A PROOF PROCEDURE FOR THE FIRST ORDER LOGIC

Miodrag Kapetanović, Aleksandar Krapež

Abstract. A new proof procedure is given for the classical predicate logic which combines variants of the tableaux and resolution methods. Soundness and completeness of the resulting system are proved.

We shall present a proof procedure for the classical predicate logic combining features of the semantic tableaux method with the well known resolution rule of J. Robinson (see [2]). Given a closed first order formula (in a language without function symbols) whose validity is to be tested, we first build its reduction tree (growing downwards): put the formula at the top and then use the usual propositional tableaux rules (as described for instance in [1, pp. 26, 28]) as well as the following quantifier rules:

\[
\begin{align*}
\forall x A(x) & \quad \Rightarrow \quad \forall x A(x) \\
A(c) & \quad \Rightarrow \quad \forall x A(x)
\end{align*}
\]

\[
\begin{align*}
\exists x A(x) & \quad \Rightarrow \quad \exists x A(x) \\
A(c) & \quad \Rightarrow \quad \exists x A(x)
\end{align*}
\]

The rules apply as follows. A branch with \(\forall x A(x)\) is extended by adding \(A(c)\) as a new end node (the same for \((\exists)\), where \(c\) is a new constant i.e. occurs in no formula on the tree. Similarly a branch containing \(\exists x A(x)\) (the same for \((\forall)\)) gets a new extension \(A(a_{n+1})\) where \(a_{n+1}\) differs from all \(a_1, \ldots, a_n\) (by which this occurrence of \(\exists x A(x)\) has been instantiated previously) and all \(a_i (1 \leq i \leq n+1)\)

AMS Subject Classification (1980): Primary 03B10 Secondary 03F07.
occur already on the tree. If there are no constants on the tree at all then instantiate \( \exists x A(x) \) by a new one. All (occurrences of all) formulas are used at most once w. r. t. a patricular branch and discharged after that. As an exception an occurrence of \( \exists x A(x) \) or \( \neg \forall x A(x) \) is discharged only in case it cannot be instantiated by any constant anymore and no new constant can appear on the tree by any further reduction. By basic sentences of our language, enriched by new constants, we mean atomic sentences and their negations. The reduction trees we are most interested in are finished ones i. e. those on which all occurrences of all formulas are discharged. They have the fundamental property expressed by the following.

**Lemma.** Given a finished reduction tree \( T^* \) of a closes formula, the formula is valid iff for every model (in the language of \( T^* \)) there is a branch of \( T^* \) such that all basic sentences from that branch hold in the model.

**Proof.** (\( \Leftarrow \)) Suppose that a given formula \( F \) is not valid i. e. there is an \( M \) such that \( M \not \models F \). Define an expansion \( M^* \) of that model as follows. First enumerate all constants appearing in \( T^* \) into a sequence \( c_0, c_1, \ldots \) and let \( M_0 = M \). Then for all \( n < \omega \) let the language \( L(M_{n+1}) = L(M_n) \cup \{ c_n \} \). Suppose that \( c_n \) was introduced by applying \( (\forall) \) to \( \forall x A(x) \) (the same in case of \( (\exists) \)). Then if \( M_n \not \models \forall x A(x) \) take some \( e \in | M | \) such that \( M_n \not \models A[e] \) and put \( c_n^{M_{n+1}} = e \). Otherwise \( c_n^{M_{n+1}} \) is arbitrary. Finally let \( | M^* | = | M |, L^* = \bigcup_{n<\omega} L(M_n), c_n^{M^*} = c_n^{M_{n+1}} \).

Obviously \( M^* \not \models F \) but suppose there is a branch \( B \) of \( T^* \) whose all basic sentences hold in \( M^* \). Then by induction it follows that all sentences from \( B \) (including \( F \)) hold in \( M^* \): if \( A \land B \in B \) then both \( A, B \in B(B \text{ is finished!}) \) and they hold in \( M^* \) by induction hypothesis (the same argument for \( \neg A, \neg (A \lor B), \neg (A \Rightarrow B) \)). If \( A \lor B \in B \) then \( A \in B \) or \( B \in B \), say \( A \in B \), and again \( M^* \models A \) by induction hypothesis (the same for \( \neg (A \land B), A \Rightarrow B \)). If \( \forall x A(x) \in B \) (the same for \( \neg \exists x A(x) \)) then \( A(c_n) \in B \) for some \( n \) and \( M^* \models A(c_n) \) by induction hypothesis. Using the definition of \( c_n^{M^*} \) we get \( M^* \models \forall x A(x) \). Finally if \( \exists x B(x) \in B \) then again \( B(c_n) \in B \) for some \( n \) and \( M^* \models B(c_n) \) by induction hypothesis. As \( F \in B \) we get the contradiction.

(\( \Rightarrow \)) Suppose that \( F \) is valid and let \( M \) be any structure in the language \( L^* \). Take the restriction \( M^C \models F \upharpoonright \{ e \in | M | \} \) for some constant \( c, c^M = e \) and choose a branch \( B \) by defining inductively its initial segments \( B_n = (F_0, F_1, \ldots, F_n), n = 0, 1, \ldots \) (we identify a node of \( T^* \) with the formula placed at it): if \( F_n \) has only one successor take it for \( F_{n+1} \). If it has at least two successors choose one (say the leftmost) that holds in \( M^C \). We shall show by induction on \( n \) that \( M^C \models F_n \) and the same time see that our definition is correct. Certainly \( M^C \models F_0 \), so suppose \( F_{n+1} \) is the only successor of \( F_n \). It could only appear as a component of some \( F_i (i \leq n) \) of the form \( \neg \neg A \) or \( A \land B \) or \( \neg (A \Rightarrow B) \) or \( \neg (A \lor B) \) or \( \forall x A(x) \) or \( \neg \exists x A(x) \). Take for example the case \( F_i = A \land B, F_{n+1} = A \). Then \( M^C \models F_i \) by induction hypothesis so \( M^C \models F_{n+1} \). If \( F_i \) is some \( \forall x A(x) \) then for some \( e F_{n+1} \) is \( A(e) \), so obviously \( M^C \models F_{n+1} \). e. t. c. Suppose now that \( F_{n+1} \) has been chosen among two or more successors of \( F_n \). Then it must be a component of some \( F_i \) of the form \( A \lor B \) or \( A \Rightarrow B \) or \( \neg (A \land B) \) or \( \exists x A(x) \), or \( \neg \forall x A(x) \). If for instance \( F_i \) is
some $A \lor B$ and $F_{n+1}$ is $B$ then $M_C \models F_i$ by induction hypothesis, so $M_C \models A$ or $M_C \models B$. The choice of $B$ implies that $M_C \not\models A$, hence $M_C$ must satisfy $B$ which shows both the possibility and correctness of our choice. The most interesting case is that of $\exists x A(x)$ (and $\forall x A(x)$). If $M_C \models \exists x A(x)$ then $M_C \models A[e]$ for some $e \in M_C$. But $e = e^{M_C}$ for some $e$ by the definition of $M_C$, so again there exists a successor of $F_n$ true in $M_C$ proving that $F_{n+1}$ has been well defined. Finally, this shows that the whole of $B$ holds in $M_C$ (including all basic sentences of $B$) i.e. the lemma is proved, since $M$ was chosen arbitrarily.

To get a complete set of rules we have to add the following dual resolution rule to the reduction rules given already:

\[(DR)\quad \text{we resolve } S_1 \text{ and } S_2 \text{ to get their resolvent } R(S_1, S_2, A) = (S_1 \setminus \{A\} \cup (S_2 \setminus \{\neg A\})).\]

In derivations we allow only those $S_i$ which are either sets of all basic sentences from a finished branch or resolvents obtained therefrom. A formula is proved if we can derive the empty set from it, using $(DR)$ and the reduction rules.

As is well known, the existence of a model of a set of basic sentences is equivalent to the existence of the appropriate valuations where we take these sentences as propositional variables. If a valuation satisfies at least one of the sets of basic sentences from $R(S_1, S_2, A), S_3, \ldots, S_k$ then it satisfies at least one from $S_1, S_2, S_3, \ldots, S_k$. The proof systems from the tautology $(B \land C) \Rightarrow ((B \land A) \lor (C \land \neg A))$ since we can look at $S_i$'s as conjunctions of their members.

Since every valuation satisfies the empty set, it follows that every valuation satisfies at least one among the sets of basic sentences from the finished branches from which we derived the empty set. Together with $(\Rightarrow)$ of Lemma this implies that our formula is valid i.e. we can prove only valid formulas.

To show the converse take a valid formula and consider its finished reduction tree $T^*$. Then the set $\{\Gamma \mid \Gamma$ is the set of all basic sentences of some branch of $T^*\}$ has, by $(\Leftarrow)$ of Lemma, the property that for any valuation at least one of its members is true. By the compactness theorem there is a finite subset $\{\Gamma_1, \ldots, \Gamma_m\}$ with the same property. Using ideas from [3] and [2] we shall prove (by induction on $n = |\Gamma_1| + \cdots + |\Gamma_m| - m$) that we can infer $\emptyset$ from this subset ($|\Gamma_i|$ is cardinality of $\Gamma_i$).

If $n = 0$ then all $\Gamma_i$ contain only one element so there are $j, k \leq m$ and an atomic sentence $A$ such that $\Gamma_j \{A\}, \Gamma_k = \{\neg A\}$. So the basis of induction is true.

For $n > 0$ take $\Gamma_i$ with at least two elements and represent it as a disjoint union of nonempty sets $\Delta_1$ and $\Delta_2$. By the induction hypothesis $\emptyset$ can be inferred from $\Gamma_1, \ldots, \Gamma_{i-1}, \Delta_1, \Gamma_{i+1}, \ldots, \Gamma_m$. Repeating the same steps in this inference, with $\Delta_1$ replaced by $\Delta_1 \cup \Delta_2 (= \Gamma_i)$ we get the inference of $\Delta_2$ from $\Gamma_1, \ldots, \Gamma_m$. Using induction hypothesis again we can extend this inference to an inference of $\emptyset$ from $\Gamma_1, \ldots, \Gamma_{i-1}, \Delta_2, \Gamma_{i+1}, \ldots, \Gamma_m$ thus obtaining the inference of $\emptyset$ from $\Gamma_1, \ldots, \Gamma_m$. 
This proves the completeness of our system: only valid formulas are provable.

REFERENCES


Matematički Institut
Knez Mihailova 35.
11000 Beograd, P.P. 367
Yugoslavia

(Received 18.10.1988)