THE STRONG COMPLETENESS OF THE TABLEAU METHOD[1]

GUSTAVO FERNÁNDEZ DÍEZ
Facultad de Filosofía, Universidad de Murcia
Apdo. 4021, E-30071 Murcia, España
gfdezdp@um.es

En este trabajo se describe una prueba directa (sin usar el teorema de compacidad) del hecho de que el método de árboles o tablas semánticas de primer orden es completo en el sentido fuerte, es decir: que si una fórmula φ es una consecuencia de primer orden de un conjunto arbitrario de fórmulas Φ (no necesariamente contable), entonces existe un subconjunto finito Φ_0 de Φ tal que el conjunto $\Phi_0 \cup \{ \neg\varphi \}$ genera una tabla cerrada. Además, se hará una observación incidental sobre un aspecto de la monotonía del método de tablas que ha dado lugar a algún error en la literatura publicada.

Palabras clave**:** Completud fuerte, prueba de Henkin, tablas semánticas, método de árboles, monotonía.

Appropriate indications are given on how to adapt Henkin's proof of the strong completeness theorem for the predicate calculus as to obtain a direct proof (without using compactness) of the fact that the most user-friendly formal method of deduction, the Beth-Smullyan method of first-order tableaux, is similarly complete, in the sense that if a formula φ is a first-order consequence of an arbitrary set Φ of formulas, there is a finite subset Φ_0 of Φ such that the set $\Phi_0 \cup \{ \neg\varphi \}$ generates a closed tableau. Also, an incidental remark will be made concerning the monotony of the method of tableaux, on a problem which has led to a technical mistake in some of the published literature.

Key-words**:** Strong completeness, Henkin's proof, semantic tableaux, tree method, monotony.

1. Introduction

Amongst all known formal systems of deduction for first-order logic, the Beth-Smullyan method of semantic tableaux is by far the most user-friendly. It is virtually algorithmic (indeed, totally algorithmic in the most rudimentary versions), extremely elegant and easy to use, and very natural too, mimicking the *principle of indirect proof*, familiar from informal

[1] The research leading to this paper has been conducted at the London School of Economics, under the direction of Professors Colin Howson and Moshé Machover, and supported by the British Council, the British Academy and *Caja de Ahorros del Mediterráneo.*

deduction. In addition to this, most of its corresponding metatheory is as simple and handy as that of any linear calculus; in particular, the method has been proven to be complete in several *weak* senses, including

(1) complete with respect to *finite sets of formulas* (e.g. [1], [2]), and

(2) complete with respect to *denumerable sets of sentences* (e.g. [3], with gaps: cf. Note 2).

Moreover, when the method is presented in a monotone format (see Def. 2.1 below), and due to the compactness of first-order logic, the weak completeness in sense (1) entails the full or *strong* completeness too, that is, completeness for *arbitrary sets of formulas* (cf. van Benthem [2]).

However, somewhat surprisingly, it seems that a direct proof of this strong completeness theorem for tableaux has never been actually published. The aim of this paper is to fill the gap by showing how we can obtain such a proof simply adapting Henkin's well-known proof [4] of the strong completeness of the predicate calculus.

Also, an incidental remark will be made concerning the monotony of the method of tableaux, on a problem which has led to a technical mistake in some of the published literature.

2. Discussion

In the sequel (except in Note 2) first-order languages will be assumed to have a single quantifier, $\forall$, and equality. An appropriate method T of constructing tableaux for *finite* sets of formulas is defined in the usual way. I follow Bell and Machover [1] on all basic settings concerning the language and T, as well as on all general terminology and notational conventions (except that I avoid the use of bold type, replacing it by plain type or italics); in particular, T will have three equality rules, corresponding to the three usual axioms of equality, rather than a substitution rule. Of course, with minor modifications what follows can also be applied to those methods having a substitution rule, as well as to most current settings where other details vary.

Now let L be a first-order language. Sometimes I simply say 'variable', 'formula', etc., meaning 'variable of L', 'formula of L', etc., respectively.

2.1. DEFINITION. Let Φ be an arbitrary set of formulas and φ any formula. We say that φ is *deducible from Φ by the first-order tableau method T* (briefly, $\Phi \vdash_T \varphi$) when there is a finite subset Φ_0 of Φ such that the set $\Phi_0 \cup \{ \neg\varphi \}$ can be confuted –i.e., generates a closed tableau–.

Notice that, by contrast with what happens with linear deductions, a closed tableau for $\Phi_0 \cup \{ \neg\varphi \}$ may become invalid if we add more formulas from $\Phi \supseteq \Phi_0$ to its initial node. Indeed, the restricted rule for the negated universal quantifications may have been applied in the tableau, using critical variables which appear free in those new formulas.

In most of these cases it is easy to find a new confutation for $\Phi_0 \cup \{ \neg\varphi \}$ to whose initial node we may add any formula from Φ without invalidating it. Indeed, we can simply imitate the original tableau, but wherever the restricted rule has been applied with a critical variable which is free in Φ, we use instead any other variable that is neither free in the upper nodes of the branch in question nor in Φ; and from that point onwards we continue the tableau likewise, but using this new variable in place of the old one. It is quite clear that the result of this transformations is another confutation for $\Phi_0 \cup \{ \neg\varphi \}$ according to the rules of *T*.

However, when the number of variables of *L* which are not free in Φ is smaller than the number of times that we need to apply the $\neg\forall$-rule in some branch, such a confutation will not exist. For example, let *L* contain a unary predicate symbol *P* and put

$\Phi = \{ \forall x \neg\neg Px, Py\text{: } y \text{ is an } L\text{-variable} \}$, and

$\varphi = \forall x Px$.

It is immediate that $\Phi \vdash_T \varphi$ simply taking $\Phi_0 = \{ \forall x \neg\neg Px \} \subseteq \Phi$. However, any closed tableau for $\{ \forall x \neg\neg Px, \neg\forall x Px \}$ must contain an application of the rule for the negated universal using some critical variable *y*, so that the addition of the corresponding formula $Py \in \Phi$ to its initial node will invalidate it[2].

[2] This point has led to some minor errors in the literature. Notably, the tableau method described in Boolos and Jeffrey [3] is not complete in the way they claim it to be (see p. 131) because the application of the $\exists$-rule is restricted to the use of a critical constant –rather than a variable– which does not occur at all in the set of premises (p. 123). If we then put, similarly as before,

It is because of this problem that, in order to guarantee the strong completeness of the tableau method, we must begin by adopting in one way or other the idea of Def. 2.1. Or else, we could allow for an extension of the language with a new variable every time that is needed to instantiate a $\neg\forall$-formula –in a way similar to the one suggested in Note 2– : but our choice will make things significantly simpler.

2.2. DEFINITION. A set Φ of formulas is *tableau inconsistent in L* when there is a finite subset Φ_0 of Φ which generates a closed tableau all whose formulas belong to *L*. Otherwise it is *tableau consistent in L*.

2.3. DEFINITION. A set Φ of formulas is *tableau Henkin in L* when it satisfies the following two conditions:

(1) it is maximal tableau consistent in *L*, that is: it is consistent in *L*, but no *L*-formula can be added to it without destroying its consistency; and

(2) for any formula $\neg\forall x\varphi$ which belongs to Φ –for some variable *x* and formula φ– there is an *L*-term *t* such that the formula $\neg\varphi(x/t)$ also belongs to Φ.

In the sequel, sometimes we shall simply say, e.g. 'consistent in *L*', meaning 'tableau consistent in *L*', etc. There will be no risk of confusion.

3. Preliminary lemmas

In some of the following lemmas I shall operate with some variables which do not appear *at all* in certain given sets of formulas; in this way we will get convenient simplifications on the details of the proofs without

$\Psi = \{ \forall xPx, \exists x\neg Px, Pc$: *c* is a constant of the language $\}$,

it turns out that the set Ψ is clearly unsatisfiable, but it does not generate a closed tree –tableau–, because the sentence $\exists x\neg Px$ cannot be instantiated.

In particular, Ψ does not generate what they call a 'canonical derivation' (pp. 123 and 131-132), since in such a tree all the premises in question must be present. Hence the result called 'Lemma I' on p. 132 does not follow. And this is one of the bases on which they build up their proof of the completeness theorem (with respect to countable sets of sentences). For that matter, in the cases of languages containing only finitely many constants (which are allowed, cf. p. 97), it follows that their system is not even weakly complete.

Despite all that, the problem seems to be easily solvable, without having to reconstruct the whole proof: for example, by allowing for an extension of the language with a new constant every time that one is needed to instantiate an existential sentence.

loosing any relevant information. However, it is easy to see that similar proofs can also be carried out taking any variables which simply are *not free* in the sets of formulas in question. The only essential difference is that in that case we must define the substitution for constants in such a way that when the constant in question lies within the scope of a quantifier with a variable which appears in the substituting term, that variable of quantification is previously changed appropriately (this treatment differs from that of Bell and Machover [1]).

3.1. LEMMA. Let φ and ψ be two formulas such that ψ is obtainable from φ by a single application of one of the tableau rules. Then, if y is a variable which does not appear at all either in φ or in ψ, and c is any constant, $\psi(c/y)$ can also be obtained from $\varphi(c/y)$ by a single application of the same tableau rule.

PROOF. First suppose that φ does not contain c, so that $\varphi(c/y) = \varphi$ itself. Then, if ψ does not contain c either, $\psi(c/y) = \psi$ and the lemma is trivial. If ψ does contain c, then it must have been introduced by the $\forall$-rule, the only rule which allows the introduction of new constants (apart from the equality rules, but they do not operate transforming a previous formula). Then $\varphi = \forall x\alpha$ for some variable x and formula α, and $\psi = \alpha(x/t)$ for some term t containing occurrences of c. In this case it is enough to apply the $\forall$-rule to φ using the term $t(c/y)$ rather than t itself. Since c does not appear in α –because it does not appear in φ–, we have:

$$\alpha(x/t(c/y)) = \alpha(x/t)(c/y) = \psi(c/y).$$

Now suppose that φ contains some occurrences of c. We reason according to the rule by means of which ψ has been obtained from φ. If it is a propositional rule the lemma becomes obvious; we deal with the quantifier rules.

Case 1. $\forall$-rule. Then $\varphi = \forall x\alpha$ for some variable x and formula α containing c, and $\psi = \alpha(x/t)$ for some term t. Since $y \neq x$ –because y does not appear *at all* in φ–, $\varphi(c/y) = \forall x\alpha(c/y)$. Then we apply the $\forall$-rule to $\varphi(c/y)$ but using the term $t(c/y)$ instead of t. And

$$\alpha(c/y)(x/t(c/y)) = \alpha(x/t)(c/y) = \psi(c/y).$$

Case 2. $\neg\forall$-rule. Then $\varphi = \neg\forall x\alpha$ for some variable x and formula α containing occurrences of c, and $\psi = \neg\alpha\,(x/z)$ for some appropriate variable z. Since $y \neq z$ we can equally well apply the $\neg\forall$-rule to $\varphi\,(c/y) = \neg\forall x\alpha\,(c/y)$ to obtain $\neg\alpha\,(c/y)(x/z)$. But $y \neq x$ too –as before–, so:

$$\neg\alpha\,(c/y)(x/z) = \neg\alpha\,(x/z)(c/y) = \psi\,(c/y).\ \blacksquare$$

3.2. LEMMA. Let B be a finite branch of a tableau containing a sequence of formulas $\varphi_0, ..., \varphi_n$ such that φ_0 belongs to the initial node, and each of the others has been obtained from the immediately preceding one by a single application of one of the tableau rules. Then, if y is a variable which does not appear at all in any of them and c is any constant, we can replace in B each φ_i (for $0 \leq i \leq n$) by $\varphi_i\,(c/y)$ and the resulting branch also agrees with the same tableau rules as the original one.

PROOF. Immediate by –strong– induction on n, applying Lemma 3.1.$\blacksquare$

3.3. LEMMA. Let Φ be any set of formulas and T a confutation for a finite subset Φ_0 of Φ. Then there is a confutation S for Φ_0 such that all constants which occur in S also occur in Φ.

PROOF. Let $\{ c_0, ..., c_n \}$ be the set of constants which occur in T but not in Φ. If it is empty there is nothing to prove. On the other hand, it has to be finite, since T is a closed tableau. We then proceed by strong induction on n.

Indeed, assume that S_m is a confutation for Φ_0 where neither of $c_0, ..., c_m$ appear (for $0 \leq m \leq n$), and let y be a variable which do not appear at all in Φ_0 or S_m –for definiteness we can take the first one in the alphabetic order of the variables of L–. Then we let S_{m+1} be the result of replacing every formula ψ in S_m by $\psi\,(c_{m+1}/y)$; since c_{m+1} does not appear in Φ, for any formula $\varphi \in \Phi_0 \subseteq \Phi$, we have $\varphi = \varphi\,(c_{m+1}/y)$. Then by Lemma 3.2 –recall S_m finite– S_{m+1} is also a tableau for Φ_0 according to the rules, and obviously closed, since S_m is; in particular, it is obvious that every application of the splitting rule for $\rightarrow$ will remain valid, since the replacement is effected on the two resulting formulas. On the other hand, if a formula ψ in S_m has been obtained directly by means of one of the equality rules, it is also immediate that $\psi\,(c_{m+1}/y)$ can also be introduced by the same rule. $\blacksquare$

3.4. LEMMA. If a set Φ of formulas is consistent in a language, it remains so in any other language obtained from it by adding an arbitrary set of new constants.

PROOF. All the new constants of any confutation for Φ can be easily removed by Lemma 3.3. ∎

3.5. LEMMA. Let Φ be a finite set of formulas, φ a formula and c any constant which does not appear in either of them. Then, if for some variable x the set $\Phi \cup \{ \varphi(x/c) \}$ is inconsistent, there are infinitely many variables which do not appear *at all* in Φ or φ, such that, if y is any of them, the set $\Phi \cup \{ \varphi(x/y) \}$ is also inconsistent.

PROOF. If x is not free in φ then we have $\varphi(x/c) = \varphi = \varphi(x/y)$ for any variable y, and since Φ is finite there must be infinitely many variables which do not occur at all in Φ or φ; in this case there is nothing to prove.

Now we assume that x is free in φ. Therefore $\varphi(x/c)$ contains some occurrences of c. Then there is an immediate way of transforming any confutation T for $\Phi \cup \{ \varphi(x/c) \}$ into a confutation S for $\Phi \cup \{ \varphi(x/y) \}$, where y is any variable which does not appear at all in T (since T is closed, therefore finite, there must be infinitely many of those variables). Indeed, all we have to do is to replace every formula ψ of T by $\psi(c/y)$. In particular, the initial node remains almost the same (since c does not appear in Φ) only with $\varphi(x/y)$ substituted for $\varphi(x/c)$. (Notice $\varphi(x/c)(c/y) = \varphi(x/y)$.)

To see that S is also a tableau according to the rules, first suppose that a formula ψ in T has been obtained from $\varphi(x/c)$ after a series of applications of the tableau rules yielding the formulas $\psi_0, \ldots, \psi_n = \psi$. Then, by Lemma 3.2 —since y does not appear at all in T— the new sequence $\varphi(x/y)$, $\psi_0(c/y)$, ..., $\psi_n(c/y)$ also agrees with the same tableau rules as the old one. Now suppose that ψ has been obtained after a series of applications of the rules from a formula $\alpha \in \Phi$. But since c does not occur in α —because it does not occur in the whole of Φ —, $\alpha = \alpha(c/y)$, so again by Lemma 3.2 the corresponding sequence in S also agrees with the same rules as the old one. Finally, if ψ has been introduced directly by means of one of the equality rules, again it is immediate that $\psi(c/y)$ can also be introduced by the same rule. As before, it is clear that every application of the splitting rule for → will remain valid in S.

The fact that S is closed is obvious given that T is. ∎

3.6. LEMMA. Let Φ be a consistent set of formulas such that $\neg\forall x\varphi \in \Phi$ for some variable x and formula φ, and let the constant c do not occur in Φ. Then the set $\Phi \cup \{ \neg\varphi(x/c) \}$ is also consistent.

PROOF. Suppose $\Phi \cup \{ \neg\varphi(x/c) \}$ is inconsistent. Then there is a finite $\Phi_0 \subseteq \Phi$ such that $\Phi_0 \cup \{ \neg\varphi(x/c) \}$ is inconsistent too. Let T then be a confutation for $\Phi_0 \cup \{ \neg\varphi(x/c) \}$. Since $\neg\forall x\varphi \in \Phi$ and c does not occur in Φ, c does not occur in φ either. Hence by Lemma 3.5 $\Phi \cup \{ \neg\varphi(x/y) \}$ is also inconsistent for some variable y which does not appear at all in Φ. Moreover, since there are infinitely many of those variables we can choose y so that it does not appear free in T –T is finite, since it is a closed tableau–.

But then $\neg\varphi(x/y)$ can be obtained immediately from $\neg\forall x\varphi$ by a single application of the $\neg\forall$-rule, so Φ itself is inconsistent. ∎

4. The proof of strong completeness

I will not reproduce the whole proof in detail, since it is similar to the one for Henkin sets of the predicate calculus, with minor modifications (Bell and Machover [1] give a detailed version of the latter, which is the one I shall closely follow here.) Thus, I shall only give an outline, indicating where some of the previous lemmas (in particular 3.4 and 3.6) are needed.

We start by adding to L a set of new individual constants of cardinality equal to that of L itself, say λ. Hence, if L^+ is the resulting new language, we have $|L^+| = |L| = \lambda$.

Then, let Φ be any arbitrary set of formulas of L. The proof proceeds from a fixed well-ordering on the formulas of L^+: $\{ \varphi_\xi : \xi < \lambda \}$; and a definition by transfinite recursion of a set Φ_ξ of formulas of L^+ (for every $\xi \leq \lambda$), satisfying the following three conditions:

(1) $\Phi_\zeta \subseteq \Phi_\xi$ for all $\zeta < \xi$,

(2) Φ_ξ is consistent in L^+, and

(3) Φ_ξ contains at most $|\xi|$ of the new constants.

The definition goes as follows. We put, of course, $\Phi_0 = \Phi$; hence we have that Φ_0 is consistent also in L by Lemma 3.4, and the two conditions hold vacuously.

Next, if ξ is a limit ordinal we put $\Phi_\xi = \cup \{ \Phi_\zeta : \zeta < \xi \}$. Hence condition (1) holds by definition and (2) and (3) follow directly from the assumption that all Φ_ζ for $\zeta < \xi$ satisfy them.

Then, if ξ is a successor ordinal, say $\xi = \zeta + 1$, we distinguish three cases:

(a) If φ_ζ is a formula other than $\neg\forall x\varphi$ (for a variable x and formula φ), and $\Phi_\zeta \cup \{ \varphi_\zeta \}$ is consistent, then we put $\Phi_\xi = \Phi_\zeta \cup \{ \varphi_\zeta \}$. It is immediate to see that the three conditions hold.

(b) If φ_ζ is of the form $\neg\forall x\varphi$ (for a variable x and formula φ), and $\Phi_\zeta \cup \{ \varphi_\zeta \}$ is consistent, then by (3) there is at least a new constant c which does neither occur in Φ_ζ nor in φ_ζ. In that case we put

$$\Phi_\xi = \Phi_\zeta \cup \{ \varphi_\zeta , \neg\varphi (x/c) \}.$$

Hence conditions (1) and (3) clearly hold, once more, and so does (2) by our previous lemma 3.5. And finally,

(c) If $\Phi_\zeta \cup \{ \varphi_\zeta \}$ is inconsistent we put $\Phi_\xi = \Phi_\zeta$, and the three conditions are trivially satisfied.

It is not difficult to show, then that the resulting set Φ_λ is a tableau Henkin set in L^+. Indeed, if an L^+-formula ψ (which must be some φ_ζ for some $\zeta < \lambda$ in the well-ordering of the formulas of L^+) has been left out of Φ_λ, this can only mean that the corresponding set $\Phi_\xi \cup \{ \varphi_\xi \}$ was inconsistent, hence so would be $\Phi_\lambda \cup \{ \varphi_\xi \}$ itself. On the other hand, the consistency of Φ_ξ alone is guaranteed by condition (2).

And finally, again by the definition of Φ_λ, it is evident that it must contain a formula $\neg\varphi (x/c)$ for every $\neg\forall x\varphi$ belonging to it, that is: Φ_λ is saturated with respect to instantiations of $\neg\forall$.

It is easy to see that Φ_λ, being a tableau Henkin set, is clearly a Hintikka set as well, and the proof that it is satisfiable proceeds just as usual —in particular, Φ_λ is shown to be satisfied by a valuation whose domain has a cardinality smaller or equal to λ—. As before, for full details we refer to [1].

Our original set is evidently included in Φ_λ , so it itself is satisfiable as well. And we have established the strong satisfaction theorem, from which the strong completeness theorem follows easily –the two theorems being, indeed, essentially equivalent–.

REFERENCES

[1] J. L. BELL and M. MACHOVER, *A course in mathematical logic*, North-Holland, Amsterdam, 1977.

[2] J. F. A. K. VAN BENTHEM, Semantic Tableaus, *Nieuw Archief voor Wiskunde (3)* 22, 1974, pp. 44-59.

[3] G. BOOLOS and R. JEFFREY, *Computability and logic* (3rd edition), Cambridge University Press, Cambridge, 1989.

[4] L. HENKIN, The completeness of the first-order functional calculus, *Journal of Symbolic Logic* 14, 1949, pp. 159-166.