R \in \sigma$;
- $GF$ is closed under Booleans;
- if $\phi(x, y) \in GF$ and $\alpha(x, y)$ is an atom with $\text{free}(\phi) \subseteq \text{free}(\alpha)$ then $\forall y(\alpha(x, y) \supset \phi(x, y)) \in GF$ and $\exists y(\alpha(x, y) \land \phi(x, y)) \in GF$, where $x, y$ are tuples of variables and $Rx$ is a relation defined on $x$. 
In particular, if $x$ and $y$ are the two variables, the atom $\alpha(x, y)$ can be understood as a relation of accessibility: the world $y$ can be reached from the worlds $x$. So the above quantification allows to consider the guarded fragment as a generalization of various modal, temporal and descriptional logics. On the other hand, many good properties of modal logics are preserved in such systems, namely, decidability, finite model property, invariance under an appropriate variant of bisimulation. The satisfiability problems for GF is complete for 2EXPTIME ([2]).

There are the weaker modifications of the simple GF, which are obtained by relaxing the rules for the quantifiers, that are the loosely GF (LGF) and the clique GF (CGF), there are also the guarded fixed point logics $\mu$GF, $\mu$LGF and $\mu$CGF which represent the extensions of GF, LGF and CGF, respectively ([3]).

2. Deduction Chains

The notion of a deduction chain (DC) was introduced by Schuette [6]. Below is the definition of DC for the sequences of formulas. Let $\Phi$ be a sequence of formulas $\varphi_0, \varphi_1, \varphi_2, \ldots$. Each formula $\varphi_1$ presented in the negation normal form (NNF). A sequence $\Psi_i$ is an axiom if it represents a sequence $\Psi_i, B, \neg B$, where $B$ is an atomic formula.

A deduction chain (DC) for $\Phi$ is a sequence $\Phi_0, \Phi_1, \Phi_2, \ldots$, where (1) the initial sequence $\Phi_0$ is a sequence $\Phi$; (2) if the sequence $\Phi_n$ of DC contains an axiom or only non-reducible formulas (the formulas to which none of the rules (1)-(3) can be applied), then $\Phi_n$ is the last formula of DC. In this case we say that DC has the length $n$; (3) if the sequence $\Phi_n$ does not contain an axiom and contains a reducible formula, then $\Phi_n$ has an immediate successor $\Phi_{n+1}$; we define $\Phi_{n+1}$ in the following way. Let the sequent $\Phi_n$ be of the form $\Psi_n, A, B_1, \ldots, B_m$, where the formulas $B_1, \ldots, B_m$ are atomic, and the formula $A$ is non-atomic. We will call $A$ the marked formula for $\Phi_n$. Then the sequence $\Phi_{n+1}$ is defined as follows.

- $A$ is $C_0 \lor C_1$. Then $\Phi_{n+1}$ has a form: $\Psi_n, C_0, C_1, B_1, \ldots, B_m$;
- $A$ is $C_0 \land C_1$. Then $\Phi_{n+1}$ is whether $\Psi_n, C_0, B_1, \ldots, B_m$ or $\Psi_n, C_1, B_1, \ldots, B_m$;
- $A$ is $\exists x C(x)$. Then $\Phi_{n+1}$ has a form: $\exists x C(x), \Psi_n, C(t_i), B_1, \ldots, B_m$, where $t_i$ is the term with minimal $i$ such that $C(t_i)$ does not occur in any of the sequences $\Phi_0, \ldots, \Phi_n$;
A is \( \forall x C(x) \). Then \( \Phi_{n+1} \) has a form: \( \Psi_n, C(u_i), B_1, ..., B_m, \) where the free variable \( u_i \) is minimal such that \( u_i \) does not occur in \( \Phi_n \).

3. Deduction Chains for the Guarded Fragment (GDC)

One can also define the notion of the deduction chain for the (GF) \((\text{guarded deduction chain} - \text{GDC})\) for the arbitrary guarded fragment. Let \( x, y \) be the tuples of variables, then the rules for the quantifiers are:

- \( A \) is \( \exists y (\alpha(x, y) \land \phi(x, y)) \). Then \( \Phi_{n+1} \) has a form: \( \exists y (\alpha(x, y) \land \phi(x, y)), \Psi_n, \alpha(x, t_i) \land \phi(x, t_i), B_1, ..., B_m, \) where \( t_i \) is the tuple of terms (tuples of terms are linearly ordered) with minimal \( i \) such that \( \alpha(x, t_i) \land \phi(x, t_i) \) does not occur in any of the sequences \( \Phi_0, ..., \Phi_n \).

- \( A \) is \( \forall y (\alpha(x, y) \supset \phi(x, y)) \). Then \( \Phi_{n+1} \) has a form: \( \Psi_n, \neg \alpha(x, u_i), \phi(x, u_i), B_1, ..., B_m, \) where \( i \) is minimal such that there is no tuple of free variables \( u_i \) is in \( \Phi_n \).

4. Completeness

Using DC one can prove the completeness theorem for the FOL. Let’s apply Schütte-styled completeness proof to the guarded fragment. Firstly one has to prove the two lemmas.

**Principal Syntactic Lemma.** If every GDC of a sequence \( \Phi \) (each formula of which is in GF and in NNF) contains an axiom, then a sequence \( \Phi \) is provable.

The proof is the same as the one in [6].

**Principal Semantical Lemma.** Let \( \Phi \) be the sequence \( A_1, ..., A_m \) (each formula \( A_i \) is in GF and in NNF). If there exists GDC for \( \Phi \) which contain no axiom, then there is a model structure \( M \) and a variable \( \alpha \) in \( |M| \) such that \( |A_1 \lor ... \lor A_m|_{M, \alpha} = f \).

**Proof.** Consider GDC, which is obtained by the application of the above rules, \( \Phi_0, \Phi_1, \Phi_2, ..., \) which contains no axiom, and \( \Phi \) is \( \Phi_0 \). Define a set \( K := \text{set}(\Phi_0) \cup \text{set}(\Phi_1) \cup \text{set}(\Phi_2) \cup ... \)
Consider the properties of GDC with respect to the set K.

- if some atomic formula is contained in a sequence \( \Phi_n \) then it’s necessarily contained in \( \Phi_k, k \geq n \);
- if some non-atomic formula is contained in a sequence \( \Phi_n \) then it is contained in \( \Phi_{n+1} \) as marked;
- if a formula \( \exists y (\alpha(x, y) \land \phi(x, y)) \) is an element of K, then all formulas \( \alpha(x, t_0) \land \phi(x, t_0), \alpha(x, t_1) \land \phi(x, t_1), \alpha(x, t_2) \land \phi(x, t_2), ... \) are also contained in K.
- there is no atomic formula B such that both B and \( \neg B \) are the elements of K.

Define a model structure \( M \) as follows:

- The universe \( |M| \) of \( M \) - a set of all terms;
- if \( R \) - n-ary symbol for relations, then \( R^\mu : |\mu|^n \to \{t, f\} \), then let us have \( ||R^\mu(t_1, ..., t_n)||_{M, \alpha} = \{t, if \neg R^\mu(t_1, ..., t_n) \in K \text{ and } f, if \neg R^\mu(t_1, ..., t_n) \notin K \} \)
- if B is in positive atomic form, then \( ||B||_{M, \alpha} = t \) exactly when \( \neg B \) is an element of K, thus \( B \in K \Rightarrow ||B||_{M, \alpha} = f \).

Since \( \{A_1, ..., A_m\} = set(\Phi) \subset K \), then \( ||A_i||_{M, \alpha} = f \) for \( 1 \leq i \leq m \). This gives \( ||A_1 \lor ... \lor A_m||_{M, \alpha} = f \). 

**Theorem 1 (Completeness).** If a sequence \( \Phi \) is valid, then it is deducible.

**Proof.** Let \( \Phi \) be a valid sequence. By the principal semantic lemma every GDC of \( \Phi \) contains an axiom. From the principal syntactic lemma it follows that \( \Phi \) is deducible. \( \square \)

The given completeness proof of the GF essentially does not differ from the one for the full FOL. The model structure \( M \) which is used in the proof of principal semantical lemma is not finite, so there is the question whether one can suggest the decision procedure for the GF on the basis of DC-rules. Though this question is still opened, we propose the decision procedure for the GF which uses the rule system which is very close to the one of DC; in fact, our decision procedure represents a ”mirror” to the tableau algorithm for the GF. There are various approaches to the GF on the basis of tableau algorithms, here we take for a basis of our DCL algorithm the decision procedure using the so called blocking technique suggested by C.Hirsch and St.Tobies [4].
Let’s add to our language the equality as a special relation. Here we use the notations from [4]. Consider the language of the relational structure of FOL with equality and without constants. All formulas are assumed to be in NNF. For a sentence $\psi \in GF$, let $clos(\psi)$ be the smallest set that contains $\psi$ and is closed under subformulas. Let $C$ be the set of free variables. With $clos(\psi, C)$ we denote the set $clos(\psi, C) = \{ \phi(a) : a \subseteq C, \phi(x) \in clos(\psi) \}$.

The width of a formula $\psi \in GF$ is defined by $width(\psi) := \max\{|free(\phi) : \phi \in clos(\psi)\}$.

Let $\psi \in GF$ be a formula in NNF. A completion tree $T_c = (V, E, \Delta, N)$ for $\psi$ is a node labelled tree $(V, E)$ with the labelling function labelling each node $\upsilon \in V$ with the set of free variables used in the formulas in a given node, $\Delta$ labelling each node with a subset of $clos(\psi, C(\upsilon))$ and the function $N$ mapping each node to a distinct natural number with the property that if $\upsilon$ is an ancestor of $\omega$, then $N(\upsilon) < N(\omega)$.

A free variable $u$ is called shared between the two nodes $\upsilon_1, \upsilon_2 \in V$, if $u \in C(\upsilon_1) \cap C(\upsilon_2)$, and $u \in C(\omega)$ for all nodes of the (unique, undirected, possibly empty) shortest path connecting $\upsilon_1$ to $\upsilon_2$. A node $v \in V$ is called directly blocked by a node $\omega \in V$, if $\omega$ is not blocked, $N(\omega) < N(v)$ and there is an injective mapping $\pi$ from $C(\upsilon)$ into $C(\omega)$ such that for all free variables $u \in C(\upsilon)$ that are shared between $\upsilon$ and $\omega$, $\pi(u) = u$, and $\pi(\Delta(\upsilon)) = \Delta(\omega)|_{\pi(C(\upsilon))}$. A node is called blocked if it is directly blocked or if its predecessor is blocked.

A completion tree $T_c$ is called a deduction tree $T_d$, if for the last node $v \in V$ of each branch of the tree $T_c$ either

- $u = u \in \Delta(v)$ (for a free variable $u \in C(\upsilon)$), or
- $\{\beta(a), \neg \beta(a)\} \subseteq \Delta(v)$ (for the atomic formula $\beta$ and the tuple $a \subseteq C(\upsilon)$).

A completion tree $T_c$ is called complete, if none of the DCL-rules (see below) can be applied to $T_c$. The application of the described DCL-rules allows to obtain the decision procedure for the given formula $\psi$. Let’s have the formula $\psi$ written in NNF. We ascribe for the formula $\psi$ the initial node $\upsilon_0, \Delta(\upsilon_0) = \{\psi\}, C(\upsilon_0) = \emptyset$ of the completion tree and construct that tree by the successive application of the admissible DCL-rules. As that process terminates (see below), after the finite number of
steps one obtains an axiom or a non-reducible formula. The decision procedure for DCL algorithm is the following:

- If the last node of each branch of the completion tree for $\psi$ represents an axiom then the output of the algorithm is "$\psi$ is valid";
- If there is the branch, the last node of which represents a non-reducible formula, then the output is "$\psi$ is not valid".

Consider the following rule system.

**DCL-algorithm**

\[
\begin{align*}
\Diamond_\wedge & \; \text{if } \phi \land \theta \in \Delta(v) \text{ then } \Delta(v) \rightarrow_\wedge \Delta(v) \cup \{\chi\} \text{ for some } \chi \in \{\phi, \theta\}; \\
\Diamond_\lor & \; \text{if } \phi \lor \theta \in \Delta(v) \text{ then } \Delta(v) \rightarrow_\lor \Delta(v) \cup \{\phi, \theta\}; \\
\Diamond_\neq & \; \text{if } a \neq b \in \Delta(v) \text{ then for all } \omega \text{ that share } a \text{ with } v, \\
& \quad C(\omega) \rightarrow_\neq (C(\omega) \{a\}) \cup \{b\} \text{ and } \Delta(\omega) \rightarrow_\neq \Delta(\omega)[a \mapsto b]; \\
\Diamond_\forall & \; \text{if } \forall y (\alpha(a, y) \supset \phi(a, y)) \in \Delta(v) \text{ and for every } b \subseteq C(v), \{\neg \alpha(a, b), \phi(a, b)\} \not\subseteq \Delta(v), \text{ and there is no child } \omega \text{ of } v \text{ with } \{\neg \alpha(a, b), \phi(a, b)\} \subseteq \Delta(\omega) \text{ for some } b \subseteq C(\omega), \text{ and } v \text{ is not blocked, then } V \rightarrow_\forall V \cup \{\omega\}. \\
\Diamond_\exists & \; \text{if } \exists y (\alpha(a, y) \supset \phi(a, y)) \in \Delta(v) \text{ and for every } b \subseteq C(v) \text{ such that } \neg \alpha(a, b) \in \Delta(v), \text{ and } \phi(a, b) \not\in \Delta(v), \\
& \quad \text{then } \Delta(v) \rightarrow_\exists \Delta(v) \cup \{\phi(a, b)\}^*. \\
\end{align*}
\]

*) If existentially quantified formula contains only one variable within the scope of existential quantifier, then it should be presented in a form: $v_0 \neq v_0, \ldots, v_n \neq v_n, \exists x \phi(x)$, where $v_0, \ldots, v_n$ — free variables that are presented in the numerator of the $\Diamond_\exists$ rule. If there is no free variable in the numerator, then the arbitrary free variable is chosen.

### 6. Termination of DCL-algorithm

**Lemma 6.1.** Let $\psi \in GF$ be a sentence in NNF. Any sequence of rule applications of the DCL algorithm starting from the initial tree terminates.
The proof of lemma follows from the following technical result in Lemma 6.2. Let \( \psi \in GF \) be a sentence in NNF with \(|\psi| = n\), \( \text{width}(\psi) = m \) and \( T_c \) a completion tree generated for \( \psi \) by application of the DCL-rules. Then, for every node \( v \) in \( T_c \),

1. \(|C(v)| \leq m\),
2. \(|\Delta(v)| \leq n \times (m + 1)^m\),
3. any \( l > 2^{n \times (m + 1)^m} \) distinct nodes in \( T_c \) contain a blocked node.

\section{Soundness}

Theorem. Let \( \psi \) be a guarded formula in NNF. If there is a sequence of rule applications from DCL algorithm starting from the initial point that yields a deduction tree such that the last node of each branch of this tree represents an axiom (\( \psi \) is deducible), then the formula \( \psi \) is valid.

Proof. To prove the soundness of DCL algorithm, we note that every axiom in the deduction tree is a valid formula, and show that every application of the rules for DCL, taken in the opposite direction, preserves the validity of the disjunction of the formulas. Since the rules \( \Diamond_{\forall}, \Diamond_{\land} \) and \( \Diamond_{\lor} \) of DCL are the same as in the definition of the DC, they represent simply the opposite rules for the ones in the system of Natural Deduction, thus they preserve the validity, and each formula obtained from the axioms using the opposite rules for \( \Diamond_{\forall}, \Diamond_{\land} \) and \( \Diamond_{\lor} \), is valid.

Let us show that the rules, inversed to the rules \( \Diamond_{\neq} \) and \( \Diamond_{\exists} \), do also preserve the validity.

**Inversed \( \Diamond_{\neq} \) rule**

If \( \{\Gamma'\} \) then \( \{\Gamma, v_i \neq v_j\} \) (\( \Gamma' \) is a result of substitution in \( \Gamma \) of all occurrences of the variable \( v_i \) by \( v_j \)).

Suppose, \( \{\Gamma'\} \) is valid and \( \{\Gamma, v_i \neq v_j\} \) is not. Let \( \Gamma \) stands for the disjunction of the formulas \( B_0 \lor B_1 \lor \ldots \lor B_n \), and \( \Gamma' \) stands for \( B'_0 \lor B'_1 \lor \ldots \lor B'_n \). Hence, we have a model \( M \) and a valuation \( \nu \), in which \(|B'_0 \lor B'_1 \lor \ldots \lor B'_n|_{M, \nu} = t\) and \(|\neg B_0 \land \neg B_1 \land \ldots \land \neg B_n \land \neg B_i \land v_i = v_j|_{M, \nu} = t\).

Since \(|v_i = v_j|_{M, \nu} = t\), we substitute every occurrence of \( v_i \) in \( \Gamma \) by \( v_j \), then we get \(|\neg B'_0 \land \neg B'_1 \land \ldots \land \neg B'_n|_{M, \nu} = t\), a contradiction.

**Inversed \( \Diamond_{\exists} \) rule**

If \( \{\Gamma, \neg \alpha(u, y), \exists x(\alpha(x, y) \land \phi(x, y)), \phi(u, y)\} \) then \( \{\Gamma, \neg \alpha(u, y), \exists x(\alpha(x, y) \land \phi(x, y))\} \).
Let the formula \( \{ \Gamma \} \lor \neg \alpha(u,y) \lor \exists x(\alpha(x,y) \land \phi(x,y)) \lor \phi(u,y) \) be valid. Assume the formula \( \{ \Gamma \} \lor \neg \alpha(u,y) \lor \exists x(\alpha(x,y) \land \phi(x,y)) \) is not valid, thus there is a model \( M \) and a valuation \( \upsilon \) s.t.

\[
|\{ \Gamma \} \lor \neg \alpha(u,y) \lor \exists x(\alpha(x,y) \land \phi(x,y))|_M, \upsilon = t \quad \text{and} \quad |\{ \Gamma \} \lor \neg \alpha(u,y) \lor \exists x(\alpha(x,y) \land \phi(x,y))|_M, \upsilon = f.
\]

From the last expression we get

\[
|\alpha(u,y) \land \forall x(\alpha(x,y) \lor \neg \phi(x,y))|_M, \upsilon = t
\]

and

\[
|\neg \phi(u,y)|_M, \upsilon = t
\]

So far, we have shown that the inverse rules for the rules of DCL algorithm do also preserve the validity, thus every formula \( \psi \), which is obtained from the axioms by application of the inverse DCL-rules, is valid, therefore the soundness has been proved.

It could be shown that all the DCL-rules also do preserve the validity when taken in another direction. However, that fact is not sufficient to prove the completeness of DCL algorithm, since the complexity of the formulas does not decrease in the application of one of the rules, namely, of \( \Diamond \exists \) rule.

8. Completeness

To prove the completeness of DCL algorithm, we use and adapt the formalism suggested in [4]. We construct the special model \( \mu \) using so called unraveling construction, we’ll use \( \mu \) as a contr-model in the completeness proof. Let \( T_c = (V,E,C,\Delta,N) \) be completion tree for \( \psi \) and let \( V_u = \{ v \in V : v \) is not blocked or directly blocked\}. The set \( P(T_c) \) is defined inductively by

- if \( [\nu_1 \nu_2 \ldots \nu_n] \in P(T_c) \) for the root \( \nu_0 \) of \( T_c \),
- if \( [\nu_1 \nu_2 \ldots \nu_n] \in P(T_c) \), the node \( \omega \) is a successor of \( \nu_n \) and \( \omega \) is not blocked, then \( [\nu_1 \nu_2 \ldots \nu_n \omega] \in P(T_c) \),
- if \( [\nu_1 \nu_2 \ldots \nu_n] \in P(T_c) \), the node \( \omega \) is a successor of \( \nu_n \) blocked by the node \( u \in V \), then \( [\nu_1 \nu_2 \ldots \nu_n \nu] \in P(T_c) \).

The set \( P(T_c) \) forms a tree, and \( \nu' \) is called a successor of \( \nu \) if it is obtained from \( \nu \) by concatenating one element \( \nu \) at the end. Then the two auxiliary functions \( F, F' \) are defined by setting \( F(p) = \nu_n \) and \( F'(p) = \nu'_n \) for every path \( p = [\nu_1 \nu_2 \ldots \nu_n] \).
The universe of $\mu$ consists of (classes of) free variables labelling nodes in $T_c$ paired with the paths at whose node they appear. Formally, $C(T_c) = \{(u, p) : p \in P(T_c) \land u \in C(F(p))\}$.

The free variables appearing at consecutive nodes of $T_c$ stand for the same element, and the same holds for the free variables related by a mapping $\pi$ verifying a block. $C(T_c)$ is factorized by defining the special relation $\sim$ as follows:

- $(a, p) \sim (a, q)$ if $q$ is a successor of $p$ in $P(T_c)$, $F'(q)$ is an unblocked successor of $F(p)$, and $a \in C(F(p)) \cap C(F'(q))$,
- $(a, p) \sim (b, q)$ if $q$ is a successor of $p$ in $P(T_c)$, $F'(q)$ is a blocked successor of $F(p)$, and $a \in C(F(p)) \cap C(F'(q))$, and $\pi(a) = b$ for the function $\pi$ that verifies that $F'(q)$ is blocked by $F(q)$.

The reflexive, transitive closure of $\sim$ is denoted with $\equiv$, and $[a, p]_\equiv$ denotes the set $\{(b, q) \in C(T_c) \mid (b, q) \equiv (a, p)\}$.

Since $(a, p) \sim (b, q)$ iff $p, q$ are neighbours in $P(T)$, the set $P([a, p]_\equiv) := \{q \mid \exists b(b, q) \in [a, p]_\equiv\}$ is a subtree of $P(T_c)$. The classes of $C(T_c)_\equiv$ will be the elements of $\mu$.

**Claim 1.** Let $p \in P(T_c)$ and $a, b \in C(F(p))$. Then $(a, p) \equiv (b, q)$ iff $a = b$.

**Claim 2.** Let $p, q \in P(T_c)$ with $p = [\frac{v_p}{v_1} \ldots \frac{v_p}{v_n}], q = [\frac{v_q}{v_1} \ldots \frac{v_q}{v_n}]$. If, for $a \in C(v_n), b \in C(\omega)$, $(a, p) \equiv (b, q)$ then $(a, p) \equiv (b, q)$.

**Claim 3.** Let $p, q \in P(T_c)$, $a \subseteq C(F(p)), b \subseteq C(F(q))$, $a, b$ non-empty tuples, and $(a, p) \equiv (b, q)$.

Then 1) for every atomic formula $\alpha$, $\alpha(a) \in \Delta(F(p))$ iff $\alpha(b) \in \Delta(F(q))$ and 2) for every existentially quantified formula $\phi$, $\phi(a) \in \Delta(F(p))$ iff $\phi(b) \in \Delta(F(q))$.

**Remark.** Note that, if the formula $\phi(a)$ is either atomic, or is existentially quantified, then the application of any direct or backward $\lor$ rule can not remove $\phi(a)$. Thus, if the formula $\phi(a)$ is presented in a given node $v$ of the completion tree $T_c$, then it is necessarily presented in its neighbour $\omega$.

**Proof.** Since both propositions are symmetric, let’s prove only one direction. It can be shown ([4]), that in $P(T_c)$ there are the paths $p_1, \ldots, p_k$ for which there exist tuples of free variables $u_1, \ldots, u_k$ with $(u_1, p_1) \equiv \ldots \equiv (u_k, p_k)$, $p = p_1, q = p_k$, $a = u_1, b = u_k$. Since $a, b$ are non-empty, so are the $u_i$. From Claim 2 we get that for any two neighbours $p_i, p_{i+1}$ in $P(T_c)$,
Let's show that for the contr-model one could choose the model \( p_i \) so that \( \mu \) is defined at \( (u_i, p_i) \). By induction on \( i \) (\( 1 \leq i \leq k \)) we show that if \( \alpha(u_i) \in \Delta(F(p_i)) \) then \( \alpha(u_{i+1}) \in \Delta(F(p_{i+1})) \) and if \( \phi(u_i) \in \Delta(F(p_i)) \) then \( \phi(u_{i+1}) \in \Delta(F(p_{i+1})) \). For \( i = 1 \) the Claim 3 holds. Suppose we have the proof up to \( i \). Assume \( p_{i+1} \) is a successor of \( p_i \) in the tree \( P(T_c) \). There are the two cases.

**Case 1.** \( F'(p_{i+1}) \) is not blocked. Then \( F(p_{i+1}) = F'(p_{i+1}) \) and by definition of \( \sim \), \( F(p_{i+1}) \) is a successor of \( F(p_i) \) in \( T_c \) and \( u_i = u_{i+1} \) holds.

- If \( \alpha(a) \in \Delta(F(p)) \), then \( \alpha(u_i) \in \Delta(F(p_i)) \) holds by induction and by Rem, this implies \( \alpha(u_{i+1}) \in \Delta(F(p_{i+1})) \);
- If \( \phi(a) \) is an existential formula, \( \phi(a) \in \Delta(F(p)) \), then by induction \( \phi(u_i) \in \Delta(F(p_i)) \), and by Rem, this implies \( \phi(u_{i+1}) \in \Delta(F(p_{i+1})) \).

**Case 2.** \( F'(p_{i+1}) \) is blocked by \( F(p_{i+1}) \) with function \( \pi \) and \( F'(p_{i+1}) \) is a successor of \( F(p_i) \) in \( T_c \). Then, by definition of \( \sim \), \( u_{i+1} = \pi(u_i) \) and \( a \in C(F(p_i)) \cap C(F'(p_{i+1})) \).

- If \( \alpha(a) \in \Delta(F(p)) \), then \( \alpha(u_i) \in \Delta(F(p_i)) \) holds by induction and due to Rem, this implies \( \alpha(u_{i+1}) = \alpha(u_{i+1}) \in \Delta(F(p_{i+1})) \).
- If \( \phi(a) \) is an existential formula, \( \phi(a) \in \Delta(F(p)) \), then by induction \( \phi(u_i) \in \Delta(F(p_i)) \), and by Rem, this implies \( \phi(u_{i+1}) \in \Delta(F'(p_{i+1})) \). Since \( F(p_{i+1}) \) blocks \( F'(p_{i+1}) \), \( \pi(\phi(u_i)) = \phi(u_{i+1}) \in \Delta(F(p_{i+1})) \).

**Definition (\(*\)).** Let's define the structure \( \mu \) over the universe \( A = C(T_c) \div \approx \). For a relation \( R \in T \) of arity \( m \), \( R^a \) is defined to be the set of tuples \( ([a_1, p_1], ..., [a_m, p_m]) \) for which there exists the path \( p \in P(T_c) \) and the free variables \( u_1, ..., u_m \) such that \( (u_i, p) \approx [a_i, p_i] \) for all \( 1 \leq i \leq m \), and \( \sim_R u_1 ... u_m \in \Delta(F(p)) \).

**Theorem (Contraposition).** For every path \( p \in P(T_c) \) and tuple \( a \subseteq C(F(p)) \), if \( \psi(a) \in \Delta(F(p)) \) and there is no deduction tree \( T_d \) for \( \psi(a) \), then there is a model \( \mu \) s.t. \( \mu \not\models \psi(a) \).

**Proof.** Let's show that for the contr-model one could choose the model \( \mu \) defined at (\(*\)). The proof goes by induction on the structure of \( \psi(a) \).

- \( \psi(a) = \beta(a) \): Suppose, the atomic formula \( \beta(a) \in \Delta(F(p)) \) and there is no \( T_d \) for \( \beta(a) \). Assume \( \mu \models \beta(a) \), then, by definition of \( \mu \), there must be a path \( p' \in P(T_c) \) and free variables \( u \) s.t. \( (a, p) \approx (u, p') \) and
\neg \beta(\text{u}) \in \Delta(F(p')). From Claim 3 it follows that \neg \beta(\text{a}) \in \Delta(F(p)) which gives the deduction tree \text{T}_d and contradicts the assumption.

- \psi(\text{a}): \neg \beta(\text{a}). The proof is similar to the one above.

- \psi(\text{a}): \beta(\text{a}) \land \gamma(\text{a}). Suppose, there is no deduction tree \text{T}_d for \psi(\text{a}). It follows that either there is no deduction tree for \beta(\text{a}), or there is no deduction tree for \gamma(\text{a}). By induction hypothesis (I.H.), \mu \not\models \beta(\text{a}) or \mu \not\models \gamma(\text{a}). Hence, \mu \not\models \beta(\text{a}) \land \gamma(\text{a}).

- \psi(\text{a}): \beta(\text{a}) \lor \gamma(\text{a}). Suppose, there is no deduction tree \text{T}_d for \psi(\text{a}). It follows that there is no deduction tree for \beta(\text{a}) and there is no deduction tree for \gamma(\text{a}) as well. By I.H., \mu \not\models \beta(\text{a}) and \mu \not\models \gamma(\text{a}). Hence, \mu \not\models \beta(\text{a}) \lor \gamma(\text{a}).

- \psi(\text{a}): Assume \exists y(\phi(\text{a}, y) \land \phi(\text{a}, y)) \in \Delta(F(p)) and there is no deduction tree \text{T}_d for \psi(\text{a}). Suppose, \mu \models \phi(\text{a}, \text{b}) for some arbitrarily chosen \text{b}, that means, by definition of \mu, that there is the path \text{p}' \in P(T_\text{c}) and free variables \text{c}, \text{d} s.t. \text{(a, p)} \approx \text{(c, p')}, \text{(b, p)} \approx \text{(d, p')} and \neg \phi(\text{c, d}) \in \Delta(F(p')). By Claim 3, \neg \phi(\text{a, b}) \in \Delta(F(p)). Since also \exists y(\phi(\text{a, y}) \land \phi(\text{a, y})) \in \Delta(F(p)), then, by \Diamond rule, \phi(\text{a, b}) \in \Delta(F(p)). There is no deduction tree \text{T}_d for the formula \phi(\text{a, b}), since otherwise \text{T}_d would also be for \psi(\text{a}). By induction, \mu \not\models \phi(\text{a, b}), so \mu \not\models \phi(\text{a, b}) and since \text{b} is arbitrarily chosen then \mu \not\models \psi(\text{a}).

- \psi(\text{a}): \forall y(\phi(\text{a, y}) \lor \phi(\text{a, y})). There are the two cases.

Case 1. There is a tuple \text{b} \subseteq C(F(p)) with \{\phi(\text{a, b}) \subseteq \Delta(F(p))\}. There is no \text{T}_d for the formula \phi(\text{a, b}) since otherwise \text{T}_d would be for \psi(\text{a}). By induction, \mu \not\models \phi(\text{a, b}) and therefore \mu \not\models \psi(\text{a}).

Case 2. There is no tuple \text{b} \subseteq C(F(p)) with \{\phi(\text{a, b}) \subseteq \Delta(F(p))\}. and there is a successor \omega of F(p) and \text{b} \subseteq C(\omega) with \{\phi(\text{a, b}) \supseteq \Delta(\omega)\}

The node \omega can be blocked or not.

1) If \omega is not blocked, then \text{p'} = [p_{\text{u}}]\in P(T_\text{c}) with (a, \text{p'}) \approx (a, \text{p}) and (b, \text{p'}) \approx (b, \text{p}). There is no \text{T}_d for the formula \phi(\text{a, b}) since otherwise \text{T}_d would also be for \psi(\text{a}). By induction, \mu \not\models \phi(\text{a, p'), \text{b, p')}) \supseteq \phi(\text{a, p'), \text{b, p')}), and, since (a, \text{p'}) \approx (a, \text{p}) and \text{(b, p')} \approx (b, \text{p}), then \mu \not\models \phi(\text{a, p'), \text{b, p')}) \supseteq \phi(\text{a, p'), \text{b, p')}).

2) If \omega is blocked by a node \text{u} with function \pi, then \text{p'} = [p_{\text{u}}]\in P(T_\text{c}), and \pi(\phi(\text{a, b}) \supseteq \phi(\text{a, b})) \subseteq \Delta(\omega). There is no \text{T}_d for the formula \phi(\text{a, b}) (otherwise there is \text{T}_d for \psi(\text{a})), therefore there
is no $T_d$ for the formula $\pi((\alpha(a, b) \supset \phi(a, b)) \subseteq \Delta(u)$. By induction, $\mu \not\models \alpha([\pi(a), p'], [\pi(b), p']) \supset \phi([\pi(a), p'], [\pi(b), p'])$. Since $(\pi(a), p') \approx (a, p)$ and $(\pi(b), p') \approx (b, p)$, then $\mu \not\models \alpha([a, p], [b, p]) \supset \phi([a, p], [b, p])$, hence $\mu \not\models \psi(a)$.

9. Conclusion

The rules of the described DCL algorithm differs from the ones of the GDC in only one, existential rule. In order to compare them let’s write them together in the same notation.

$\Diamond_{\text{GDC}}$ If $\{\exists y (\alpha(x, y) \land \phi(x, y)), \Gamma\}$ then $\exists y (\alpha(x, y) \land \phi(x, y))$, $\alpha(x, t_i) \land \phi(x, t_i)$ does not occur in the previous steps of the rule application:

$\Diamond_{\text{DCL}}$ If $\{\neg \alpha(u, y), \exists y (\alpha(x, y) \land \phi(x, y)), \Gamma\}$ then $\{\neg \alpha(u, y), \exists y (\alpha(x, y) \land \phi(x, y)), \phi(u, y), \Gamma\}$.

The first existential rule produces the new tuple of terms while the second does not produce any new tuple. In partial case when the existentially quantified formula contains only one variable within the scope of the quantifier (so it could be rewritten as $\exists y ((y = y) \land \phi(y))$ the two presented rules are the same, but in general case they are different. So the following questions are of interest for the further research:

1. To try to develop the decision procedure using the blocking technique on the basis of the rule system for the GDC like it was done with the DCL- rules in this our paper,
2. To compare GDC decision procedure to the DCL algorithm and
3. To make a comparison between the "classical" Schütte-style completeness proofs for the full FOL, for the GF of FOL and the GDC completeness proof to see that it is the restriction of quantifiers (guarded condition) that allows to get the finite model structure (by using the blocking technique) to prove the completeness.

References


Kemerovo State University, Novokuznetsk Branch
654080 Novokuznetsk, Tsyolkovskogo, 23, Russia
e-mail: andr-kuznetsov@yandex.ru