Planning and Reasoning

Sveva Pepe

Thursday 3rd December, 2020

1 First Order Logic

FOL is logic about objects, it is like set of statements about objects. We can say that certain proposition, condition over some object could be true or false.
The basic idea is that we have *terms*, like constants, variables and functions. A function is something that gives an object from other object, for example "the owner of a car". Everything build from variables, constants anf function symbol is a term. So, if we have a subformula only made by function symbols, constants and variables is a representation of an object, element of the domain. Then we build proposition using terms, boolean conditions; for instance, we can say that certain condition P is true over two object.

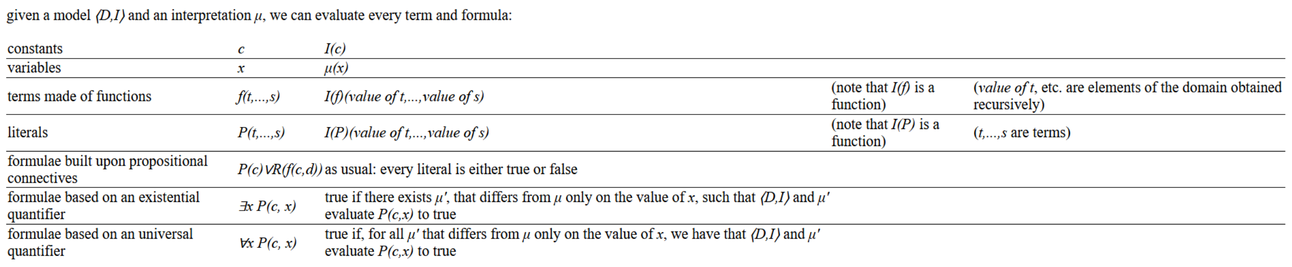
1.1 Semantics

We have a formula and we want evaluate if it is satisfiable or not.

Is there any way or any possible interpretation of the variables, function and constants that makes this formula true?

We only give the formula, which is the representation some pieces of informations about these objects, their relationships and their propositions (predicates). So, we have statements and we want to check whether a statement is valid. The idea is that we use a model.
In propositional logic a model is an assignement to variables and since in a propositional logic a variable could be true or false, the model is simply an assignement of true or false of the variables. In FOL things are much more complicated because we have set of all objects (domain of the formula) and then we have things that represent object (variables, constants and functions symbols). So, we have to give an interpretation for each of them. If we have an interpretation we need to determine the values of every possible interpretation of an object. Every terms that represent an object the interpretation should give me enough information for telling exactly which object is. This depends on the interpretation and the interpretation includes both the elements of the domain and also the way the elements are associated to terms, which object is the value of every possible term; and again for every formula the interpretation gives me enough information that allow me to compute the value of a formula (if formula is true or false).
The formal definition of an interpretation is quite complicated because actually an interpretation contains the domain and then we need the values for the constants and the terms. We split the values of model and an interpretation. A model is the value for constants and function symbols, while the interpretation contains the variables. At this level is not cleare the difference between constants and variables, so the levels of the term are exactly the same, so constant is something that represent an object and again a variable is something that represent another object. The only difference is that a variable is something which we may include it in a quantifier ($\exists x$, where x is a variable), instead a constant is fixed cannot be quantified.
An interpretation gives the values of the variables, so what's needed for quantifier, instead, a model only contains everything else but the variables, so domain and the way we can interpret the value of every terms, every formula given the values of the variables as well.
For each constants the domain tells the value of the constant, so the object associated to the constant (object that constant represent). Then the meaning of the function, of course a function symbol, tells us that the value of this is a function from n object to another object and again, it is also an assignement to the predicates.

Check in the table below that for a constant we have the interpretation of a constants in the model, then for each variable, its object is in the interpretation and not in the model. Then given a term $f(t, ..., s)$ so a function apply to some another terms, we can simply check in the model that given the values of the terms we can apply the function and obtain a new values. Same for proposition (literals). For the quantifier, we simply take the interpretation, we replace the value of the variable x with another value if there exists a replacement that allows to make the subformula true. The same for the universal quantifier.


given a model $\langle D,I\rangle$ and an interpretation μ, we can evaluate every term and formula:

constants	c	$I(c)$		
variables	x	$\mu(x)$		
terms made of functions	$f(t,\ldots,s)$	$I(f)$(*value of* t,...,*value of* s)	(note that $I(f)$ is a function)	(*value of* t, etc. are elements of the domain obtained recursively)
literals	$P(t,\ldots,s)$	$I(P)$(*value of* t,...,*value of* s)	(note that $I(P)$ is a function)	($t,\ldots,s$ are terms)
formulae built upon propositional connectives	$P(c) \vee R(f(c,d))$	as usual: every literal is either true or false		
formulae based on an existential quantifier	$\exists x\ P(c, x)$	true if there exists μ', that differs from μ only on the value of x, such that $\langle D,I\rangle$ and μ' evaluate $P(c,x)$ to true		
formulae based on an universal quantifier	$\forall x\ P(c, x)$	true if, for all μ' that differs from μ only on the value of x, we have that $\langle D,I\rangle$ and μ' evaluate $P(c,x)$ to true		

A formula is satisfiable if there exists a model and an interpretation making it true.
If all variables are bound then actually the interpretation is irrelevant because in the semantics of FOL we have this concept of changing the value. When we have exist what we do is that we check where a change of the interpretation, a change of the value of the variable in the interpretation makes the subformula true, and if we do this for all variables then the original value of the interpretation is irrelevant because we are changing that in all possible ways. Basically the interpretation is mostly useful for giving a semantics to the quantifier rather than being really a set of values that it is used to evaluate them. It is like a partial structure rather than something that it is just needed.
Of course the name of variables in the quantifiers is irrelevant, for instance, write $\exists x P(x)$ is equivalent of $\exists y P(y)$.
Two properties of quantifier needed for tableau method is:

$$\forall x(A \wedge B) \equiv A \wedge \forall x B$$

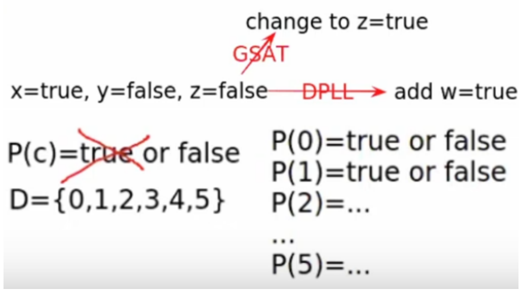
$$\forall x(A \vee B) \equiv A \vee \forall x B$$

1.2 Tableau for FOL

The idea is that it is not practical to extend the backtracking method to FOL because in the backtracking method (also GSAT) we are trying the evaluation of the variables, trying the evaluation of the truth values. In both cases we have something like this:

$$\{x = true, y = false, z = false\}$$

In GSAT we start with an incomplete interpretation and then we try to change, for instance z, with another value ($z = true$). So, what we do is that we try to change the value of the variables. In DPLL, instead, given a partial interpretation we extend this example with another value, like *add* $w = true$.

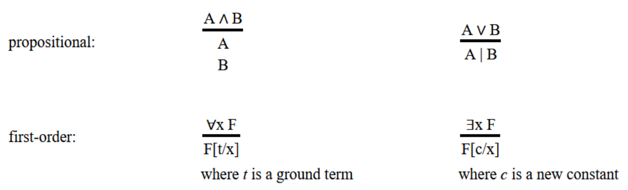


We can see that in GSAT we change the value of variable and in DPLL we just add values recursively.
The problem of apply these two methods in FOL is that in FOL we have potentially an infinite number of truth values. What we evaluate to true or false is, for instance, $P(c) = true$ or $false$. So, the problem is that this is not just one value because for example if the domain by $D = \{0, 1, 2, 3, 4, 5\}$ (6 number) this means that, in order to provide a model for formula, we cannot just say that something is true or false but we need to specify everything regarding to this. The problem is that with just one predicate we can simply choose a domain that it is so large that we need large number of assignement. Actually there is not constrants about that D should be finite, so it could be also be infinite, so in this case it is impossible to provide an evaluation in terms of just enumerating of true or false value.

It is not just an assignement because in propositional logic we have 3 variables (x,y,z) and we have only 8 interpretations, so GSAT works by moving around this 8 interpretations. In DPLL we add values and when it reaches the number of variables we have DPLL stops, it cannot be undeterminate. In FOL the problem is that give you a formula, also a simple formula, like P(c), and if you can think at the domain it can be very large or infinite because the domain is not even fixed. The domain is something that it is in the model and if you want

to apply the same trick like in GSAT or DPLL the problem is that in this case the model does not contains only true and false but also the domain and this is not fixed.
We cannot iterate the truth value but we can simply just try to reconstruct that by trying to assign things to the constant and to the predicate. It is possible to establish satisfiability, not always; in fact, FOL in not decidable, so it is not complete algorithm to establish satisfiability but it still possible.

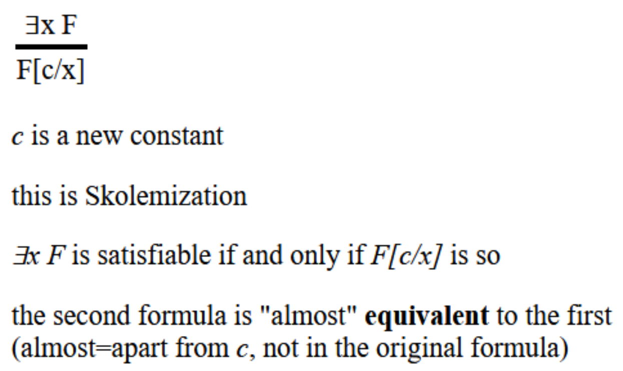
1.2.1 Rules Tableau - Propositional vs FOL



propositional:	$\dfrac{A \wedge B}{\begin{matrix}A\\B\end{matrix}}$	$\dfrac{A \vee B}{A \mid B}$
first-order:	$\dfrac{\forall x\ F}{F[t/x]}$ where *t* is a ground term	$\dfrac{\exists x\ F}{F[c/x]}$ where *c* is a new constant

For the case of $\forall x F$, F is always true and in particular, the formula is true by replacing x with arbitrary ground term (t)
Why this?
Because if the formula is always true, it will be true for the object that term represents. Every object is just a representation of a term (constants, variables etc).
Instead, in the case of $\exists x F$, so if there exists a value of x that makes F true, then it will be true for c (constant).

Existential quantification rule



$$\frac{\exists x\ F}{F[c/x]}$$

c is a new constant

this is Skolemization

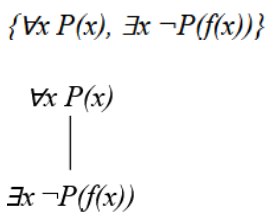
$\exists x\ F$ is satisfiable if and only if $F[c/x]$ is so

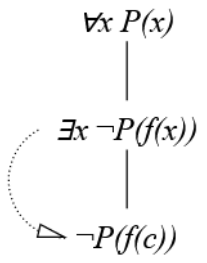
the second formula is "almost" **equivalent** to the first (almost=apart from *c*, not in the original formula)

The idea is that we are apply the *skolemization*.
Suppose to have a formula like this: $\exists x.P(x)$ and we want to prove that it is equivalent to a formula P(c), where c does not occurs everywhere else in the formula. $\exists x.P(x)$ means that there exists a value of x such that a model exists. So, actually it is model of an interpretation (formula is true) if there exists a model such that changing the value of x somehow, P(x) is true.
If we look at P(c) the meaning is that a model exists if it includes the value of c.

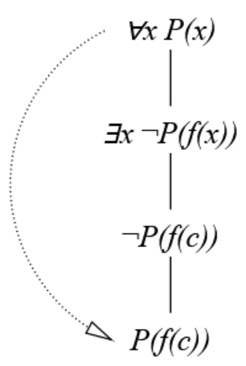
The idea is that basically the quantification $\exists$ ("there exists"), in the first formula means that we say if a model exists such that changing the value of x (there exist value for x) the formula is true. Instead, when we look at P(c) we can say that this formula is satisfied if there exists a model such that P(c) is true, but the model includes a value for c. The value for c, the value put insides the predicate, in this case comes from a model and in the previous case it is not included in the model but it is still obtained by changing the value of x in some ways. The meaning is preserved as long as we are checking that exists a model that makes the formula true.
There is, of course, the reason why the skolemization in the expansion of the Existential quantifier we need a new constant because we do not want to interfere with other constants that are already in the formula. Everytime we apply existential quantifier we add new constant that it is not present in the formula.

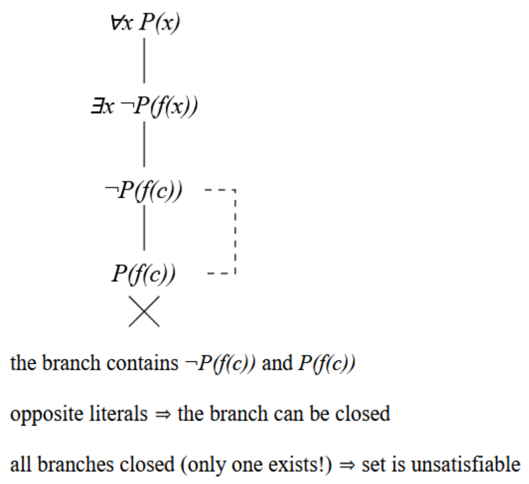
Example





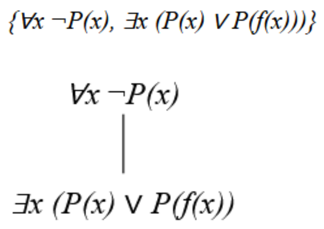
As we done for propositional logic we place the formulas one over the other and then we proceed. The first rule says that we have existential quantifier, so re remove the existential but we should also replace the x with new constant c.

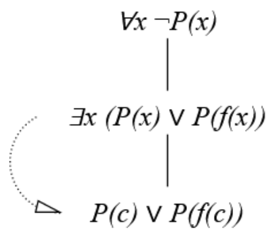


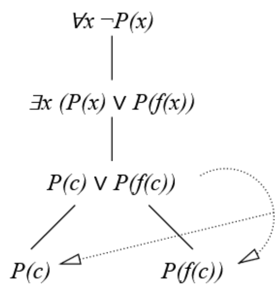

∀x P(x)
∃x ¬P(f(x))
¬P(f(c))
P(f(c))
the branch contains ¬P(f(c)) and P(f(c))
opposite literals ⇒ the branch can be closed
all branches closed (only one exists!) ⇒ set is unsatisfiable

After this we expand the universal quantifier and we can use every term that we want, for instance we can use c, f(c), f(f(c)). In our case we choose f(c) in place of x. We can see in the picture above that we have a contraddiction because the last two formulas are one opposite of the other. The branch is closed and it is the only branch in the tableau, so the formula is unsatisfiable.

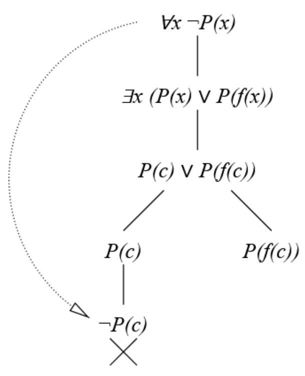
Second Example

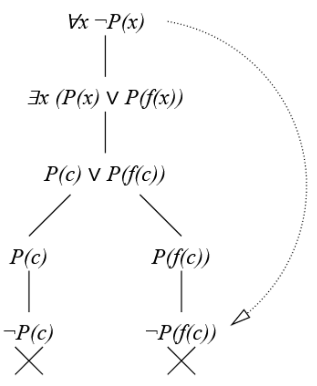






One of the rules of this version of the tableau method is that, in the universal quantifier, a variable can only be replaced by ground term. A ground term is a term that only contains function symbols and constant. You cannot use any variable, not replace variable with variable. So, the problem is that in this original formula we already have a function symbol (f) but we do not have any constant yet. So, we cannot apply ∀ rule, so what we do instead, is to expand the second formula, the existential quantification. We do this by introducing a new constant c. We split the formula because we have the disjuction and then since we have a new constant now, we can apply the universal quantifier. The rule for universal quantifier do not require the use of term that it is already in the formula, because if I want we can apply f(f(c)), for instance, but it is useless for this example.

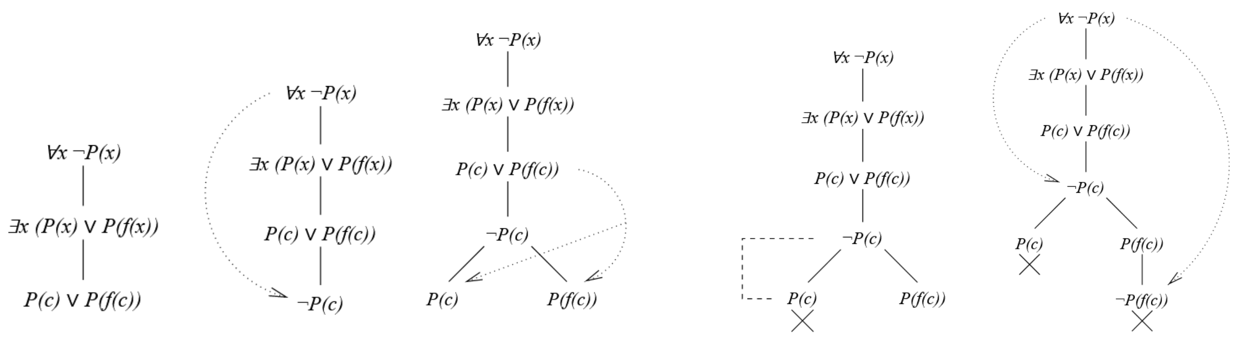




Since P is always false it is also false for c, so we apply the universal quantifier in the case of the term c and so we have a clash for that branch. Then we proceed with the other branch, so remember that we haven't finished yet because in the other branch we do not have expanded yet the universal quantifier. We replace x with f(c) an it is a valid possibility because the rule says that we can replace x with a ground term, so it is not just a constant but constants with function applies to them. We can see that we close also this branch and the formula is unsatisfiable.

The idea is that, while in the case of GSAT or DPLL we are trying hard to find a model that satisfied the formula, instead, in the tableau method we are basically doing the oppositive. We are trying hard to derive an unsatisfiability, so we try as much as possible to getting to a pair of contradictory literals. We expand the rule exactly with only value make contraddiction between literals, so we are trying exactly the only value that makes branch closed. We are trying hard to make a contraddiction. We are choosing the ground term to end up with contraddiction. Only if we fail we might perhaps says that formula is satisfiable. Actually this is not the case, not in FOL, we can almost never conclude that the formula is satisfiable we can only says that in some cases we can but typically it is not general possible.

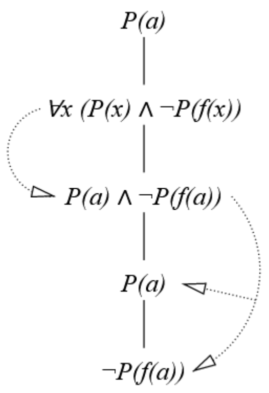
Alternative development

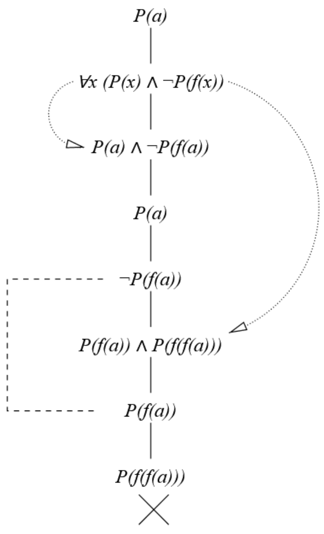


We decide to expand before the disjuction the universal quantifier because now we have the possible replacement for x, x could be replace by every ground term. We replace x with c and obtain $\neg P(c)$. We have a contraddiction of the first branch after applied the disjuction, so split formula.
What about now (we are in the above Figure 4 from the left)? Should we stop of not?
According to the propositional rules we should expand every formula in every branch, and we have 2 branches in this example and we have expanded all formulas but for the universal quantifier may need to be expanded in the same branch more than ones. So we expanded again and we finished with the above Figure 5 (from the left).
What we have do is a different policy of expansion. If the rules are not also keep with a policy of the application of rules, so, if I just says that rules are correct or rules are complete we should also come to the conclusions that which ever way I applied the rules I always end up with the same conclusion because otherwise it is not consistency the rule. What I instead says now is that the method of expansion is always the same and I can still do that in which ever order I want but there is a case that same formula may need to be expanded more than 1 time, only for universal quantifier.

An example that really cannot close without apply the same rule twice

{P(a), ∀x (P(x) ∧ ¬P(f(x))}

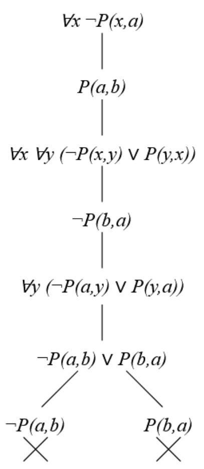




In the orginal formula we have constant "a" and function symbol "f". So, the ground terms are: a, f(a),f(f(a)) and so on. We try to replace x with a, then with f(a) and we obtain a contraddiction. Both expansion are needed to have closed branch.

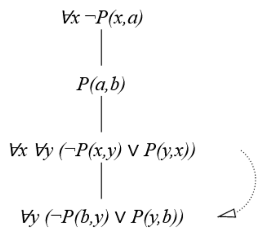
Another example

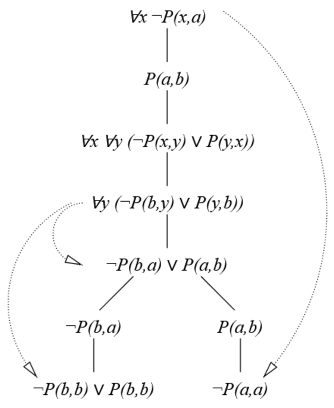
{ ∀x ¬P(x,a), P(a,b), ∀x∀y (¬P(x,y) ∨ P(y.x)) }



We apply some human intelligence to decide the terms the ground terms that you expanding in the universal quantifier formula. We would like instead, to fully automated the method.
The reason why we do this, so we apply the universal in this order because we understood that the meaning of this formula is an implication so we try to make one the opposite of the other and so then obtain a clash.

Another way to do the previous exercise





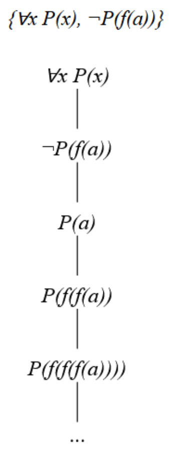
We can have a policy in which we can make the wrong choice, so we replace x with b and not with a. We proceed with a number of useless formulas. We wasted the time. Now we need to prove the unsatisfiability of two banches instead of just one.

We can see that the main problem is really how to apply the rule for the universal quantifier because the problem is that, for all remaining rules there is only one possible application (if we have a conjucntion we put formulas in column, in a disjuction we split and so on). These rules are deterministic.
Instead, for universal quantified, first, we have to choose a ground term and second, we might need to apply the same rule a number of times in the branch. It is not deterministic because we need might to choose a ground term and there might be an infinite number of ground terms in the tableau and we might need to apply the same rules several times with different ground terms. This is the real problem. One of the things that we can do is to apply randomly the universal quantifier but we might end up with an increase difficulty. Another thing that you can do is that we can apply the other rules first and then the universal quantifier.

Infinite expansion
The problem is that it can be proved that this method is correct, so if the method end up with the tableau that has a closed branches the formula is unsatisfiable. But in the other direction is not exactly like this because there is a theorem that proves that if a formula is unsatisfiable then there is closed tableau, so a tableau with all branches that are closed. There is one way but it doesn't say how, so it didn't say each ones rules.
What can be said is that if I keep apply the wrong rule I never closed the tableau, instead, if we do not do this the tableau can be closed. In fact, in the previous example if we replace x with f(a) we closed the tableau, but we do not do it so it never closed. So, given a term t and branch of a tableau containing $\forall x F$, the policy at some point will expand the formula using t.

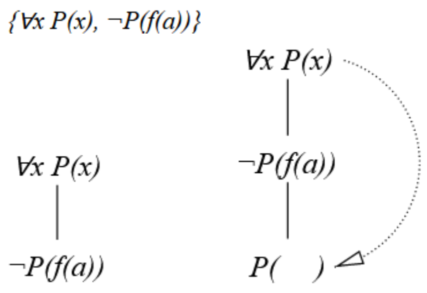
1.3 Tableau with unification

There is a potentially large number of terms that you could use. If I only have constants it is easy because we the only possible ground terms are just the constants; for instance 10 constants 10 possible ground terms. One function symbols is enough for infinite number of ground terms because even we have function symbol of arity 1 we can apply it as many times as we want. Just with 1 constant and 1 function symbols we end up

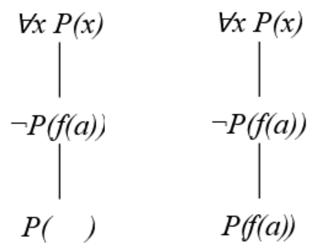


In the figure we can see that since the rule for universal quantification can be applied an unbounded number of times, it may lead to a forever-expanding tableau. There is no way to close the tableau.

with potential infinite number of ground terms and each one can be used for replacing an universal quantifier variable. So, we have an infinite number of possible expansion of every universal quantification. One function symbol and one universal quantifier formula means infinite number of possible expansion. This is the problem. One of the problem in some cases we may even need to apply the expansion twice, etc; and the other problem is to identify which terms are actually useful because not all expansions are really useful. The problem of choosing is solving by delaying the application of the rule or we can apply the rule now and choose the ground term later. This is not how the rule works, but this is the principle.

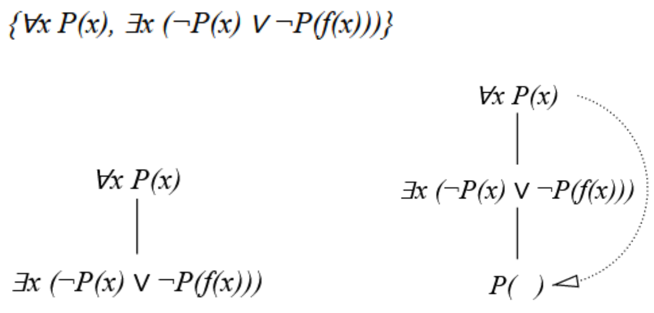


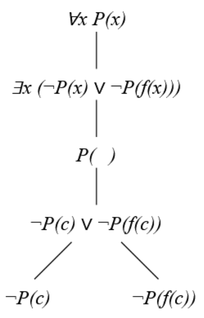
We expand the formula but we do not choose the ground term. We leave an "empty" pair of parentheses, which means that it will be a ground terms but we will choose it later.



Now we could close the branch.
How?
We didn't specify ground term so I could choose any ground term I want . The idea is that in the parentheses I put something that makes P( ) the opposite of $\neg P(f(a))$. So, I know what I could use now, I immediately know but this is just by chance because at this point I could just immediately choose what to put inside the parentheses. Now I can just close because I have a pair of contraddictory literals.

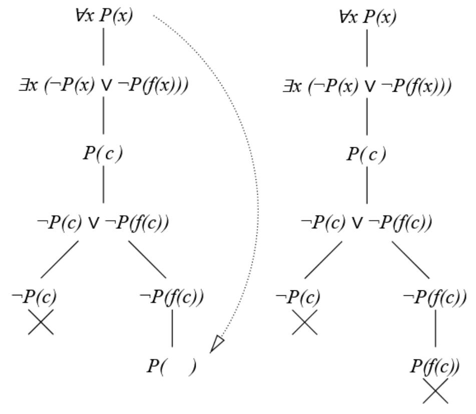
Another example - usefulness delay choice is evident





I presume that at some point in the future I will add a ground term, so we have P( ). Then I do the expansion

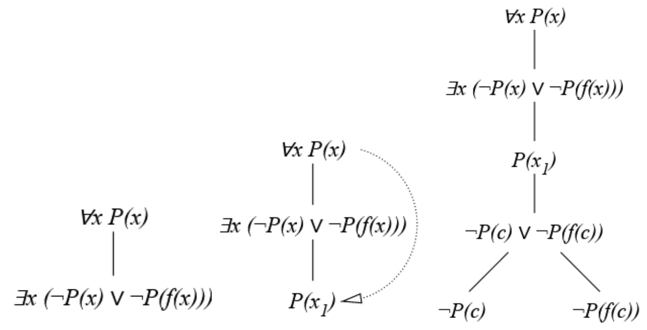
of the existential quantifier formula and disjuction (above figure 3 from the left). Now we can see that I can decide what I should put in the parentheses.
Why?
Because, for example, if I put c I close the first branch (from the left).
So, I just delay the choice, this is the exactly the same tableau that would result by repling x with c. It is just to make the decision at the later point when I can decide. Instead of just going randomly, I delay the point where the decision is easy because it easy to see what is useful at this point.



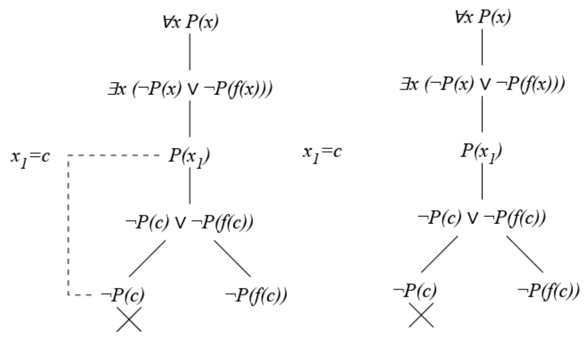
In the picture in the left, you see again that I do not decide immediately, I do the replacement with empty parentheses but now at the second step it is easy to understand what I need to choose to close the branch. The final tableau is the right picture.

Of course in the actual application of this method I do not do the expansion in this way, I always delay the application of expanding the universal quantifier but still, the point is that if I apply the wrong policy of the application of the rules I still can close the tableau because nothing is really wrong at this point. Since I do not choose the term, I actually not making any choice, I do not make any commitment but I'm just deciding later what to put and the method still works.
There is a problem which this method that I choose. I expand and I leave it "empty" but I need a formal semantic because otherwise I do not know how to expand formula like $\exists y P(\ ,y)$ or P(,). What I mean is that P( ) is replace by ground term, so P(t) for instance but I do not fix the ground term yes. But since the ground term and variables both represent values of the domain, instead of leaving an empty space I use a variable. So, the variable is something which I assume will be evaluated in the same way as the ground term will be later use for that empty space.

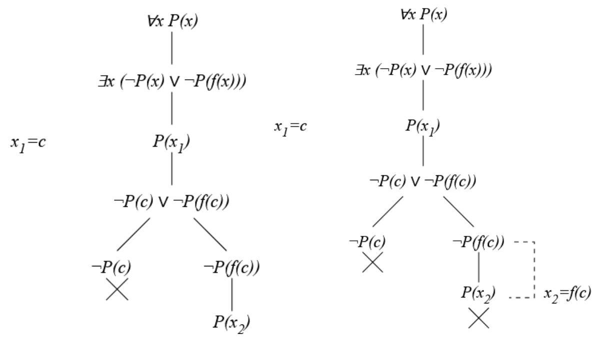
Example variables



Everytime I expand an universal quantifier variable I do this by a new variable and the variable will be a place-holder (representation) of the term that will be choose later. We should this instead of empty space. We obtain $P(x_1)$, then we expand the existential quantifier and do the disjunction.



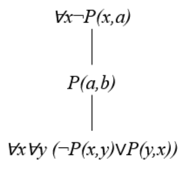
I can make the subformula $P(x_1)$ the opposite of P(c) by replacing x_1 with c, so by deciding that x_1 is exactly a placeholder for c. Instead of filling the empty spacing I use a variable and then I decide the value of the variable where this value is a ground term.
$x_1 = c$ we close the first branch and than proceed. But now the point is that, since I have decided the value for x_1 we cannot replace now $x_1 = f(c)$, this is forbidden because x_1 is already c now, it doesn't matter that it is in another branch so x_1 is fixed.

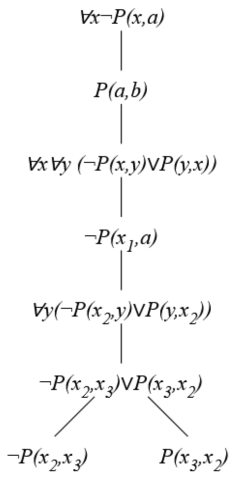


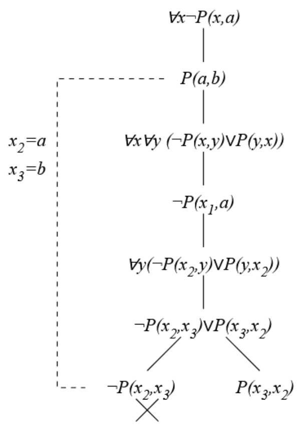
What I can do instead?
We make another expansion and I have a new variable now, x_2. Now we can set $x_2 = f(c)$ and we close the second branch.
I have two expansions with 2 different placeholder variables and I give them two different values. The point is that know I delay the choice of the ground term at to the point in this is really know to be useful.

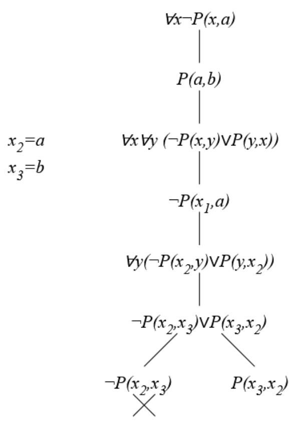
Example of rigid variables

{ ∀x ¬P(x,a), P(a,b), ∀x∀y (P(x,y) → P(y,x)) }

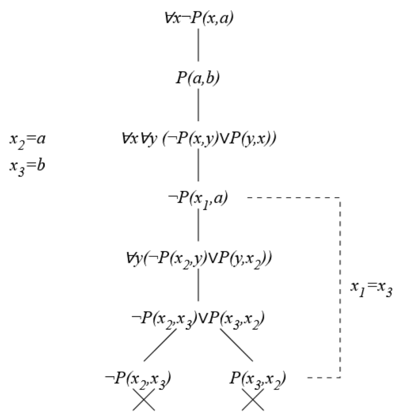








We can make a contraddiction just by making $x_2 = a$ and $x_3 = b$, see above figure 3 (from the left).



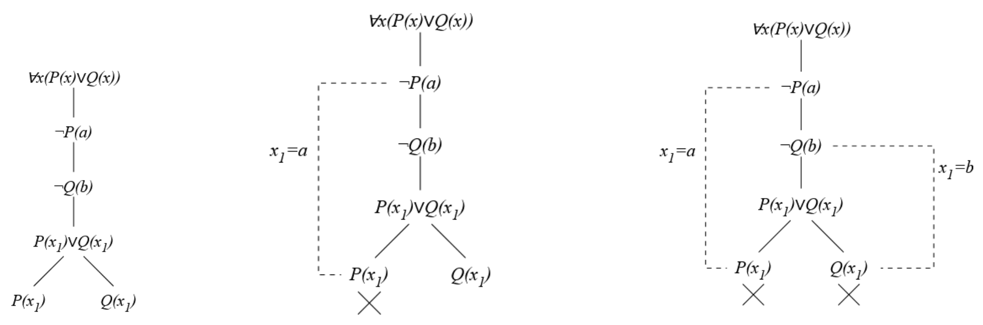
When I do this something that do not hold only in this branch x_2 and x_3 come out in all tableau. So, in the second branch we have $P(b, a)$. This is the called rigidity of variable, so a variable one is chosen has that value in all tableau and not in just one branch. And by choosing $x_1 = x_3 = b$ we have a clash also in the second branch.

Example that violating rigidity of variables and may lead to unsoundness, the tableau method may end up with wrong result

{ ∀x(P(x)∨Q(x)), ¬P(a), ¬Q(b) }

This formula is satisfiable, the model is:

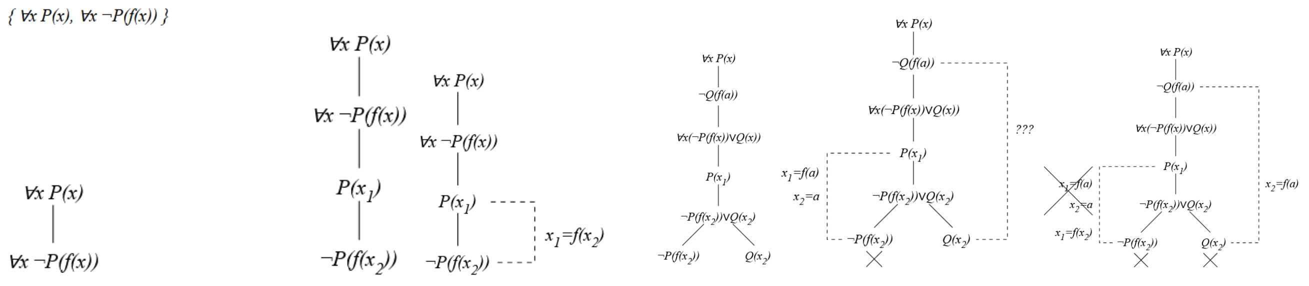
$$D = \{0, 1\} \qquad a \to 0, \qquad b \to 1$$
$$P(a) = P(0) = 0 = false, \qquad P(b) = P(1) = 1 = true$$
$$Q(a) = Q(0) = 1 = true, \qquad Q(b) = Q(1) = 0 = false$$



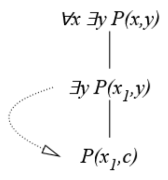
If I do not follow the rule of rigidity, you can see that this tableau closed all its branches. This is wrong because the set is not unsatisfiable.

It is also possible to avoid choosing exactly a wrong term so, instead of $x_1 = a$ I could choose $x_1 = f(x_2)$. So, this way I can still choose x_2 to leave a part of the ground term unfixed, so x_2 can be chosen later when I will be useful.

Example unclosed terms



1.3.1 Skolemization with unification



The skolemization changes when I have this kind of free variable, so I cannot just say that if I have to expand $\exists y.P(x_1, y)$ I cannot do this just by replacing y with a constant but I need to apply a function, a new function. So, we need to replace y with $f(x_1)$ where f is a function symbol.