Completeness of First Order Logic

Gianluigi Bellin

March 24, 2004

1 Propositional Calculus

1.1 Syntax and Semantics.

A language $\mathcal{L}$ for the propositional calculus is given by a (possibly infinite) list **Atoms** of *propositional letters* p_1, p_2, ...

Formulas. The *formulas* of the language are given by the grammar

$$A \;:=\; P \mid \neg A \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid A_0 \rightarrow A_1$$

$A \equiv B$ is defined as $(A \rightarrow B) \wedge (B \rightarrow A)$.

Note: A propositional letter p stands for a sentence, e.g., "*I am bored*". The thought expressed by a sentence is called a proposition. Propositions (and sentences that express them) are entities that can be true or false. The truth or falsity of a molecular statement depends uniquely on the truth or falsity of the atomic components, i.e., propositional connectives are *truth-functional*.

An *interpretation* or *valuation* of a propositional language $\mathcal{L}$ is a total function $\mathcal{V}$: **Atoms** $\rightarrow \{T, F\}$, where "T, F" stand for the truth values *true* or *false*. We extend valuations from atomic to molecular formulas inductively as follows:

1. $\mathcal{V}(\neg A) = T$ if it is **not** the case that $\mathcal{V}(A) = T$; $\mathcal{V}(\neg A) = F$ otherwise.
2. $\mathcal{V}(A_0 \wedge A_1) = T$ if $\mathcal{V}(A_0) = T$ **and** $\mathcal{V}(A_1) = T$; $\mathcal{V}(A_0 \wedge A_1) = F$ otherwise.
3. $\mathcal{V}(A_0 \vee A_1) = T$ if $\mathcal{V}(A_0) = T$ **or** $\mathcal{V}(A_1) = T$; $\mathcal{V}(A_0 \vee A_1) = F$ otherwise.
4. $\mathcal{V}(A_0 \rightarrow A_1) = T$ if either $\mathcal{V}(A_0) = F$ or $\mathcal{V}(A_1) = T$; $\mathcal{V}(A_0 \rightarrow A_1) = F$ otherwise.

Definition. A propositional formula is *satisfied* [*falsified*] by a valuation $\mathcal{V}$ if $\mathcal{V}(A) = F$ [$\mathcal{V}(A) = F$]. A proposition A is *valid* (or a *tautology*) if for all valuations $\mathcal{V}$ we have $\mathcal{V}(A) = T$, in symbols $\models A$. A proposition A is a *contradiction* if $\mathcal{V}(A) = F$ for all valuations $\mathcal{V}$. A proposition A is *logical consequence* (ot *valid consequence*) of a set of propositions Γ (in symbols, $\Gamma \models A$) if for every valuation $\mathcal{V}$, we have $\mathcal{V}(A) = T$ whenever $\mathcal{V}(C) = T$ for all C in Γ. Two propositions A and B are *logically equivalent* if $A \models B$ and $B \models A$.

Exercise 1: Verify the following facts:

$$\models \neg\neg A \equiv A \qquad \models \neg(A \wedge B) \equiv (\neg A \vee \neg B) \qquad \models \neg(A \vee B) \equiv (\neg A \wedge \neg B)$$

$$\models \neg(A \to B) \equiv (A \wedge \neg B) \qquad \models (A \to B) \equiv (\neg A \vee B)$$

We say that a formula is in *negation normal form* if it is generated by the following grammar:

$$A \ := \ p \mid \neg p \mid A_0 \vee A_1 \mid A_0 \wedge A_1$$

Exercise 2. Using Exercise 1, show that for every formula A in $\mathcal{L}$ there exists a formula A' in negation normal form such that $\models A \equiv A'$.

1.2 "Semantic Tableaux" for propositional logic

We define a procedure *"semantic tableaux"* for propositional classical logic. This procedure, given a formula A in $\mathcal{L}$, accepts A if $\models A$ and returns a model $\mathcal{M}$ such that $\not\models_{\mathcal{M}} A$ otherwise. In fact, we do something more: we give a procedure which works for *finite sets of formulas.*

It is convenient to work with formulas in *negation normal form.* Thus given a formula A which we want to test for validity, we first turn A into an equivalent formula A' in negation normal form (see Exercise 2). Let $\Gamma = C_1, \ldots, C_n$ be a finite set of formulas in negation normal form. A formal expression of the form $\Rightarrow \Gamma$ will be called a *sequent*; the intended interpretation of $\Rightarrow \Gamma$ is

$$\models \ (\bigvee_{j=1}^{n} C_j).$$

Definition. (*valid, falsifiable sequent*) Let S be the sequent $\Rightarrow C_1, \ldots, C_n$. We say that S is *valid* (in symbols, $\models S$) if *for every interpretation* $\mathcal{V}$

- $\mathcal{V}(C_i) = T$ for some $i \leq n$.

A sequent which is not valid is *falsifiable*, i.e., there exists an interpretation $\mathcal{V}$ such that

- $\mathcal{V}(C_i) = F$ for all $i \leq n$.

Completeness Theorem. *The following "semantic tableaux" procedure for propositional logic given a formula A of $\mathcal{L}$ in negation normal form*

- *returns a tree of sequents τ with $\Rightarrow A$ at its root such that every sequent in τ is valid, if $\models A$;*
- *it returns an interpretation $\mathcal{V}$ which falsifies A otherwise.*

Procedure: Construct a tree of sequents τ (with the root at the bottom), called *refutation tree*, as follows.

> *Stage (o)*: let τ_0 be $\Rightarrow A$ (the root of the tree);
>
> *Stage (n+1)*: for each sequent S which occurs at one of the leaves of τ_n, if S has the form of a sequent-axiom, then the procedure terminates on that branch.
>
> Otherwise S has the form of a *sequent-conclusion* of one of the rules in Table 1; in this case write the appropriate *sequent-premise*(s) above it.

Let τ_{n+1} be the tree obtained in this way; if $\tau_{n+1} = \tau_n$ then let $\tau = \tau_n$ and the procedure terminates.

Let S be $\Rightarrow C_1, \ldots, C_n$. Let us define the *size* $s(S)$ of S by letting

$$s(S) \;=\; \left(\Sigma_{i=1}^{n} s(C_i)\right)$$

where $s(p) = 0 = s(\neg p)$ and $s(A \wedge B) = s(A \vee B) = s(A) + s(B) + 1$.

Proposition 1. *The refutation tree τ is finite.*

Proof: At each step of the procedure the size of each new leaf decreases. $\square$

Proposition 2. *For each application of a logical rule the sequent-conclusion of each rule is falsifiable if and only if at least one of the sequent premise is falsifiable.*

Proof: Consider the $\wedge$-rule. If there exists a valuation $\mathcal{V}$ such that $\mathcal{V}(C) = F$ for all $C \in \Gamma$, and $\mathcal{V}(A \wedge B) = F$, then either

axiom
$\Rightarrow \Gamma, p, \Gamma', \neg p$
structural rule
exchange: $\dfrac{\Rightarrow \Gamma, p^{\pm}}{\Rightarrow p^{\pm}, \Gamma}$ where $p^{\pm} = p$ or $\neg p$.
logical rules
$\wedge$: $\dfrac{\Rightarrow \Gamma, A \quad \Rightarrow \Gamma, B}{\Rightarrow \ A \wedge B, \Gamma}$ $\qquad$ $\vee$: $\dfrac{\Rightarrow \Gamma, A, B}{\Rightarrow A \vee B, \Gamma}$

Table 1: Propositional rules

(1) $\mathcal{V}(A) = F$, and in this case $\mathcal{V}$ falsifies the left sequent-premise $\Rightarrow \Gamma, A$, or
(2) $\mathcal{V}(B) = F$, and in this case $\mathcal{V}$ falsifies the right sequent-premise $\Rightarrow \Gamma, B$. Conversely, let $\mathcal{V}$ be a valutation which falsifies one of the sequent premise, hence $\mathcal{V}(C) = F$ per ogni $C \in \Gamma$. If $\mathcal{V}$ falsifies the left sequent-premise, then $\mathcal{V}(A) = F$, hence $\mathcal{V}(A \wedge B) = F$, hence the sequent-conclusion is falsified. If $\mathcal{V}$ falsifies the right sequent-premise, then $\mathcal{V}(B) = F$, hence again $\mathcal{V}(A \to B) = V$, and the sequent conclusion is falsified.

□

An equivalent way to state Proposition 2 is the following: *Propositional rules preserve validity and are semantically invertible.*

Proposition 3. *A sequent of the form Axiom is not falsifiable.*

Proof: No valuation $\mathcal{V}$ can make both p and $\neg p$ false.

□

A *branch* of the refutation tree τ is *closed* if and only if its leaf is a sequent of the form *Axiom*.

Proof of the Theorem. When the procedure terminates, two cases are possible:

- All branches of τ are closed. By induction on the depth of τ, using propositions 2 and 3, we show that no sequent in τ is falsifiable, hence the root $\Rightarrow A$ is not falsifiable, i.e., $\models A$.

- A branch $\beta = S_0, \ldots, S_\ell$ of τ, from the root S_0 to the leaf $S_\ell = \Rightarrow p_1, \ldots, p_m, \neg q_1, \ldots, \neg q_m$, is open. Define $\mathcal{V}$: **Atoms** $\rightarrow \{T, F\}$ as follows:

$$\mathcal{V}(p_i) = F, \text{ for } i \leq m; \quad \mathcal{V}(q_j) = T, \text{ for } j \leq n;$$

$$\mathcal{V}(p) = \text{arbitrary, if } p_i \neq p \neq q_j, \text{ for } i \leq m, j \leq n$$

By induction on the length of β, using proposition 2, we show that $\mathcal{V}$ falsifies every sequent S_i in β hence $\mathcal{V}(A) = F$.

□

1.3 Exercises.

Exercise 1. Show that if $X_1, \ldots X_n$ occur in the same line, then for all i, j, X_i is logically equivalent to X_j. (*Hint: Show that all the sequents* $\Rightarrow \neg X_1', X_2'$, $\Rightarrow \neg X_2', X_3'$, $\ldots$ $\Rightarrow \neg X_n', X_1'$, *are valid, where* X_i' [or $\neg X_j'$] *is* X_i, [or $\neg X_j$] *in negation normal form.*

(a) A; $\neg\neg A$; $(A \wedge A)$; $(A \vee A)$; $(A \wedge (A \vee B))$; $(A \vee (A \wedge B)$.

(b) $\neg A$; $A \rightarrow (B \wedge \neg B)$.

(c) $\neg(A \vee B)$; $(\neg A \wedge \neg B)$. (*De Morgan*)

(d) $\neg(A \wedge B)$; $(\neg A \vee \neg B)$. (*De Morgan*)

(e) $(A \vee B)$; $(B \vee A)$; $(\neg B \rightarrow A)$; $\neg(\neg A \wedge \neg B)$; $((A \rightarrow B) \rightarrow B)$.

(f) $(A \wedge B)$; $(B \wedge A)$; $\neg(A \rightarrow \neg B)$; $\neg(\neg A \vee \neg B)$.

(g) $(A \rightarrow B)$; $(\neg A \vee B)$; $\neg(A \wedge \neg B)$; $(\neg B \rightarrow \neg A)$.

(h) $(A \rightarrow \neg B)$; $(B \rightarrow \neg A)$. (*contrapposition*)

(i) $(A \leftrightarrow B) =_{def} (A \rightarrow B) \wedge (B \rightarrow A)$ $((A \wedge B) \vee (\neg A \wedge \neg B))$.

(j) $\neg(A \leftrightarrow B)$; $((A \vee B) \wedge \neg(A \wedge B))$; $((\neg A) \leftrightarrow B)$.

(k) $(A \wedge (B \vee C))$; $((A \wedge B) \vee (A \wedge C))$. (*distributivity*)

(l) $(A \vee (B \wedge C))$; $((A \vee B) \wedge (A \vee C))$. (*distributivity*)

(m) $((A \vee B) \to C)$; $((A \to C) \wedge (B \to C))$.

(n) $(A \to (B \wedge C))$; $((A \to B) \wedge (A \to C))$.

(o) $(A \to (B \to C))$; $((A \wedge B) \to C)$.

Exercise 2. Verify whether or not the following sequents are falsifiable:

(a) $\Rightarrow \neg(A \vee (B \wedge C)), (C \vee (B \wedge A))$.

(b) $\Rightarrow \neg(A \wedge (B \vee C)), (C \vee (B \wedge A))$.

(c) $\Rightarrow \neg(A \vee (B \wedge C)), (C \wedge (B \vee A))$.

(d) $\Rightarrow \neg(A \to (B \vee C)), ((A \to B) \vee C)$.

(e) $\Rightarrow \neg((A \to B) \vee C)), (A \to (B \vee C))$.

(f) $\Rightarrow ((A \to B) \to A) \to A$.

2 Predicate Calculus

2.1 Syntax

A language $\mathcal{L} = (\mathbf{Pred}, \mathbf{Fun})$ of the calculus of predicates consists of

- a set $\mathbf{Pred} = \{P_1^{n_1}, \ldots, P_i^{n_i}, \ldots\}$ of predicate letters, where $P_i^{n_i}$ has arity $n_i \geq 0$. A 0-ary predicate P^0 is a propositional letter.
- a set $\mathbf{Fun} = \{f_1^{n_1}, \ldots, f_i^{n_i}, \ldots\}$, of function symbols, where $f_i^{n_i}$ has arity n_i. A 0-ary simbol f^0 is a symbol of *constant*.

A first-order language always contains an infinite list of *individual variables* **Var**: $v_0, v_1, \ldots v_i, \ldots$.

Note: A predicate P^n with $n > 0$ is an "unsaturated" expression, with n "holes": e.g., P^2 may represent a predicate "... *is equal to* $---$" or "... *is the father of* $---$".

Terms. The *terms* of the language are defined by the grammar

$$t \ := \ c \mid x \mid f^n(t_1, \ldots, t_n)$$

where c denotes a constant and x an individual variable.

We define *closed terms* and *free occurrences* of a variable in a term inductively:

(i) a symbol for constant c is a *closed* term;

(ii) a symbol x is a *free variable*;

(iii) if $t_1, \ldots \ t_n$ are *closed terms*, then $f^n(t_1, \ldots, t_n)$ is a *closed term*; if x *occurs free* in some term t_i, then x *occors free* in $f^n(t_1, \ldots, t_n)$.

Formulas. Let **Formulas** be the language defined by the following grammar:

$$A \ := \ P^n(t_1, \ldots, t_n) \mid \neg A \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid A_0 \to A_1 \mid \forall x.A \mid \exists x.A$$

Formulas of the form $P^n(t_1, \ldots, t_n)$ are called *atomic*; let **Atoms** be the set of all atomic formulas. Free and bound occurrences of variables in a formula are defined thus:

(i) The *free occurrences* of variables in $P^n(t_1, \ldots, t_n)$ are the free variables of $t_1, \ldots, t_n$;

(ii) The *free occurrences* of variables in $\neg A$, $A_0 \wedge A_1$, $A_0 \vee A_1$, $A_0 \to A_1$ are the free occurrences in A, A_0 and A_1;

(iii) The *free occurrences* of variables in $\forall x.A$ and $\exists x.A$, are the free occurrences of variables in A, less the free occurrences of x in A, which are *bound* by the quantifier.

An occurrence of a variable is *bound* if it is not free. A formula is *closed* if no variable occurs free in it.

2.2 Semantics

An *interpretation* $\mathcal{M}$ of the language $\mathcal{L}$ consists of a non-empty domain M and of a function $(\ )_{\mathcal{M}}$ which assigns

- to each symbol of function f^n an n-ary function $f^n_{\mathcal{M}} : M^n \to M$;

- to each n-ary predicate symbol P^n of $\mathcal{L}$ an n-ary relation $P^n_{\mathcal{M}} \subseteq M^n$.

A constant symbol is interpreted by an element of the domain M.

An *assignment* α is a function $\alpha : \mathbf{Var} \to M$. Given a language $\mathcal{L}$, we use the Greek letter σ for a pair $(\mathcal{M}, \alpha)$ where $\mathcal{M}$ is an interpretation of $\mathcal{L}$ and $\alpha : \mathbf{Var} \to M$ is an assignment of elements in the domain M of $\mathcal{M}$ to the variables.

Given $\sigma = (\mathcal{M}, \alpha)$ where $\mathcal{M}$ is an interpretation and α an assignment, the *value* t^σ of a *term* t under σ will be an element of M, defined by induction on the definition of a term:

(i) if t is a constant c, then $c^\sigma = c_{\mathcal{M}} \in M$;

(ii) if t is a variable x, then $x^\sigma = \alpha(x) \in M$;

(iii) If $t = f(t_1, \ldots, t_n)$, then $t^\sigma = f_{\mathcal{M}}({t_1}^\sigma, \ldots, {t_n}^\sigma) \in M$.

Given an element $d \in M$, we write α^d_i for an assignment which is like α for all variables v_j with $j \neq i$, but assigns d to v_i. Thus we have $\alpha^d_i(v_j) = \alpha(v_j)$ for all $j \neq i$, but $\alpha^d_i(v_i) = d$. Similarly we may write $\alpha^{d,d'}_{i,j}$ for the assignment which may differ from α only by setting $v_i \mapsto d$ and $v_j \mapsto d'$.

Definition. (Tarski) We define what it means to say that a pair $\sigma = (\mathcal{M}, \alpha)$ *satisfies* a formula A (in symbols $\mathcal{M} \models A[\alpha]$) inductively as follows:

1. if $A = P^n(t_1, \ldots, t_n)$ then $\mathcal{M} \models A[\alpha]$ if and only if $\langle t^\sigma_1, \ldots, t^\sigma_n \rangle \in P^n_{\mathcal{M}}$;
2. $\mathcal{M} \models \neg A[\alpha]$ if and only if it is **not** the case that $\mathcal{M} \models A[\alpha]$;
3. $\mathcal{M} \models A_0 \wedge A_1[\alpha]$ if an only if $\mathcal{M} \models A_0[\alpha]$ **and** $\mathcal{M} \models A_1[\alpha]$;
4. $\mathcal{M} \models A_0 \vee A_1[\alpha]$ if and only if $\mathcal{M} \models A_0[\alpha]$ **or** $\mathcal{M} \models A_1[\alpha]$;
5. $\mathcal{M} \models A_0 \to A_1[\alpha]$ if and only if $\mathcal{M} \models A_0[\alpha]$ **implies** $\mathcal{M} \models A_1[\alpha]$;
6. $\mathcal{M} \models \exists v_i.A[\alpha]$ if and only if **for some** $d \in M$ we have $\mathcal{M} \models A[\alpha^d_i]$;
7. $\mathcal{M} \models (\forall v_i.A)[\alpha]$ if and only if **for all** $d \in M$, $\mathcal{M} \models A[\alpha^d_i]$.

Note: In the propositional calculus *interpretations* and *valuations* coincide: they are simply assignments of truth values to propositional letter. On the contrary, in predicate logic a *valuation* is given by an *interpretation* $\mathcal{M}$ of the functions and predicates on a given universe of discourse M, and an *assignment* α of elements of the universe M to the free variables: the truth value of a first-order formula is only determined by a pair $\sigma = (\mathcal{M}, \alpha)$. By clause 1, such a σ determines a total function $(\ )^\sigma$: **Atoms** $\rightarrow \{T, F\}$ where T, F are the truth values *true*, *false* and clauses 2-7 extend such a total function to all formulas $(\ )^\sigma$: **Formulas** $\rightarrow \{T, F\}$. Some textbooks use the notation $A^\sigma = T$ if σ satisfies A, i.e., $\mathcal{M} \models A[\alpha]$, and $A^\sigma = F$ if σ does not satisfy A.

A formula A is *true in the interpretation* $\mathcal{M}$, (in symbols, $\mathcal{M} \models A$), if and only if $\mathcal{M} \models A[\alpha]$ for every assignment α. In this case we also say the $\mathcal{M}$ is a *model* of A. Similarly, if Γ is a set of formulas, an interpretation $\mathcal{M}$ of Γ is *model* of Γ (in symbols $\mathcal{M} \models \Gamma$) if and only if $\mathcal{M}$ is a model of every formula $C \in \Gamma$.

A formula A is *logically valid* if and only if for every interpretation $\mathcal{M}$ and every assignment α it holds that $\mathcal{M} \models A[\alpha]$. If Γ is a set of formulas and A is a formula, we say that A is *valid consequence* of Γ if for every interpretation $\mathcal{M}$ and every assignment α

$$\mathcal{M} \models C[\alpha] \text{ for all } C \in \Gamma \quad \text{implies} \quad \mathcal{M} \models A[\alpha].$$

2.2.1 Substitution

Let r, t be terms. We write $t[x/r]$ for the result of substituting the term r for all occurrences of x in t. More precisely, we have

1. $x[x/r] = r$;
2. $y[x/r] = y$, if $y \neq x$;
3. $c[x/r] = c$, if c is a constant;
4. $f(t_1, \dots, t_n)[x/r] = f(t_1[x/r], \dots, t_n[x/r])$.

It is possible to prove the following

Lemma 0. Let r and t be terms, let v_i be a variable, let $\sigma = (\mathcal{M}, \alpha)$ and $\sigma' = (\mathcal{M}, \alpha_i^{t^\sigma})$. Then

$$(r[v_i/t])^\sigma = r^{\sigma'}$$

In words, the value under σ of the term obtained by substituting t for v_i in r is the same as the value of under $(\mathcal{M}, \alpha')$ of r, where α' is the assignment which agrees with α on all variables $v_j \neq v_i$ but that assigns t^σ to v_i.

If r is a term and A is a formula, we define $A[x/r]$, the result of substituting the term r for all free occurrences of x in A as follows:

1. $P^n(t_1, \dots, t_n)[x/r] = P^n(t_1[x/r], \dots, t_n[x/r])$;
2. $(\neg A)[x/r] = \neg A[x/r]$,
3. $(A_0 \wedge A_1)[x/r] = A_0[x/r] \wedge A_1[x/r]$, and similarly for disjunctions and implications;
4. $(\forall x.A)[x/r] = \forall x.A$;
5. $(\forall y.A)[x/r] = \forall y.(A[x/r])$ if $y \neq x$ and y does not occur in r;
6. $(\forall y.A)[x/r] = \forall z.(A[y/z][x/r])$ if $y \neq x$ occurs in r and z is the first variable in the infinite list **Var** such that z does not occur in A or in r.

It is possible to prove the following facts:

Lemma 01. *For all $\sigma = (\mathcal{M}, \alpha)$, if z does not occur in A then*

$$(\forall x.A)^\sigma = (\forall z.A[x/z])^\sigma.$$

Theorem 0: *For all formulas A, variables v_i, terms t and valuations $\sigma = (\mathcal{M}, \alpha)$, $\sigma' = (\mathcal{M}, \alpha_i^{t^\sigma})$ we have*

$$A[v_i/t]^\sigma = A^{\sigma'}.$$

2.3 "Semantic Tableaux" for first order logic

We extend our *"semantic tableaux"* procedure from propositional to first order predicate logic. This procedure, given a formula A in $\mathcal{L}$, accepts A if $\models A$ but may not terminate if $\not\models_{\mathcal{M}} A$. However, in this case from the infinite tree we may extract a model $\mathcal{M}$ and an assignment α such that $\models_{\mathcal{M}} \neg A[\alpha]$.

Exercise: Show that

$$\models \neg\exists x.A \equiv \forall x.\neg A \quad \text{and} \quad \models \neg\forall x.A \equiv \exists x.\neg A$$

Combining this fact with Exercise 2 in section 1.1, we see that, as in the propositional calculus, there exists a procedure to transform a formulas B in an equivalent one A in negation normal form, i.e., a formula generated according to the grammar

$$A \;:=\; P^n(t_1, \dots, t_n) \mid \neg P^n(t_1, \dots, t_n) \mid \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \forall x.A \mid \exists x.A$$

such that for every interpretation $\mathcal{M}$ of $\mathcal{L}$ and every assignment α, we have $\mathcal{M} \models B[\alpha]$ if and only if $\mathcal{M} \models A[\alpha]$.

We shall work with an infinite sequence of *parameters* a_0, a_1, ... which are to be interpreted in a universe U as the other variables but which cannot occur as bound variables. Thus we will suppose that parameters a_i are our ufficial variables v_{2i} and the usual variables x, which can be bound, are the variables v_{2i+1}. An expression $a : U$ is a declaration of the parameter a; we write $\overline{a} : U$ as abbreviation for $a_1 : U, \ldots, a_n : U$.

Let $\Gamma = C_1, \ldots, C_n$ be a finite set of formulas in negation normal form. A sequent for first order logic is a formal expression of the form $a_0 : U, \ldots, a_i : U \Rightarrow \Gamma$; the intended interpretation of $\Rightarrow \Gamma$ is

$$\models \forall a_0 . \ldots \forall a_i \big(\bigvee_{j=1}^{n} C_j\big).$$

Definition. (*valid, falsifiable sequent*) Let S be the sequent $a_0 : U, \ldots, a_i : U \Rightarrow C_1, \ldots, C_n$. We say that S is *valid* (in symbols, $\models S$) if *for every interpretation* $\mathcal{M}$ and any assignment α

- $\mathcal{M} \models \forall a_0 \ldots \forall a_i . (\bigvee_{j=1}^{n} C_j)[\alpha]$.

A sequent which is not valid is *falsifiable*, i.e., there exists an interpretation $\mathcal{M}$ and an assignment α such that

- $\mathcal{M} \models \neg C_i[\alpha]$ for all $i \leq n$.

Completeness Theorem. *The following "semantic tableaux" procedure for first order predicate logic given a formula A of $\mathcal{L}$ in negation normal form*

- *returns a tree of sequents τ with $a_0 : U \Rightarrow A$ at its root such that every sequent in τ is valid, if $\models A$;*
- *otherwise, if $\not\models A$ either the procedure terminates with some open branch or it does not terminate.*
- *Given a possibly infinite open branch in τ it is possible to build a term-model $\mathcal{M}$ such that the universe of $\mathcal{M}$ consists of terms built up from the parameters $\{a_0, a_1, \ldots, a_i, \ldots\}$ using also the constants and function symbols of $\mathcal{L}$.*

For simplicity we shall consider only the case when the language does not contain constant or function symbols and the formula A does not contain free variables.

Procedure: Construct a tree of sequents τ with the root at the bottom as follows.

Stage (o): let τ_0 be $(a_0 : U) \Rightarrow A$, the root of the tree;

Stage (n+1): for each sequent S which occurs at one of the leaves of τ_n, if S has the form of a sequent-axiom, then the procedure terminates on that branch;

otherwise S has the form of a *sequent-conclusion* of one of the rules in Table 2; then write the appropriate *sequent-premise*(s) above it.

Let τ_{n+1} be the tree obtained in this way; if $\tau_{n+1} = \tau_n$ then let $\tau = \tau_n$ and the procedure terminates. It is clear that Proposition 1 no longer holds in

axiom

$$\overline{a} : U \Rightarrow \Gamma, P^n(t_1, \ldots, t_n), \Gamma', \neg P^n(t_1, \ldots, t_n)$$

structural rule

exchange:

$$\frac{\overline{a} : U \Rightarrow \Gamma, p^{\pm}}{\overline{a} : U \Rightarrow p^{\pm}, \Gamma} \qquad \text{where } p^{\pm} = p \text{ or } \neg p.$$

logical rules

$\wedge$ *rule:*

$$\frac{\overline{a} : U \Rightarrow \Gamma, A \quad \overline{a} : U \Rightarrow \Gamma, B}{\overline{a} : U \Rightarrow \ A \wedge B, \Gamma}$$

$\vee$ *rule:*

$$\frac{\overline{a} : U \Rightarrow \Gamma, A, B}{\overline{a} : U \Rightarrow A \vee B, \Gamma}$$

$\forall$ *rule:*

$$\frac{a_0 : U, \ldots, a_i : U, a_{i+1} : U \Rightarrow \Gamma, A[a_{i+1}/x]}{a_0 : U, \ldots, a_i : U \Rightarrow \ \forall x.A, \Gamma}$$

$\exists$ *rule:*

$$\frac{a_0 : U, \ldots, a_i : U \Rightarrow \Gamma, A[a_0/x], \ldots, A[a_i/x], \exists x.A}{a_0 : U, \ldots, a_i : U \Rightarrow \exists x.A, \Gamma}$$

Table 2: Propositional rules

the predicate calculus: because of the $\exists$-rule it is quite possible that the refutation tree τ is infinite. Nevertheless, Proposition 2 remains true:

Proposition 2. *For each application of a logical rule, the sequent-conclusion of each rule is falsifiable if and only if at least one of the sequent premise is falsifiable.*

Proof. ($\exists$-rule): If the sequent-conclusion is falsified by a valuation $\sigma = (\mathcal{M}, \alpha)$, then for every element $d \in M$ the assignment α_i^d yields $\mathcal{M} \not\models \Gamma[\alpha_i^d]$, i.e., it is impossible to satisfy A in $\mathcal{M}$. Hence it is also impossible to satisfy any $A[a_j/x]$ in $\mathcal{M}$. The converse is obvious.

($\forall$-rule): Let v_j be the variable x possibly occurring in A; by Lemma 01 in section 2.2.1 we may assume that v_j dos not occur in Γ.
If the sequent-conclusion is falsified by a valuation $\sigma = (\mathcal{M}, \alpha)$, then for some element $d \in M$ the valuation $\sigma' = (\mathcal{M}, \alpha_j^d)$ falsifies $\Rightarrow \Gamma, A$. Since a_{i+1} ($= v_{2i+2}$) does not occur in the sequent-conclusion, $\sigma'' = (\mathcal{M}, \alpha_{j,2i+2}^{d,d}])$ also falsifies $\Rightarrow \Gamma, A$. But $A^{\sigma''} = A[\_i + 1/x]$ (i.e., the valuation σ'' gives the same truth value to A and to $A[a_{i+1}/x]$) hence the sequent-premise $\Rightarrow \Gamma, A[a_{i+1}/x]$ is also falsified by σ''.
Conversely, if $\sigma'' = (\mathcal{M}, \alpha_{2i+2}^d)$ falsifies $\Rightarrow \Gamma, A[a_{i+1}/x]$ where a_{i+1} (i.e., v_{2i+2}) does not occur in Γ, then also $\sigma' = (\mathcal{M}, \alpha_j^d)$ falsifies $\Rightarrow \Gamma, A$ (where $x = v_j$) (see Theorem 0 in section 2.2.1), hence also $\sigma = (\mathcal{M}, \alpha)$ falsifies $\Rightarrow \Gamma, \forall x.A$. The proof is finished.

Since for all $\sigma = (\mathcal{M}, \alpha)$, $P^n(t_1, \ldots, t_n)^\sigma \neq (\neg P^n(t_1, \ldots, t_n))^\sigma$, we still have

Proposition 3. *A sequent of the form Axiom is not falsifiable.*

□

A *branch* of the refutation tree τ is *closed* if and only if its leaf is a sequent of the form *Axiom*.

Let β be a possibly infinite *open* branch of the refutation tree τ and let ℓ $= a_0, a_1, \ldots, a_n \ldots$ be the (possibly infinite) list of all parameters occurring in β. From the form of the $\exists$-rule the following Lemma is clear:

Lemma 1. *If β is an open branch in τ and $\exists x.A$ occurs in some sequent of β, then for all $a \in \ell$ also $A[a/x]$ also occurs in some sequent of β.*

We also need the following principle from set-theory, in the case of binary branching trees.

König's Lemma: *Every finitely branching infinite tree contains an infinite path.*

Proof. Suppose we have a path β from the root to a node v such that the subtree τ with root in v is infinite and let $v_1, \ldots, v_n$ be the nodes related to v by an edge (the "children" of v). Then at least one of v_i must be the root of an infinite subtree τ_i; indeed otherwise τ itself would have finite depth $max_{i \leq n}(\text{ depth}(\tau_i) + 1)$, a contradiction. Therefore the path β can be extended to v_i.

□

Proof of the Theorem. Either the procedure terminates with all branches in τ closed, or it does not.

- If all branches of τ are closed, then by induction on the depth of τ, using propositions 2 and 3, we show that no sequent in τ is falsifiable, hence the root $\Rightarrow A$ is not falsifiable, i.e., $\models A$.

- Otherwise, either the tree is finite and contains an open branch β, or it is infinite, and then by König's Lemma it contain an infinite open branch.

- Thus let β be a possibly infinite open branch. Define an interpretation (*term model*) $\mathcal{M}$ of A as follows.

  For the domain M of the interpretaton we take precisely the list ℓ.

  (¶) For the interpretation of the predicate letter P^n we set

  $\langle a_{i_1}, \ldots, a_{i_n} \rangle \in P_{\mathcal{M}}$ if and only if $\neg P^n(a_{i_1}, \ldots, a_{i_n})$ occurs in some sequent of β.

Lemma 2. *Given a sequent S without free variables, let $\mathcal{M}$ be the "term model" built from an open branch of the tree τ for S and let α be the assignment $\alpha : a_i \mapsto a_i$ for all parameters a_i. Then for all formulas B occurring in some sequent of β, the valuation $\sigma = (\mathcal{M}, \alpha)$ falsifies B.*

Proof. By induction on the logical complexity of a formula B.
If B is atomic or the negation of an atom, the result holds by (¶).
If $B = B_1 \wedge B_2$ [or $B_1 \vee B_2$], then a subformula B_i [both subformulas B_1 and B_2] occur in a sequent of β, and the results follows by the inductive hypothesis by the same argument as in the propositional case.
If $B = \forall x.C$, then $C[x/a]$ also occurs in β for some parameter $a \in \ell = M$, hence $C[x/a]^\sigma = F$ by the inductive hypothesis, and so $\forall x.C^\sigma = F$.
If $B = \exists x.C$ occurs in β, then for all parameters $a \in \ell = M$ the formula $C[x/a]$ also occurs in β by Lemma 1, hence $C[x/a]^\sigma = F$ by the inductive hypothesis, and so $\exists x.C^\sigma = F$.
The proof of Lemma 2 is finished, hence also the proof of the Theorem.

2.4 Other formal systems for predicate logic

There are several equivalent *axiomatic systems*, i.e., systems of axioms and rules of inference, for first order predicate logic; let us write $\Gamma \vdash A$ if A is derivable from a set of assumptions Γ in any such axiomatic system.

Let $\Gamma \vdash A$ where $\Gamma = C_1, \ldots, C - n$ is a finite set of formmulas. Let A' be the formula in negation normal form equivalent to A and $\neg\Gamma'$ the set of formulas D_i in negation normal form such that $D_i \equiv \neg C_i$ for $i \neq n$. It is possible to show that $\Gamma \vdash A$ *if and only if* $a_0 : U \Rightarrow \neg\Gamma', A'$ *has a* closed *refutation tree* τ. In other words, closed refutation trees may be regarded as proof-systems: indeed they are a variant of Gentzen's sequent calculi for classical logic.

A key remark about axiomatic systems is the following: since in any derivation of A from Γ only a finite number of assumptions is used, $\Gamma \vdash A$ implies $\Gamma_0 \vdash A$ for some finite subset $\Gamma_0 = C_1, \ldots, C_n$ of Γ. An important corollary of this remark is the following principle.

A possibly infinite set Γ of formulas is *inconsistent* if $\Gamma \vdash A \wedge \neg A$; it is *consistent* otherwise.

Compactness Principle: *A set* Γ *of formulas is consistent if and only if every finite subset* Γ_0 *of* Γ *is consistent.*

It is possible to extend our procedure to infinitely long sequents Γ. Let us consider an application of the compactness principle in this context. Let Γ a possibly infinite sequence of formulas, let Γ_n be initial segment $C_1, \ldots, C_n$ of Γ and let $\neg\Gamma_n$ be the sequence $D_1, \ldots, D_n$ of formulas in negation normal form such that $D_i \equiv \neg C_i$. We apply the procedure of the completeness theorem successively to the sequences $\neg\Gamma_1, \neg\Gamma_2, \ldots, \neg\Gamma_i$. If Γ is consistent, then no one of the sequents $a_0 : U \Rightarrow \neg\Gamma_i$ has a closed refutation tree; otherwise, we would have $\Gamma_i \vdash A \wedge \neg A$ and hence $\Gamma \vdash A \wedge \neg A$, i.e., Γ would be inconsistent. Therefore there is a valuation $\sigma_i = (\mathcal{M}_i, \alpha_i)$ which falsifies $\neg\Gamma_i$, i.e., which satisfies Γ_i. Moreover, a valuation σ_i which satisfies Γ_i also satisfies Γ_j with $j < i$. It follows that there is a valuation $\sigma = (\mathcal{M}, \alpha)$ which satisfies Γ as a whole; i.e., the compactness principle can be stated as follows:

Compactness Principle: *A set* Γ *of formulas is satisfiable if and only if every finite subset* Γ_0 *of* Γ *is satisfiable.*

3 Equality

It is possible to extend any axiomatic system for the predicate calculus to an axiom system for the logic of equality, by adding all axioms of the form

$$\vdash t = t \qquad A[x/t_i], t_i = t_j \vdash A[x/t_j]$$

for all t, t_i, t_j and A.

It is possible to extend the above semantic procedure to the logic of equality. We shall not do this in detail, but let us consider what it is involved in this extension. First, we can introduce axioms and rules of the form

$$= \textit{axiom :} \qquad\qquad = \textit{rule:}$$

$$\overline{a} : U \Rightarrow t = t \qquad \frac{\overline{a} : U \Rightarrow \Gamma, A[t_i/x], A[t_j/x], \Gamma', \neg(t_i = t_j)}{\overline{a} : U \Rightarrow \ \neg(t_i = t_j), \Gamma, A[t_i/x], \Gamma'}$$

We run our procedure with these additional rules. If we obtain an open branch β, we must construct a countermodel $\mathcal{M}$; the main difference is that now the objects of M shall be *equivalence classes of parameters in* ℓ, rathen than parameters themselves: we let $a_i \sim a_j$ and put a_i and a_j in the same equivalence class if and only if $\neg a_i = a_j$ ever occurs in a sequent of β; we extend the definition of the equivalence classes to terms by letting $f^n(t_1, \dots, t_n) \sim f^n(u_1, \dots, u_n)$ if and only if $t_1 \sim u_1, \dots, t_n \sim u_n$.

Notice that as a consequence, it may be possible to obtain a *finite* countermodel out of an infinite branch with infinitely many distinct parameters.