

Chapter 3

CLASSICAL LOGIC—BASIC TABLEAUS

Several varieties of proof procedures have been developed for first-order classical logic. Among them the *semantic tableau* procedure has a considerable attraction, [Smu68, Fit96]. It is intuitive, close to the intended semantics, and is automatable. For higher-order classical logic, semantic tableaux are not as often seen—most treatments in the literature are axiomatic. Among the notable exceptions are [Tol75, Smi93, Koh95, Gil01]. In fact, semantic tableaux retain much of their first-order ability to charm, and they are what I present here. Automatability becomes more problematic, however, for reasons that will become clear as we proceed. Consequently the presentation should be thought of as meant for human use, and intelligence in the construction of proofs is expected.

This chapter examines what I call a *basic* tableau system; rules are lifted from those of first-order classical logic, and two straightforward rules for predicate abstracts are added. It is a higher-order version of the second-order system given in [Tol75]. I will show it corresponds to the *generalized* Henkin models from Section 5 of Chapter 2. In Chapters 5 and 6 I make additions to the system to expand its class of theorems and narrow its semantics to Henkin models.

1. A Different Language

In creating tableau proofs I use a modified version of the language defined in Chapter 2. That is, I give tableau proofs of sentences from the original language $L(C)$, but the proofs themselves can involve formulas from a broader language that is called $L^+(C)$. Before presenting the tableau rules, I describe the way in which the language is extended for proof purposes.

Existential quantifiers are treated at higher orders exactly as they are in the first-order case. If we know an existentially quantified formula is true, a new symbol is introduced into the language for which we say, in effect, let that be something whose value makes the formula true. As usual, newness is critical. For this purpose it is convenient to enhance the collection of free variables by adding a second kind, called parameters.

DEFINITION 3.1 (PARAMETERS) *In $L(C)$, for each type t there is an infinite collection of free variables of that type. The language $L^+(C)$ differs from $L(C)$ in that, for each t there is also a second infinite list of free variables of type t , called parameters, a list disjoint from that of the free variables of $L(C)$ itself. Parameters may appear in formulas in the same way as the original list of free variables but they are never quantified or λ bound. p, q, P, Q, \dots are used to represent parameters.*

Parameters appear in tableau proofs. They do not appear in the sentences being proved. Since they come from an alphabet distinct from the original free variables, an alphabet that is never quantified or λ bound, we never need to worry about whether the introduction of a parameter will lead to its inadvertent capture by a quantifier or a λ —introducing them will always involve a free substitution. Thus rules that involve them can be relatively simple.

Special Terminology Technically, parameters are a special kind of free variable. But to keep terminology simple, I will continue to use the phrase *free variable* for the free variables of $L(C)$ only, and when I want to include parameters in the discussion I will explicitly say so.

The notion of truth in generalized Henkin models must also be adjusted to take formulas of $L^+(C)$ into account. As I have just noted, parameters are special free variables, so when dealing semantically with $L^+(C)$, valuations must be defined for parameters as well as for the free variables of $L(C)$. Essentially, the difference between a generalized Henkin frame and a generalized Henkin model lies in the requirement that the extension of a formula appearing in a predicate abstract must correspond to the designation of that abstract, which is a member of the appropriate Henkin domain. In $L^+(C)$ there are parameters, so there are more formulas and predicate abstracts than in $L(C)$. Then requiring that something be a generalized Henkin model with respect to $L^+(C)$ is apparently a stronger condition than requiring it be one with respect to $L(C)$, though Section 6 establishes that this is not actually so.

DEFINITION 3.2 (GROUNDED) *A term or a formula of $L^+(C)$ is grounded if it contains no free variables of $L(C)$, though it may contain parameters.*

The notion of grounded extends the notion of closed. Specifically, a grounded formula of $L^+(C)$ that happens to be a formula of $L(C)$ is a closed formula of $L(C)$, and similarly for terms.

2. Basic Tableaus

I now present the *basic* tableau system. It does not contain machinery for dealing with equality—that comes in Chapter 5. The rules come from [Tol75], where they were given for second-order logic. These rules, in turn, trace back to the sequent-style higher-order rules of [Pra68] and [Tak67].

All tableau proofs are proofs of *sentences*—closed formulas—of $L(C)$. A tableau proof of Φ is a tree that has $\neg\Phi$ at its root, grounded formulas of $L^+(C)$ at all nodes, is constructed following certain *branch extension* rules to be given below, and is *closed*, which means it embodies a contradiction. Such a tree intuitively says $\neg\Phi$ cannot happen, and so Φ is valid.

The branch extension rules for propositional connectives are quite straightforward and well-known. Here they are, including rules for various defined connectives.

DEFINITION 3.3 (CONJUNCTIVE RULES)

$$\begin{array}{c} \frac{X \wedge Y}{X} \\ Y \end{array} \quad \frac{\neg(X \vee Y)}{\neg X} \quad \frac{\neg(X \supset Y)}{X} \quad \frac{X \equiv Y}{X \supset Y} \quad \frac{X \supset Y}{Y \supset X}$$

For the conjunctive rules, if the formula above the line appears on a branch of a tableau, the items below the line may be added to the end of the branch. The rule for double negation is of the same nature, except that only a single added item is involved.

DEFINITION 3.4 (DOUBLE NEGATION RULE)

$$\frac{\neg\neg X}{X}$$

Next come the disjunctive rules. For these, if the formula above the line appears on a tableau branch, the end node can have two children added, labeled respectively with the two items shown below the line in the rule. In this case one says there is tableau branching.

DEFINITION 3.5 (DISJUNCTIVE RULES)

$$\frac{X \vee Y}{X \mid Y} \qquad \frac{\neg(X \wedge Y)}{\neg X \mid \neg Y}$$

$$\frac{X \supset Y}{\neg X \mid Y} \qquad \frac{\neg(X \equiv Y)}{\neg(X \supset Y) \mid \neg(Y \supset X)}$$

This completes the propositional connective rules. The motivation should be intuitively obvious. For instance, if $X \wedge Y$ is true in a model, both X and Y are true there, and so a branch containing $X \wedge Y$ can be extended with X and Y . If $X \vee Y$ is true in a model, one of them is true there. The corresponding tableau rule says if $X \vee Y$ occurs on a branch, the branch splits using X and Y as the two cases. One or the other represents the “correct” situation.

Though the universal quantifier has been taken as basic, it is convenient, and just as easy, to have tableau rules for both universal and existential quantifiers directly. To state the rules simply, I use the following convention. Suppose $\Phi(\alpha^t)$ is a formula in which the variable α^t , of type t , may have free occurrences. And suppose τ^t is a term of type t . Then $\Phi(\tau^t)$ is the result of carrying out the substitution $\{\alpha^t/\tau^t\}$ in $\Phi(\alpha^t)$, replacing all free occurrences of α^t with occurrences of τ^t . Now, here are the existential quantifier rules.

DEFINITION 3.6 (EXISTENTIAL RULES) *In the following, p^t is a parameter of type t that is new to the tableau branch.*

$$\frac{(\exists \alpha^t)\Phi(\alpha^t)}{\Phi(p^t)} \qquad \frac{\neg(\forall \alpha^t)\Phi(\alpha^t)}{\neg\Phi(p^t)}$$

The rules above embody the familiar notion of existential instantiation. Since the convention is that parameters are never quantified or λ -bound, we don't have to worry about accidental variable capture. More precisely, in the rules above, the substitution $\{\alpha^t/p^t\}$ is free for the formula $\Phi(\alpha^t)$.

The universal rules are somewhat more straightforward. Once again, note that in them the substitution $\{\alpha^t/\tau^t\}$ is free for the formula $\Phi(\alpha^t)$.

DEFINITION 3.7 (UNIVERSAL RULES) *In the following, τ^t is any grounded term of type t of $L^+(C)$.*

$$\frac{(\forall \alpha^t)\Phi(\alpha^t)}{\Phi(\tau^t)} \qquad \frac{\neg(\exists \alpha^t)\Phi(\alpha^t)}{\neg\Phi(\tau^t)}$$

Finally we have the rules for predicate abstracts. Earlier notation is extended a bit, so that if $\Phi(\alpha_1, \dots, \alpha_n)$ is a formula, $\alpha_1, \dots, \alpha_n$ are distinct free variables, and τ_1, \dots, τ_n are grounded terms of the same respective types as $\alpha_1, \dots, \alpha_n$, then $\Phi(\tau_1, \dots, \tau_n)$ is the result of simultaneously substituting each τ_i for all free occurrences of α_i in Φ .

DEFINITION 3.8 (ABSTRACT RULES)

$$\frac{\langle \lambda \alpha_1, \dots, \alpha_n. \Phi(\alpha_1, \dots, \alpha_n) \rangle (\tau_1, \dots, \tau_n)}{\Phi(\tau_1, \dots, \tau_n)} \\ \frac{\neg \langle \lambda \alpha_1, \dots, \alpha_n. \Phi(\alpha_1, \dots, \alpha_n) \rangle (\tau_1, \dots, \tau_n)}{\neg \Phi(\tau_1, \dots, \tau_n)}$$

Now what, exactly, constitutes a proof.

DEFINITION 3.9 (CLOSURE) *A tableau branch is closed if it contains Φ and $\neg\Phi$, where Φ is a grounded formula. A tableau is closed if each branch is closed.*

DEFINITION 3.10 (TABLEAU PROOF) *For a sentence Φ of $L(C)$, a closed tableau beginning with $\neg\Phi$ is a proof of Φ .*

DEFINITION 3.11 (TABLEAU DERIVATION) *A tableau derivation of a sentence Φ from a set of sentences S , all of $L(C)$, is a closed tableau beginning with $\neg\Phi$, allowing the additional rule: at any point any member of S can be added to the end of any open branch.*

This concludes the presentation of the tableau rules. In the next section I give several examples of tableaux. Classical first-order tableau rules, as in [Smu68, Fit96] are *analytic*—they only involve subformulas of the formula being proved. (It is not the case with the *cut* rule, but this is an eliminable rule.) Higher-order rules, for the most part, have an analytic nature as well. The important exception is the rule for the universal quantifier. It allows us to pass from $(\forall \alpha^t)\Phi(\alpha^t)$ to $\Phi(\tau^t)$ where τ^t is an arbitrary grounded term. Since terms can involve predicate abstracts, applications of this rule can introduce formulas that are not subformulas of the one being proved—indeed, they may be much more complicated. There is no way around this. In a sense, the introduction of predicate abstracts embodies the “creative element” of mathematics.

3. Tableau Examples

Tableaus for first-order classical logic are well-known, but the abstraction rules of the previous section are not as widely familiar. I give



<http://www.springer.com/978-1-4020-0604-3>

Types, Tableaus, and Gödel's God

Fitting, M.

2002, XV, 181 p., Hardcover

ISBN: 978-1-4020-0604-3