Tableaux for the Logic of Proofs

Bryan Renne

Dept. of Computer Science
CUNY Graduate Center
365 Fifth Avenue
New York, NY 10016
email: brenne@gc.cuny.edu
web: http://cs.gc.cuny.edu/∼brenne/

Abstract. The Logic of Proofs, **LP**, is an explicit provability logic due to Artemov. The introduction of **LP** answered a long-standing question concerning the intended semantics of Gödel's provability calculus and provability semantics for intuitionistic logic. The explicit nature of **LP** and its ability to naturally represent both modal logic and typed λ-calculi, especially in light of the Curry-Howard Isomorphism, makes its applicability to Computer Science a primary focus of research in this area. In the present paper, I develop a tableau system for **LP** and give a semantic proof of cut elimination.

1 Introduction

Artemov's Logic of Proofs, **LP**, provides Gödel's intended connection between intuitionistic logic and classical proofs by combining propositions and proofs into the same calculus of proof-carrying formulas while retaining an appropriate provability semantics [1]. Proof terms in **LP**, called *proof polynomials*, are given an explicit provability reading: "$t : X$," where t is a proof term and X is a formula, is read as, "t is a proof of X."

LP naturally subsumes both modal logic and typed λ-calculi ([1]) in that it can code proofs and programs in a single system. In light of the Curry-Howard Isomorphism, this combination of modal iteration with the explicit character of λ-terms indicates the importance of **LP** in Computer Science (see Artemov's keynote addresses for CLS'03 and ESSLLI'03 [2, 3]).

After recounting the Hilbert-style theory of **LP**, I will present a tableau proof system for **LP** in the style of Smullyan (see [4]). I will show that this tableau system is sound and complete with respect to a natural semantics for **LP** that was developed by Mkrtychev in [5] (a survey of Mkrtchev's models, known as M-models, can be found in [6]). The semantic proof of completeness will also yield cut elimination in **LP** tableaux.

As Smullyan observes in [4], certain Gentzen-style theories are closely related to tableaux (and vice versa), and **LP** tableaux will share such a relationship with Artemov's cut-free Gentzen-style theory of **LP**, $\mathbf{LPG}^-$ (see [1]). Similarly, **LP** tableaux with cut are related to the Gentzen-style theory with cut, **LPG**. Cut elimination for **LP** tableaux then implies admissibility of cut in $\mathbf{LPG}^-$, so cut can be eliminated from **LPG**.

Using syntactic methods, cut elimination for a fragment of **LPG** was established in [1], where cut elimination for the whole of **LPG** was stated with only a brief outline of a proof. The current paper provides the first detailed proof of cut elimination in **LP**.

2 The logic and its tableau system

2.1 Syntax

The "modals" in **LP** are called *terms* and are built up inductively from the *atomic* terms: *proof constants* c_i and *proof variables* x_i, where $i \in \mathbb{N}$.

Definition 1 (Terms). *For each $i \in \mathbb{N}$, both c_i and x_i are* atomic *terms. If t and s are terms, then so is each of $(t+s)$, $(!t)$, and $(t \cdot s)$. Let $\mathcal{T}$ denote the set of* **LP** *terms.*

Formulas in **LP** are built up from propositional variables A_i, where $i \in \mathbb{N}$, and the absurdity constant $\perp$ in the usual way using implication $\to$. Modal-like formulas are introduced by combining a formula X and a term t to produce $t{:}X$.

Definition 2 (Formulas). *$\perp$ is a formula. For each $i \in \mathbb{N}$, A_i is a formula. Whenever X and Y are formulas and t is a term, both $(X \to Y)$ and $(t{:}X)$ are formulas. Let $\mathcal{F}$ denote the set of* **LP** *formulas.*

Definition 3 (LP). *The axioms of* **LP** *are:*

1. *Finite set of axiom schemes for classical logic using $\to$ and $\perp$*
2. *$t{:}X \to X$, "reflection"*
3. *$t{:}(X \to Y) \to (s{:}X \to (t \cdot s){:}Y)$, "application"*
4. *$t{:}X \to !t{:}(t{:}X)$, "proof checking"*
5. *$t_i{:}X \to (t_0 + t_1){:}X$, where $i \in \{0, 1\}$, "sum"*

The rules of inference of **LP** *are modus ponens and* constant necessitation*: from any axiom $\mathcal{A}$, infer $c_i{:}\mathcal{A}$, where $i \in \mathbb{N}$.*

2.2 Semantics

Mkrtychev showed in [5] that the Hilbert-style theory presented above is sound and complete with respect to his semantics. This semantics assigns to each term t a set of formulas $*(t)$, which may be finite, infinite, or empty. In an M-model, $X \in *(t)$ will be taken to mean that $t{:}X$ holds. To satisfy reflection, M-models also require that whenever $X \in *(t)$ in a model then X will also hold in the model. The function $*$ is called a *proof-theorem assignment.*

Definition 4. *A* proof-theorem assignment *is a map $* : \mathcal{T} \to 2^{\mathcal{F}}$ satisfying:*

1. $(X \rightarrow Y) \in *(t)$ *and* $X \in *(s)$ *implies* $Y \in *(t \cdot s)$
2. $*(t) \cup *(s) \subseteq *(t+s)$
3. $X \in *(t)$ *implies* $(t{:}X) \in *(!t)$

Since there are no restrictions on the formulas that may be assigned to proof variables x_i, it is evident that the terms x_i intuitively serve as variables, since, in a given proof, each x_i can be taken to prove any desired set of formulas. Proof constants c_i similarly have little restriction, though for the moment it will be most instructive to consider them as labels of axioms.

It is also already apparent that a term produced from atomic terms using $+$, $\cdot$, and ! encodes (in some sense) a particular proof that makes use of the facts represented by the atomic terms. In particular, if a compound term t is built up from proof constants $c_1, \ldots, c_n$ and proof variables $x_1, \ldots, x_m$, then t encodes some proof involving the axioms encoded in $c_1, \ldots, c_n$ and the assumptions encoded in $x_1, \ldots, x_m$. This proof is evidence for those formulas contained in $*(t)$. For an explicit description of an arithmetical encoding of proofs and how this brings about an arithmetic semantics for **LP**, see [1].

Definition 5. *A* pre-model *is a pair* $M = (v, *)$, *where* v *is a truth assignment to the propositional variables* A_i. *The pair uniquely determines the usual one-place semantic forcing relation* $\vDash_M$ *on* **LP** *formulas satisfying:*

1. $\nvDash_M \bot$
2. $\vDash_M A_i$ *iff* $v(A_i)$ *is true, for* A_i *a propositional variable*
3. $\vDash_M X \rightarrow Y$ *iff* $\nvDash_M X$ *or* $\vDash_M Y$
4. $\vDash_M t{:}X$ *iff* $X \in *(t)$

Definition 6. *A pre-model* M *is* reflexive *if* $X \in *(t)$ *implies* $\vDash_M X$.

Definition 7. *A* model *is a reflexive pre-model.*

Definition 8. X *is* true *in a (pre-)model* M *when* $\vDash_M X$. *If for every model* M *it is the case that* $\vDash_M X$, *then* X *is* valid.

Definition 9. *A* constant specification *is a set* CS *of formulas of the form* $c_i : \mathcal{A}$, *where* $i \in \mathbb{N}$ *and* $\mathcal{A}$ *is an* **LP** *axiom. An* **LP** *model* M *is said to be a* CS-model *if for every* $X \in CS$, $\vDash_M X$.

Definition 9 describes the sense in which **LP** models are relativized to a particular set of named axioms, where a proof constant serves as a name of a particular axiom. When it is important to highlight the fact that the theory is relativized to a particular constant specification CS, I will denote the theory by $\mathbf{LP}^{CS}$.

2.3 Tableaux

The tableau method of proof is really a search for a countermodel. A tableau itself is a tree whose nodes are labeled by formulas adjoined with some bookkeeping convention on labels to determine whether the search considers a formula true.[1] To this end, I will prefix formulas by either **T** or **F**. Given a tableau, it is the intention that, under suitable conditions, a *branch* in the tree—that is, a path from the root to a leaf—corresponds to a model that makes those **T**-prefixed formulas occurring on the branch true and those with an **F**-prefix false.

A branch fails to be a countermodel when that branch cannot represent a model. This occurs when the branch is *closed*, which means it is contradictory in that it contains both $\mathbf{T}X$ and $\mathbf{F}X$ for some X or it contains $\mathbf{T}\bot$.

A tableau search for a model of a finite set S of prefixed formulas begins with the construction of a linear tree whose nodes consist of the formulas in S. This is an initial tableau, called the tableau *beginning with* S. Then, given a tableau τ, when a prefixed formula at a node n in the tree has the structure of one of the tableau rules, I may apply the rule to produce a larger tree. This larger tree is also called a tableau: it is the tableau *generated* or *produced* from τ. Using an extension of the notation of Smullyan (see [4]), **LP** tableau rules are of the type α, β, or β^X. Let m be any leaf node occurring on a branch containing n. In the α case, either of the prefixed formulas α_1 or α_2 may be attached as a child of m to produce another tableau (though it is typically the case that one is attached as a child of m and the other is attached as a child of the first). In the β case, both prefixed formulas β_1 and β_2 are attached as children of m to produce another tableau. The β^X case is similar to the β case.[2] See Fig. 1 for a graphic representation of the α, β, and β^X rules. These rules are applied successively until all branches are closed or until application of a rule fails to add any new nodes to a branch. To prove X, I set $S = \{\mathbf{F}X\}$.

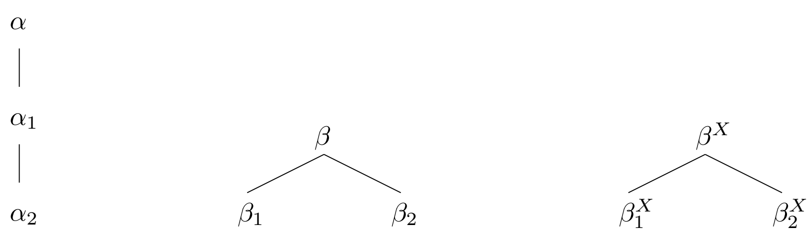

Fig. 1. Definition of the α, β, and β^X rules

[1] Note that I will often use a node's label as its name, even though a tableau may have different nodes with the same label. In this case, fixing any specific node with this name for the duration of the discussion is sufficient to avoid ambiguity. This is because any node with the specified name will do during such discussions, as long as the node remains fixed.

[2] Note that the α, β, and β^X formulas need not occur at leaves, as n is taken anywhere in the tree.

The α, β, and β^X formulas, whose rules were mentioned immediately above, are defined in Fig. 2. The β^X rule for products of terms has a parameter X, which indicates what is to be the antecedent of the implication in the β_1 case. X can be any **LP** formula.

α	α_1	α_2
$\mathbf{F}X \to Y$	$\mathbf{T}X$	$\mathbf{F}Y$
$\mathbf{F}(s+t){:}X$	$\mathbf{F}s{:}X$	$\mathbf{F}t{:}X$
$\mathbf{T}t{:}X$	$\mathbf{T}X$	$\mathbf{T}X$
$\mathbf{F}!t{:}(t{:}X)$	$\mathbf{F}t{:}X$	$\mathbf{F}t{:}X$

And, if $c_i{:}\mathcal{A} \in CS$,

α	α_1	α_2
$\mathbf{F}c_i{:}\mathcal{A}$	$\mathbf{F}\mathcal{A}$	$\mathbf{F}\mathcal{A}$

β	β_1	β_2
$\mathbf{T}X \to Y$	$\mathbf{F}X$	$\mathbf{T}Y$

β^X	β_1^X	β_2^X
$\mathbf{F}(s \cdot t){:}Y$	$\mathbf{F}s{:}(X \to Y)$	$\mathbf{F}t{:}X$

Fig. 2. Definition of α, β, and β^X formulas. Note that the *constant rule*—the α rule for prefixed formulas of the form $\mathbf{F}c_i : \mathcal{A}$—may only be applied when $\mathcal{A}$ is an axiom labeled by c_i in the constant specification CS.

The *constant rule*—the α-rule operating on formulas of the form $\mathbf{F}c_i : A$—is of particular consequence to **LP** tableaux. Since **LP** models are relativized to a particular constant specification CS, tableau search must also be accordingly relativized. That is, the tableau search for a model requires a constant specification CS as a parameter. The CS parameter indicates when the constant rule may be applied, since the tableau search is to produce an $\mathbf{LP}^{CS}$ countermodel.

A branch θ is *closed* whenever one of the following holds:

1. $\mathbf{T}\bot$ appears on θ
2. For some formula X, both $\mathbf{T}X$ and $\mathbf{F}X$ appear on θ

A tableau is *closed* if every branch is closed. If a branch or tableau is not closed, it is *open*. For a formula X, a *tableau proof* of X is a closed tableau beginning with $\mathbf{F}X$. A branch θ of a tableau is *satisfiable* if there is a model M such that $\models_M X$ if $\mathbf{T}X$ appears on θ and $\not\models_M X$ if $\mathbf{F}X$ appears on θ. A tableau is *satisfiable* if one of its branches is satisfiable. The next two lemmas are consequences of the definition of M-models.

Lemma 1. *If a model M satisfies α, then it satisfies both α_1 and α_2.*

Lemma 2. *If a model M satisfies β, then it satisfies β_1 or β_2. Also, if a model M satisfies β^X, then it satisfies β_1^X or β_2^X.*

Example 1. Let the propositional connective $\vee$ be defined in the usual way as $X \vee Y := (X \rightarrow \bot) \rightarrow Y$. It can be easily verified that the following derived α and β rules follow for $\vee$:

α	α_1	α_2
$\mathbf{F}X \vee Y$	$\mathbf{F}X$	$\mathbf{F}Y$

β	β_1	β_2
$\mathbf{T}X \vee Y$	$\mathbf{T}X$	$\mathbf{T}Y$

Take the constant specification

$$CS = \{c_1 : (x_1 : A_1 \rightarrow (x_1 : A_1 \vee x_2 : A_2)), c_2 : (x_2 : A_2 \rightarrow (x_1 : A_1 \vee x_2 : A_2))\} \ .$$

Then, using the derived rules, Fig. 3 is a proof of the formula

$$(x_1 : A_1 \vee x_2 : A_2) \rightarrow (c_1 x_1 + c_2 x_2) : (x_1 : A_1 \vee x_2 : A_2) \ .$$

The first branching in the tableau is caused by application of the derived β rule for $\vee$, which is applied to $\mathbf{T}x_1 : A_1 \vee x_2 : A_2$, producing $\beta_1 = \mathbf{T}x_1 : A_1$ and $\beta_2 = \mathbf{T}x_2 : A_2$. Below this β_2, $\beta^{x_2 : A_2}$ is applied to $\mathbf{F}(c_2 x_2) : (x_1 : A_1 \vee x_2 : A_2)$. The antecedent $x_2 : A_2$ is chosen because this closes both of the produced branches. The right branch is closed because it contradicts its parent. Because c_2 is indeed a label of the axiom $x_2 : A_2 \rightarrow (x_1 : A_1 \vee x_2 : A_2)$, application of the constant rule is possible. Since the resulting α is an axiom labeled by $\mathbf{F}$, this branch also closes (as is shown). The case of the $\mathbf{T}x_1 : A_1$-side of the tree is similar.

Since every branch of the tableau closes, there can be no model of any branch, and hence there can be no model of the root of this tree. Therefore, the formula I wished to prove cannot be false in a model, and hence the formula holds in every model.[3]

2.4 Soundness and Completeness

Together, Lemmas 1 and 2 imply the following corollary, which I will make use of in the proof of soundness.

Corollary 1. *If a tableau τ is satisfiable, then for any tableau τ' produced from τ by application of a tableau rule, τ' is also satisfiable.*

Theorem 1 (Soundness). *If there is a tableau proof of a formula X under a constant specification CS, then X has a CS-model.*

[3] Of course these final sentences depend on the forthcoming soundness result.

- $\mathbf{F}(x_1 : A_1 \vee x_2 : A_2) \to (c_1 \cdot x_1 + c_2 \cdot x_2) : (x_1 : A_1 \vee x_2 : A_2)$
- $\mathbf{T}x_1 : A_1 \vee x_2 : A_2$
- $\mathbf{F}(c_1 \cdot x_1 + c_2 \cdot x_2) : (x_1 : A_1 \vee x_2 : A_2)$
- $\mathbf{F}(c_1 \cdot x_1) : (x_1 : A_1 \vee x_2 : A_2)$
- $\mathbf{F}(c_2 \cdot x_2) : (x_1 : A_1 \vee x_2 : A_2)$
  - $\mathbf{T}x_1 : A_1$
    - $\mathbf{F}c_1 : (x_1 : A_1 \to (x_1 : A_1 \vee x_2 : A_2))$
      - $\mathbf{F}x_1 : A_1 \to (x_1 : A_1 \vee x_2 : A_2)$
      - $\mathbf{T}x_1 : A_1$
      - $\mathbf{F}x_1 : A_1 \vee x_2 : A_2$
      - $\mathbf{F}x_1 : A_1$
      - $\mathbf{F}x_2 : A_2$
      - $\otimes$
    - $\mathbf{F}x_1 : A_1$
      - $\otimes$
  - $\mathbf{T}x_2 : A_2$
    - $\mathbf{F}c_2 : (x_2 : A_2 \to (x_1 : A_1 \vee x_2 : A_2))$
      - $\mathbf{F}x_2 : A_2 \to (x_1 : A_1 \vee x_2 : A_2)$
      - $\mathbf{T}x_2 : A_2$
      - $\mathbf{F}x_1 : A_1 \vee x_2 : A_2$
      - $\mathbf{F}x_1 : A_1$
      - $\mathbf{F}x_2 : A_2$
      - $\otimes$
    - $\mathbf{F}x_2 : A_2$
      - $\otimes$

Fig. 3. A tableau proof of the formula in Example 1. The tableau makes use of the derived $\vee$ rules. Closed branches are indicated by $\otimes$.

Proof. If X does not have a CS-model, then there is a CS-model M such that $\not\models_M X$. Thus the tableau τ consisting of only $\mathbf{F}X$ is satisfiable, for M satisfies it. But by Corollary 1, every tableau produced from τ is also satisfiable. Therefore no tableau beginning with $\mathbf{F}X$ can close, so there can be no tableau proof of X. □

Completeness is proven using a semantic method that makes use of a tableau-specific notion of consistency and has the look of a Lindenbaum argument. Specifically,

Definition 10. *For a finite set S of prefixed* **LP** *formulas, S is* consistent *if no tableau beginning with S closes. For S an infinite set, S is* consistent *if every finite subset is consistent. A set is* inconsistent *if it is not consistent.*

After mentioning the following standard lemma, I will introduce an additional, related notion that will be technically convenient.

Lemma 3. *Every consistent set can be extended to a maximally consistent set.*

Definition 11. *A set S of prefixed* **LP** *formulas is* downward saturated *when each of the following holds:*

1. *S is consistent*
2. *$\alpha_1, \alpha_2 \in S$ if $\alpha \in S$*
3. *$\beta_1 \in S$ or $\beta_2 \in S$ if $\beta \in S$*
4. *for every* **LP** *formula X, $\beta_1^X \in S$ or $\beta_2^X \in S$ if $\beta^X \in S$*

Lemma 4. *Any maximally consistent set is downward saturated.*

Proof. By a straight-forward verification of each case. □

Lemma 5. *Suppose a set S is downward saturated. If $S \cup \{\mathbf{F}X\}$ is inconsistent, then there is a closed tableau τ for a finite subset, where τ is produced without applying any tableau rule to a member of S.*

Proof. If $S \cup \{\mathbf{F}X\}$ is inconsistent, there is a finite $S' \subseteq S$ such that application of some tableau rules to the tableau beginning with $S' \cup \{\mathbf{F}X\}$ produces a closed tableau. I wish to show that application of a rule to a member of S is unnecessary; that is, whenever τ is a closed tableau produced from the tableau beginning with $S' \cup \{\mathbf{F}X\}$, there exists a closed tableau τ' such that fewer rules are applied to members of S in τ' than in τ'. τ' will be constructed from τ, so well-ordering of the naturals combined with the fact that the result holds when no rules are applied to a member of S implies the overall result.

If no rule was applied to a member of S in τ, then I am done. So assume that some rule is applied to a member of S in τ. I must then consider each of the cases α, β, and β^X, though the β and β^X cases are the same.

In the α case, I can take $S_1 := S' \cup \{\alpha_1, \alpha_2\}$ and consider τ' as the tableau beginning with $S_1 \cup \{\mathbf{F}X\}$, which then has one less tableau rule applied to a member of S than does τ.

In the β^X case, where X is some **LP** formula, since $S' \cup \{\mathbf{F}X\}$ is inconsistent, both the tableau beginning with $S' \cup \{\mathbf{F}X, \beta_1^X\}$ and the tableau beginning with $S' \cup \{\mathbf{F}X, \beta_2^X\}$ close. Let $S_1 := S' \cup \{\beta_1^X\}$ and $S_2 := S' \cup \{\beta_2^X\}$; both S_1 and S_2 are finite. S is downward saturated, so $\beta_1^X \in S$ or $\beta_2^X \in S$, and since S is also consistent, S_1 or S_2 is consistent. For concreteness, take $\beta_1^X \in S$. Notice that the β^X rule need not be applied, for let τ' the tableau that results from trimming off the β_2^X subtree of τ, and thus τ' is a closed tableau beginning with $S_1 \cup \{\mathbf{F}X\}$ that has fewer rules applied to a member of S than does τ. The β case is handled similarly. $\square$

Theorem 2 (Completeness). *If X has a CS-model, then X has a tableau proof.*

Proof. Suppose X has no tableau proof. Then the set $\{\mathbf{F}X\}$ is consistent and, by Lemma 3, can be extended to a maximally consistent set S. By Lemma 4, S is downward saturated.

I will construct a CS-model M so that if $\mathbf{T}Y \in S$ then $\vDash_M Y$ and if $\mathbf{F}Y \in S$ then $\nvDash_M Y$, which yields the result, since $\mathbf{F}X \in S$. To specify the model $M = (v, *)$, I must specify v and $*$. Set $v(A_i)$ true if and only if $\mathbf{T}A_i \in S$, where A_i is a propositional variable. For any proof term t, let $*(t) = \{X \mid \mathbf{F}t\!:\!X \notin S\}$. $*$ satisfies the properties of a proof-theorem assignment:

1. Assume that $(Y \rightarrow Z) \in *(t)$ and $Y \in *(s)$. This means $\mathbf{F}t\!:\!(Y \rightarrow Z) \notin S$ and $\mathbf{F}s : Y \notin S$. S is downward saturated, so it cannot be the case that $\mathbf{F}(t \cdot s)\!:\!Z \in S$. So, $Z \in *(t \cdot s)$.
2. Assume that $Y \in *(t)$ or $Y \in *(s)$. This means $\mathbf{F}t\!:\!Y \notin S$ or $\mathbf{F}s\!:\!Y \notin S$. S is downward saturated, so it cannot be the case that $\mathbf{F}(t + s) : Y \in S$. So, $(t + s)\!:\!Y \in S$.
3. Assume that $t\!:\!Y \in *(t)$. This means $\mathbf{F}t\!:\!Y \notin S$. S is downward saturated, so it cannot be the case that $\mathbf{F}!t\!:\!(t\!:\!Y) \in S$. So, $(t\!:\!Y) \in *(!t)$.

Thus, M is a pre-model. Notice that M also satisfies each formula in CS, since every member of CS is an axiom labeled by a proof constant.

I now wish to show by induction on the complexity of the formula Y that $\mathbf{T}Y \in S$ implies $\vDash_M Y$ and $\mathbf{F}Y \in S$ implies $\nvDash_M Y$ for M. In the base case, Y is atomic and is thus either A_i or $\bot$. The former case is satisfied by definition of v and the latter can only occur when $\mathbf{F}\bot \in S$, which M trivially satisfies. I now assume that the result holds for formulas of complexity less than Y and prove it also holds for Y. Each case is considered separately:

1. Y is $W \rightarrow Z$.
   (a) $\mathbf{T}W \rightarrow Z \in S$ implies $\mathbf{F}W \in S$ or $\mathbf{T}Z \in S$ since S is downward saturated. By the inductive hypothesis, $\nvDash_M W$ or $\vDash_M Z$, and thus $\vDash_M W \rightarrow Z$ by definition.
   (b) If $\mathbf{F}W \rightarrow Z \in S$, then $\mathbf{T}W \in S$ and $\mathbf{F}Z \in S$ by downward saturation. By the inductive hypothesis, $\vDash_M W$ and $\nvDash_M Z$, and thus $\nvDash_M W \rightarrow Z$ by definition.

2. Y is $t\!:\!Z$.
   (a) $\mathbf{T}t\!:\!Z \in S$ implies $\mathbf{F}t\!:\!Z \notin S$, since S is consistent. Thus, $Z \in *(t)$ and $\vDash_M t\!:\!Z$.
   (b) If $\mathbf{F}t\!:\!Z \in S$, by definition of the proof-theorem assignment $*$, $Z \notin *(t)$. Therefore, $\nvDash_M t\!:\!Z$.

Hence the result holds for all formulas Y.

What now remains is to show that the pre-model M is in fact a model, which will complete the proof. That is, I must show

$$Y \in *(t) \text{ implies } \vDash_M Y \ . \tag{1}$$

I will show this by induction on the complexity of t. Recall that $Y \in *(t)$ is equivalent to saying $\mathbf{F}t\!:\!Y \notin S$.

In the base case of this induction, t is atomic and is thus either c_i or x_i. In the former case, the maximal consistency of S implies that if $\mathbf{F}c_i\!:\!Y \notin S$ then $S \cup \{\mathbf{F}c_i\!:\!Y\}$ is inconsistent. By Lemma 5, there is a closed tableau for a finite subset without applying any rules to a member of S. There are two subcases:

- If $c_i\!:\!Y \in CS$, then Y is an axiom $\mathcal{A}$. But every axiom is true in every model and thus $\vDash_M \mathcal{A}$.
- If $c_i\!:\!Y \notin CS$, no tableau rule applies, so it must be the case that $\mathbf{T}c_i\!:\!Y \in S$. Since S is downward saturated, $\mathbf{T}Y \in S$, and I have already shown that this implies $\vDash_M Y$.

The case for which t is the atomic term x_i is the same as the second subcase above.

I now assume that the property (1) holds of all terms s of complexity less than t and show it also holds for t. By the maximal consistency of S, if $\mathbf{F}t\!:\!Y \notin S$, then $S \cup \{\mathbf{F}t : Y\}$ is inconsistent. If $\mathbf{T}t : Y \in S$, then downward saturation implies $\mathbf{T}Y \in S$ and hence $\vDash_M Y$ by what I have already shown. So assume that $\mathbf{T}t\!:\!Y \notin S$. By Lemma 5, there exists a closed tableau τ for a finite subset of $S \cup \{\mathbf{F}t\!:\!Y\}$ without applying any rule to a member of S. If no tableau rule applies to a formula of the form $\mathbf{F}t\!:\!Y$, then it must be the case that $\mathbf{T}t\!:\!Y \in S$, which was already handled. So assume some tableau rule applies to $\mathbf{F}t\!:\!Y$.

- If this rule is an α, then since $\mathbf{T}t\!:\!Y \notin S$ and τ is closed, $S \cup \{\alpha_1, \alpha_2\}$ must also be inconsistent. So $\alpha_1, \alpha_2 \notin S$. But each α_i is of the form $\mathbf{F}s : Y$ for some term s of less complexity than t and hence, by the inductive hypothesis, $\vDash_M Y$.
- If this rule is a β, then since $\mathbf{T}t\!:\!Y \notin S$ and τ is closed, $S \cup \{\beta_1\}$ or $S \cup \{\beta_2\}$ must also be inconsistent. So $\beta_1 \notin S$ or $\beta_2 \notin S$. But each β_i is of the form $\mathbf{F}s\!:\!Y$ for some term s of less complexity than t and since at least one $\beta_i \notin S$, the inductive hypothesis implies $\vDash_M Y$.
- The β^X case is the same as the β case.

Hence Equation 1 holds of all terms t and all formulas Y.

So, I have shown that M is indeed a CS-model (of $\{\mathbf{F}X\}$) and thus the proof is complete. □

2.5 Cut elimination

In tableaux, the cut rule allows a new tableau to be produced from any existing tableau by choosing any leaf node m and adding both $\mathbf{T}X$ and $\mathbf{F}X$ as children of m, where X is some **LP** formula; see Fig. 4. The following corollary is a result originally proved by Artemov via syntactic methods [1]. Tableaux give an alternative proof.

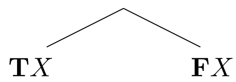

Fig. 4. The cut rule

Corollary 2. *Cut is an admissible rule.*

Proof. Cut is clearly a sound rule. However, anything that is provable with cut is also provable without, since tableaux without cut are complete. □

LPG is the Gentzen-style system describing **LP**, and $\mathbf{LPG}^-$ is **LPG** without the Gentzen cut rule. $\mathbf{LPG}^-$ and the present tableau system are related in the sense described by Smullyan in Chapter XI of his book [4]. In particular, any tableau proof can be converted to a block tableau proof, which can be converted to a derivation in $\mathbf{LPG}^-$. The converse also holds, as this argument is invertible. Thus, $\mathbf{LPG}^-$ is sufficient to capture all of **LP** (equivalently, all of **LPG**).

2.6 Tableaux for weaker systems

In Fitting's Kripke-style semantics for **LP** (see [7]), **LP** terms are considered as possible evidence for the formulas to which they are attached; that is, "$t : F$" is read, "t is an evidence for F." In a system strong enough to internalize its own proofs, a proof is a rather demanding form of evidence. It may be the case that such a high degree of reflection requires too much in some applications. In these cases, sublogics of **LP** that describe weaker notions of evidence may be more appropriate. Any such sublogic has its own tableau system, which is found by considering only those α, β, and β^X rules that correspond to the particular sublogic.

Example 2. The sublogic of **LP** that does not use $+$ can be considered a quasi-single-conclusion sublogic, in that the multiplicity of formulas for which a term is evidence is dictated by the constant specification; this is because evidences cannot be disjunctively combined: given terms t and s, I cannot combine them to produce a term u that is evidence for those statements for which either t or s is evidence. The tableau system for this logic is the same as that presented above, sans the α rule for $+$ in terms.

3 Acknowledgments

I am greatly indebted to Professor Melvin Fitting for his tireless editing of the various versions of this paper and also for his insightful suggestions and observations. Additionally, I wish to thank Professor Sergei Artemov for his editorial assistance, support, and his many teachings, including his lectures on **LP**.

References

1. Artemov, S.N.: Explicit provability and constructive semantics. The Bulletin of Symbolic Logic **7** (2001)
2. Artemov, S.N.: Back to the future: explicit logic for Computer Science. Lecture Notes in Computer Science **2803** (2003) 43 Computer Science Logic 2003 Invited Lecture.
3. Artemov, S.N.: Back to the future: explicit logic for Computer Science. Special invited lecture at European Summer School for Language, Logic, and Information (ESSLLI) (2003)
4. Smullyan, R.M.: First-Order Logic. Dover Publications, Inc., New York (1995)
5. Mkrtychev, A.: Models for the logic of proofs. Lecture Notes in Computer Science **1234** (1997) 266–275
6. Kuznets, R.: On the complexity of explicit modal logics. Lecture Notes in Computer Science **1862** (2000) 371–383
7. Fitting, M.: A semantics for the Logic of Proofs. Technical Report TR-2003012, City University of New York (2003) http://www.cs.gc.cuny.edu/tr/.