CS 512 Formal Methods, Spring 2017	Instructor: Assaf Kfoury

First Order Tableau: Ground Instantiation Method

April 06, 2017 Rawane Issa

(These lecture notes are **not** proofread and proof-checked by the instructor.)

Initial few words on FOL definability

In the problem sets you will be asked to define in first order the following conjectures/theorems (note that you are not asked to prove any of these statments. "Conjectures" by definition are conclusion based on incomplete information that have thus far no been proven. Moreover, Fermat's Little Theorem and the Four color theorem is are both non trivial results that took centuries to prove).

1. **Twin prime conjecture.** No matter how far you go along the line of the naturals, you will always find two primes separated by exactly one composite.
2. **Goldbach's conjecture.** Every even number bigger than 2 can be expressed as the sum of two prime numbers.
3. **Fermat's Last theorem.** This theorem states that there are no three positive integers x, y, z such that $x^n + y^n = z^n$ for all $n > 2$. There are infinitely many solutions for the cases of $n = 1$ and $n = 2$. Note that for $n = 2$ this is simply the Pythagorean theorem.
4. **Four color theorem.** We need no more than 4 colors in order to color any map divided into contiguous area. This theorem, like the previous one, was a conjecture for a very long time. It was eventually proved by computer scientist via exhaustive search.

There are several proof assistants that assist in formally proving mathematical statements. Notable mentions include Isabelle and Coq.

FOL Tableau

The methods we develop for FOL tableau will include all the expansion rules and remarks we've made for Classical PL tableau. Note that when we say classical propositional language, then we mean that the axioms of our logic will include the law of excluded middle, contrary to other forms of propositional logic such as intuitionistic PL.

In FOL we will have to carefully look at wffs which are not necessarily closed (i.e. which are not necessarily sentences): we did not have to worry about this in propositional logic (as PL is not quantified) and only encountered this in QBF. However, this is not a limitation because free variables are implicitly universally quantified as we will see later on.

There are two methods we can use in order to extend tableau to FOL:

1. Using ground instantiation,via elements from Gilmore's algorithm and material from Herbrand Theory.
2. Using Unification theory: You have probably used many software tools which use unification behind the scenes without bringing it to your attention. e.g: The query engine of SQL and other relational databases uses unification; Programming languages such as Java, ML,C# and Haskell use unification in their type inference engines.

In both approaches we will start with a finite set of first order sentences Γ in PNF: remember that we can write any wff in PNF and get an equivalent (not necessarily equal) formula *(equivalent in the sense that both have the same models which satisfies them).*

Ground Instantiation

The first method will use many of the elements we have already encountered in Herbrand Theory.

We will use all the expansion rules from classical PL (Handout 9) and add two new rules to them: one for each quantifier. Remember that the tableau method is a downward growing tree: In PL we learned that if we are lucky and apply the rules of tableau "cleverly", then we will be able to close all paths in the tree and conclude that the set of wffs which we started from is unsat (and its negation is valid).

The two new rules that extend the previous tableau method to FOL are:

1. $\dfrac{\forall x\ \varphi(x)}{\varphi[x := t]}$ This rule replaces the quantified variable x with a ground term.
2. $\dfrac{\exists x\ \varphi(x)}{\varphi[x := c]}$ This rule replaces the quantified variable x with a fresh new constant symbol.

These simple expansions will create fundamental issues in FOL tableau that we did not have in PL.

Complications of this method

Several complications arise from this seemingly simple extension of tableau to FOL:

1. The universal quantification rule is non deterministic: there are infinitely many ground terms that we can pick from and replace x with. If we have the constant symbols a, b and the unary function f , for example, then our set of ground terms will be $\{a, b, fa, fb, ffb, ffa, \ldots\}$. Imagine that we also add a binary function g into the picture, our set of ground terms will grow even bigger.
2. A natural conclusion that we can make is that we need to pick our ground terms "cleverly" via some heuristic in order to avoid scenarios where this method does not terminate (as we will see in one of the examples).
3. The existential quantifier is deterministic: it only requires us to replace x with a new fresh constant symbol. However it presents its own complications as we will soon see.

4. In PL, we could always go down a path and apply a rule exactly once. In FOL however we cannot guarantee this: we cannot be "strict" in this sense and cannot take advantage of "work" that we've already done because of the implication of the universal quantification rule. In some instances we will have no way around applying a rule more than once. Some sets of wffs will actually have no tableau which is not strict.

5. Another source of complication is that the existential quantifier increases the size of the set of ground terms by extending it with a fresh new constants symbol along with all its related ground terms: If we apply the existential quantification rule, then we will have to deal with a bigger set of ground terms when dealing with the expansion rule for the universal quantifier. How can we control this ?

Few words on Soundness and Completeness

An important remark to make is that these manipulations are made on a syntactic level (via pattern matching): it is therefore natural for us to call the tableau method a proof method.
We will also need to study the Soundness and Completeness of this method:
Are these additional rules sound? It should intuitively seem so since the ground terms are elements from the universe, and the quantification happens over all elements in the universe.
How about completeness ? We can show from Herbrand theory that we only need to worry about the ground terms in order to answer this question.

Examples.

In this section, we will remark on the examples found in Handout 26.

If we look at the example on page 7, we first notice that the given example is not an arbitrary/blind application of tableau. The picked ground terms for the existential quantifier are clearly based on a pre-existing idea of how we would eventually close each path in the tableau. There is a lot of "looking forward" being applied in order to efficiently use tableau and avoid scenarios similar to the one presented on page 12. On the simple example presented on page 12, we were immediately presented with issues resulting from blindly applying the expansion rules because our expansions did not permit us to discover the inherit contradiction in the initial set of wffs (notice that in order to find the contradiction, we must bring down one of the premises, which the example does not do). The moral of the story is that if we automate (write a program for) the tableau method and apply the expansion rules arbitrarily to a set of wffs, our PC might run into these kinds of infinite loops. While it might be "easy" for us to look forward and cleverly pick the expansion rules, formally defining these heuristics for a computer are possible but may not be as easy.

Also notice that this example is not strict since we chose to apply the universal quantifier rule to the same wff twice in order to eventually close all paths. Further note that example on page 7 has no tableau which is strict (remember that the tableau of a set of wffs is not unique and depends on the order in which one applies the expansion rules).

Unification

In the second method, we will try to avoid some of the complications that arise from the first method:

1. We will delay the use of the universal quantifier as much as possible
2. Contrary to the previous method, we will instantiate $\forall$ via a new fresh variable and not a ground term.
3. Contrary to the previous method, we will have free variables that we will implicitly think of as universally quantified outside of this method.

Some intuition for next lecture

Unification is a big area invented by CSist. It has many variations, one of which is known is called matching problem. Examples of what unification is:

Suppose you have a function symbol f applied to a, and another function f applied to x. We can unify both of these application via the function $\sigma(x) = a$.
Given two different parse trees of function applications, can we unify both tree? We match (special case of unification) and substitute elements in both trees until both become the same tree, or until we can tell that both trees can't be unified.