Discrete Mathematics (II) Spring 2013

Lecture 14: Tableau Proof of Predicate Logic

Lecturer: Yi Li

1 Overview

In this lecture, we will show you how to prove the validity of sentences of predicate logic.

To handle quantifiers, we introduce new atomic tableau. In this lecture, we focus on how to handle quantifiers.

2 Atomic tableau

In predicate logic, we have introduced variables and two quantifiers $\exists$ and $\forall$. All sentences without quantifiers are handled in the same way as propositions in propositional logic, we just consider the cases with quantifiers.

As we have done in propositional logic, we introduce the atomic tableaux according to their semantics. In previous lecture, we already have $\neg(\forall x)\varphi(x) \equiv (\exists x)\neg\varphi(x)$.We have the following new atomic tableaux as in Figure 1. For convenience, we call the tableaux with "for all ground term t"

$T\ (\forall x)\varphi(x)$	$F\ (\forall x)\varphi(x)$	$F\ (\exists x)\varphi(x)$	$T\ (\forall x)\varphi(x)$
$\vert$	$\vert$	$\vert$	$\vert$
$T\ \varphi(t)$	$F\ \varphi(c)$	$F\ \varphi(t)$	$T\ \varphi(c)$
for any ground term t of $\mathcal{L}_C$	for a new constant c	for any ground term t of $\mathcal{L}_C$	for a new constant c

Figure 1: Atomic Tableaux with Quantifiers

$\forall$-style and the other two $\exists$-style.

According to the definition of truth of $\forall x\varphi(x)$, all elements in domain of a given structure should be referred as a witness. For $\forall$-style, we use for any ground term t. However, once we fix it for some special element in a domain in a tableau, it can not name some others. Then we must duplicate the reducing node in order to guarantee that we can make sure every ground term is its witness. We need treat it specially when we define what entry is reduced. It will be introduced late.

Definition 1. *We define tableaux as binary trees labeled with signed sentence(of $\mathcal{L}_C$) called entries by induction.*

1. *All atomic tableaux are tableaux. The requirement that c be new in Cases 7b and 8a here simply means that c is one of the constants c_i added on to $\mathcal{L}$ to get $\mathcal{L}_C$(which therefore does not appear in φ).*

2. *if τ is a finite tableau, P a path on τ, E and entry of τ occurring on P. $\tau\prime$ is obtained from τ by adjoining an atomic tableau with root entry E to τ at the end of the path P, then $\tau\prime$ is also a tableau. Here the requirement that c be new in Case 7b and 8a means that it is one of the c_i that do not appear in any entries on P. (In actual practice it is simpler in terms of bookkeeping to choose one not appearing at any node of τ.)*

3. *If τ_0 is a finite tableau and $\tau_0, \tau_1, \ldots, \tau_n, \ldots$ is a sequence of tableaux such that, for every $n \geq 0, \tau_{n+1}$ is constructed from τ_n by an application of 2, then $\tau = \cup\tau_n$ is also a tableau.*

For $\exists$-style, we always introduce new constant which does not occur before along the path which it is in. If the constant is not new to this language, it should name a specified element. So when different structure is concerned, c could not just be that witness. However, if it is a new constant, it can be interpreted as your wish. Then we would not be in trouble in this way. Simply, we can relax to a constant is not in current tableau.

It is important that we should always introduce a new constant when handle a $\exists$-style entry. Consider the following example.

Example 1. *Is $(\exists x)R(x) \to (\forall x)R(x)$ valid?*

Proof. With a semantic approach, we know that it is false for the truth that some member has a property does not mean that all members have the same property. But if we use tableau proof, we could prove it by wrongly using $\exists$-style tableaux. Just show you the proof as shown in Figure 2.

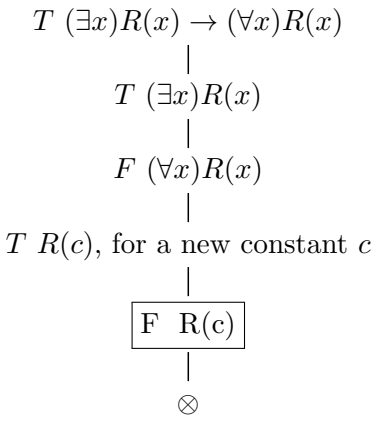

Figure 2: A wrong example about $\exists$-style

This is a tableau proof. But we should be aware the entry within a box. Actually, it should also introduce a new constant. Then the tableau is noncontradictory. So when a $\exists$-style tableau is applied, you must introduce a new constant never ever occurs before. □

Definition 2. *Tableaux from S. The definition for tableaux from S is the same as for ordinary tableaux except that we include an additional formation rule:*

2. *If τ is a finite tableau from S, φ a sentence from S, P a path on τ and $\tau\prime$ is obtained from τ by adjoining $T\varphi$ to the end of the path P, then $\tau\prime$ is also a tableau from S.*

3 Tableau proof

In predicate logic, there are formulas and sentence. Here we only consider the proof system for a sentence, a special formula without free variables. We only consider the tableau proof of *sentence*. With these atomic tableaux, we can define tableau and corresponding properties similarly as in propositional logic.

Definition 3. Tableau proofs (from S)*: Let τ be a tableau and P a path in τ.*

1. *P is* contradictory *if, for some sentence $\alpha, T\alpha$ and $F\alpha$ both appear as labels of nodes of P.*
2. *τ is* contradictory *if every path on τ is contradictory.*
3. *τ is a* proof of α *(*from S*) if τ is a finite contradictory tableau (from S) with its root node labeled $F\alpha$. If there is proof τ of α (from S), we say α is provable (from S) and write $\vdash \alpha$ ($S \vdash \alpha$).*
4. *S is* inconsistent *if there is a proof of $\alpha \wedge \neg\alpha$ from S for some sentence α.*

In proposition logic, every signed proposition is reduced exactly only once. Here, we should pay attention to sentence leading with quantifiers. For sentence $(\exists x)\varphi(x)$, it contains no more information than $\varphi(c)$. Then we can claim to have finished it. However, $\varphi(t)$ far from exhausts the information of sentence $(\forall x)\varphi(x)$. It just give us one instance asserted by entry $T(\forall x)\varphi(x)$. With this in mind, we can define reduction and finish of an entry in $\forall$-style.

Definition 4. *Let $\tau = \cup\tau_n$ be a tableau (from S), P a path in τ, E an entry on P and ω the i^{th} occurrence of E on P (i.e., the i^{th} node on P labeled with E).*

1. *ω is reduced on P if*
   (a) *E is neither of the form $T(\forall x)\varphi(x)$ nor $F(\exists x)\varphi(x)$ and , for some j, τ_{j+1} is gotten from τ_j by an application Rule (ii) of Definition 1 to E and a path on τ_j which is an initial segment of P. (In this case, we say that E occurs on P as the root entry of an atomic tableau.)*

   or

   (b) *E is of the form $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x), T\varphi(t_i)$ or $F\varphi(t_i)$, respectively, is an entry on P and there is an $(i+1)^{st}$ occurrence of E on P.*
2. *τ is finished if every occurrence of every entry on τ is reduced on every noncontradictory path containing it (and $T\varphi$ appears on every noncontradictory path of τ for every φ in S). It is unfinished otherwise.*

Example 2. *Check the following formulas:*

1. $((\forall x)\varphi(x) \to (\exists x)\varphi(x))$.
2. $(\forall x)(P(x) \to Q(x)) \to ((\forall x)P(x) \to (\forall x)Q(x))$

In previous examples, all tableaux are finite. However, it is only a small part of them. Consider the following example.

Example 3. *Is the sentence* $(\exists y(\neg R(y,y) \vee P(y,y)) \wedge \forall x R(x,x))$ *invalid?*

Solution. If a sentence φ is invalid, then $\neg\varphi$ must be valid. So we can have the tableau proof in figure 3.

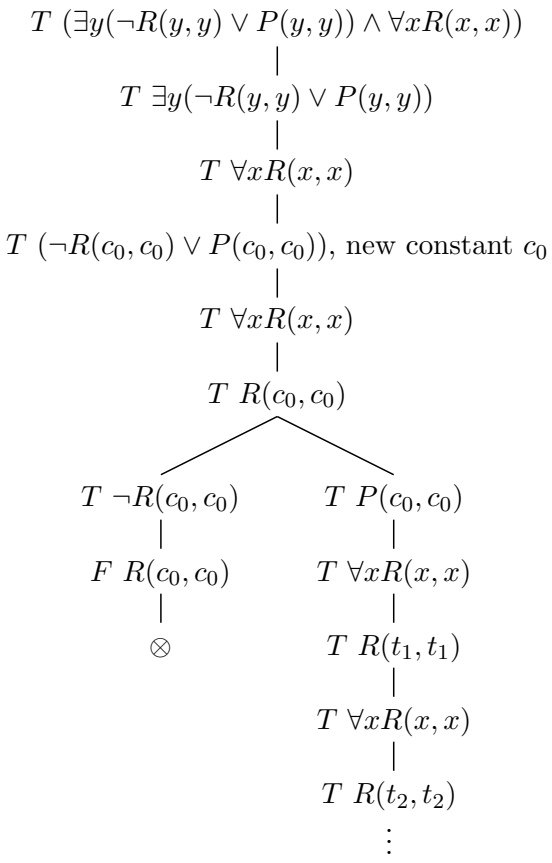

Figure 3: An example of tableau proof

The tableau is not contradictory. So the sentence is not invalid. □

Obviously, no contradiction could be introduced in the right branch. For $T(\forall x)R(x,x)$ exists, we can always generate new unreduced entries. So it is a infinite path. Then, when should we terminate reduction procedure?

We observe that we are reducing the same set of entries periodically without introducing any valuable information in the non-contradictory path in previous example. With this observation, we could terminate the reduction on that path. Generally, we always reduce those $\exists$-style entries first. If a $\forall$-style entry use all constants generated by $\exists$-style entries on that path, we could also terminate reduction on that path.

To prove a formula, we should transform it into a sentence by adding universal quantifier closure in front of it and then apply tableau proof of sentence.

Example 4. *Check the statement* $\{(\psi(x) \to (\exists x)\varphi(x))\} \vdash (\exists x)(\psi(x) \to \varphi(x))$

4 Complete systematic tableau

As we have shown, ∀-style may cause an infinite path for we can always try to reduce a node just generated. So we also need a mechanical way to guarantee every node could be reduced.

Similarly, we introduce complete systematic tableau. In it, we define how to introduce ground term t_i and new constant c_i. If we violate the rule, we may get wrong result.

Definition 5. *We construct the CST, the complete systematic tableau, with any given signed sentence as the label of its root, by induction.*

1. *We begin with τ_0 an atomic tableau with root the given signed sentence. This atomic tableau is uniquely specified by requiring that in Cases 7a and 8b we use the term t_i and that in Cases 7b and 8a we use c_i for the least allowable i.*
2. *If E is not of the form $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$, we adjoin the atomic tableau with apex E to the end of every noncontradictory path in τ that contains ω. For E of the form $T(\exists x)\varphi(x)$ or $F(\forall x)\varphi(x)$, we use the least constant c_j not yet appearing in the tableau.*
3. *If E is of the form $T(\forall x)\varphi(x)$ or $F(\exists x)\varphi(x)$ and ω is the i^{th} occurrence of E on P we adjoin*

$$\begin{array}{c} E \\ | \\ T\varphi(t_i) \end{array} \qquad \textit{or} \qquad \begin{array}{c} E \\ | \\ F\varphi(t_i) \end{array}$$

respectively, to the end of every noncontradictory path in τ containing ω.

If premises S is introduced, it becomes more complicated. Because there may be infinite sentence in S. We need determine when the sentence could be introduced into the current path. For example, we can introduce one in the odd stage, if the root is defined as 0.

Here, we strictly number all newly introduced constant uniformly. With premises S, we could split reduction into two stages, odd and even. For example, we can reduce entry in tableau at odd stage and introduce sentence in S at even stage.

5 Properties

With CST, we can guarantee that every tableau is finished. Even without premises, the tableau proof of a sentence still could be an infinite procedure because of ∀-style entry and semantics of $\forall x\varphi(x)$.

If it is a contradictory tableaux, which means a tableau proof or refutation, the CST is also finite. It means that a proof could always terminate in finite steps.

Exercises

1. Let φ and ψ be any formula either with no free variables or with only x free as appropriate. Give tableau proofs of each the following.

(a) $(\forall x)(\varphi(x) \vee \psi(x)) \leftrightarrow (\forall x)\varphi(x) \vee (\forall x)\psi(x)$

(b) $(\exists x)(\varphi \to \psi(x)) \to (\varphi \to (\exists x)\psi(x))$, x not free in φ.

(c) $(\exists x)(\varphi(x) \to \psi(x)) \to (\varphi(x) \to (\exists x)\psi(x))$.

(d) $((\exists x)\varphi(x) \to \psi) \to (\forall x)(\varphi(x) \to \psi)$, x is not free in ψ.

2. Let φ and ψ be any formula with free variables x, y and z; let w be any variable not appearing in φ and ψ. Give tableau proofs of the following.

   (a) $\exists x \forall y(\forall z\varphi \vee \psi) \leftrightarrow \exists x \forall y \forall w(\varphi(z/w) \vee \psi)$.

   (b) $\forall x \exists y(\varphi \to \forall z\psi(z)) \to \forall x \exists y \forall w(\varphi \to \psi(z/w))$.

3. Let $\varphi(x_1, \ldots, x_n)$ be a formula in a languange $\mathcal{L}$ with all free variables displayed and let $c_1, \ldots, c_n$ be constant symbols not in $\mathcal{L}$. Show that $\forall x_1 \ldots \forall x_n \varphi(x_1, \ldots, x_n)$ is tableau provable if and only if $\varphi(c_1, \ldots, c_2)$ is.