Semantic cut-elimination for two explicit modal logics

Bryan Renne

Computer Science, CUNY Graduate Center

bryan@renne.org

Abstract. Explicit modal logics contain modal-like terms that label formulas in a way that mimics deduction in the system. These logics have certain proof-theoretic advantages over the usual modal logics, perhaps the most important of which is conventional cut-elimination.

The present paper studies tableau proof systems for two explicit modal logics, LP and S4LP. Using a certain method to prove the correctness of these systems, we obtain a semantic proof of cut-elimination for these logics.

1 Introduction

Explicit modal logics differ from ordinary modal logic in that the former introduce formula-labeling terms into the language of propositional logic. These terms label formulas in a way that mimics deduction in the system, so the terms may be thought of as reasons or evidence as to why a formula holds (or is known). In this approach, if t is such a term and φ is a formula, then $t{:}\varphi$ is a new formula whose epistemic reading is "φ is known for reason t." Compare this with the epistemic reading of the usual modal formula $\Box\varphi$: "φ is known (for some reason)."

The present paper studies two explicit modal logics. The first is the the most elementary explicit modal logic, the Logic of Proofs (LP). The second is S4LP, whose language extends that of LP by introducing an S4 modality. We define tableau systems for both LP and S4LP, and prove the correctness (soundness and completeness) of these systems. As a corollary of the completeness argument, we also obtain a semantic proof of cut-elimination for both LP and S4LP. In the case of S4LP, this answers affirmatively the question left open by Fitting in (Fitting 2004) as to whether S4LP is cut-free.

We now present the Hilbert-style theories LP and S4LP.

Proceedings of the Eleventh ESSLLI Student Session
Janneke Huitink & Sophia Katrenko (editors)

2 The syntax

2.1 The logic LP

The language of LP extends that of propositional logic by introducing a countable collection of *proof variables* $x_1, x_2, x_3, \ldots$, a countable collection of *proof constants* $c_1, c_2, c_3, \ldots$, the binary function symbols $+$ and $\cdot$, and the unary function symbol $!$. *Proof terms* are built up from proof variables and proof constants using the function symbols. The rules of formula formation are those of propositional logic in addition to the following: if t is a proof term and φ is an LP formula, then $t{:}\varphi$ is also an LP formula. Note that proof terms will sometimes be called *evidence terms* or perhaps even *terms*. Now, letting t and s be arbitrary terms and φ and ψ be arbitrary formulas, the theory LP consists of the following axiom and rule schemas:

- *Propositional logic*

  PL. A finite collection of axiom schemas for propositional logic

  RPL. Modus ponens: infer ψ from $\varphi \supset \psi$ and φ

- *Evidence management*

  LP1. $t{:}(\varphi \supset \psi) \supset (s{:}\varphi \supset (t \cdot s){:}\psi)$

  LP2. $t{:}\varphi \supset\, !t{:}(t{:}\varphi)$

  LP3. $t{:}\varphi \vee s{:}\varphi \supset (t + s){:}\varphi$

  LP4. $t{:}\varphi \supset \varphi$

  RLP. Constant necessitation: infer $c{:}A$ for A an axiom and c a proof constant

LP1 is the property of *application* for evidence terms, which is an internalized modus ponens: if t is evidence for an implication and s is evidence for the antecedent, then $t \cdot s$ is evidence for the consequent. **LP2** is the property of *proof checking*: if t is evidence for φ, then $!t$ (read "bang t") is evidence for the fact that $t{:}\varphi$, so $!t$ verifies that indeed t is evidence for φ. **LP3** is a *sum* or monotonicity property: if t is evidence for φ, then combining t with the information in s to produce either $t + s$ or $s + t$ yields something that is still evidence for φ.[1] **LP4** is an explicit *reflection* property: if t is evidence for φ, then φ must be true. **RLP** says that the proof constants are atomic reasons for the most basic facts, the axioms. Since constants serve as justification for our basic facts, they may be viewed as the simplest sort of justification.

The following *internalization property* can be shown by induction on the length of the derivation in LP: for every LP theorem φ, there is an evidence term t containing no proof variables such that $t{:}\varphi$ is an LP theorem. This provides a sense in which LP encodes its own derivations using evidence terms, which bolsters the intuitive conception of terms as reasons or evidence.

[1] Notice that $+$ is not commutative. In particular, $(t + s){:}\varphi \supset (s + t){:}\varphi$ is not a theorem of LP.

2.2 The logic S4LP

The language of S4LP extends that of LP by adding a unary S4 modality $\Box$. The rules of formula formation are those of LP in addition to the following: if φ is an S4LP formula, then so is $\Box\varphi$. Now, letting t be an arbitrary term and φ and ψ be arbitrary formulas, the theory S4LP consists of the axiom and rule schemas of LP in addition to the following:

- S4

  K1. $\Box(\varphi \supset \psi) \supset (\Box\varphi \supset \Box\psi)$

  K2. $\Box\varphi \supset \varphi$

  K3. $\Box\varphi \supset \Box\Box\varphi$

  RK. $\Box$ necessitation: infer $\Box\varphi$ from φ

- *Connection Principle*

  C. $t{:}\varphi \supset \Box\varphi$

Assigning $\Box\varphi$ the epistemic reading, "φ is known," the Connection Principle can be read, "If φ is known for a reason, then φ is known." Note that the internalization property also holds of S4LP.

3 The semantics

LP has an arithmetic semantics (Artemov 2001), a minimal semantics (Mkrtychev 1997), a Kripke-style semantics (Fitting 2003; Artemov 2006; Artemov 2004; Fitting 2005), and a game semantics (Renne 2006). In this paper, we will make use of the Kripke-style semantics—otherwise known in this area as the Fitting semantics—because this semantics also interprets S4LP.

3.1 The Fitting semantics

A model in the Fitting semantics consists of an S4 Kripke model[2] (G, R, V) together with a certain mapping $\mathcal{E}$ from worlds and terms to sets of formulas, with the intent that $\mathcal{E}(\Gamma, t)$ is the set of formulas for which t serves as evidence at world Γ. For convenience, call a formula φ *knowable* at a world Γ if φ is true at all worlds accessible from Γ; that is, $\Gamma R \Delta$ implies φ is true at Δ. We then say that a formula of the form $t{:}\varphi$ is true at Γ if t is evidence for φ—that is, $\varphi \in \mathcal{E}(\Gamma, t)$—and φ is also knowable at Γ. Now for the details.

A *model* M is a tuple $(G, R, \mathcal{E}, V)$, where (G, R, V) is an S4 Kripke model and $\mathcal{E}$ is an evidence function. An *evidence function* is a map from worlds and terms to sets of formulas that satisfies each of the following properties:

[2] An S4 Kripke model is a triple (G, R, V), where G is a nonempty set whose elements are referred to as *worlds*, R is a reflexive and transitive binary relation on G, and V is a map from worlds to sets of propositional letters (so $V(\Gamma)$ is the set of propositional letters taken to be true at world Γ).

- *Evidence Closure*
  - *Application:* $\varphi \supset \psi \in \mathcal{E}(\Gamma, t)$ and $\varphi \in \mathcal{E}(\Gamma, s)$ implies $\psi \in \mathcal{E}(\Gamma, t \cdot s)$
  - *Proof Checker:* $\varphi \in \mathcal{E}(\Gamma, t)$ implies $t{:}\varphi \in \mathcal{E}(\Gamma, !t)$
  - *Sum:* $\mathcal{E}(\Gamma, t) \cup \mathcal{E}(\Gamma, s) \subseteq \mathcal{E}(\Gamma, t + s)$
  - *Constant Specification:* $A \in \mathcal{E}(\Gamma, c)$ for A an axiom and c a proof constant
- *Evidence Monotonicity:* $\Gamma R \Delta$ implies $\mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Delta, t)$ for every term t

Truth of a formula φ in M is then defined by induction on the complexity of φ, where the propositional cases are handled as is usual. A formula of the form $\Box\psi$ is said to be true at a world Γ in M whenever ψ is knowable at Γ. A formula of the form $t{:}\psi$ is said to be true at Γ whenever $\psi \in \mathcal{E}(\Gamma, t)$ and ψ is knowable at Γ. Notation: $M, \Gamma \models \varphi$ means that φ is true at Γ in M and $M, \Gamma \not\models \varphi$ means that φ is not true at Γ in M. In addition, $M \models \varphi$ means that $M, \Gamma \models \varphi$ for every Γ in M. As usual, a formula φ is said to be *valid* if $M \models \varphi$ for every model M.

3.2 Artemov's extension

In an $\mathsf{S4LP}$ model $(G, R, \mathcal{E}, V)$, the $\mathsf{S4}$ modality and the evidence terms both use the same relation R in their interpretation. Artemov observed in (Artemov 2006) that this need not be the case. In particular, $\mathsf{S4LP}$ may also be modeled by tuples $(G, R, R_e, \mathcal{E}, V)$, where R_e is a new reflexive and transitive binary relation on G satisfying $R \subseteq R_e$ (the other items of the tuple are as before). In these models, truth of formulas $t{:}\varphi$ is given by the relation R_e—as are the Evidence Closure and Evidence Monotonicity conditions—while truth of formulas $\Box\varphi$ is given by the relation R. Notice that the condition $R \subseteq R_e$ guarantees that such models satisfy the Connection Principle. The basic Fitting semantics of the previous subsection (Section 3.1) is obtained by taking $R = R_e$.

Artemov's extension allows a greater degree of flexibility in modeling. In particular, in such models the modality $\Box$ need not be an $\mathsf{S4}$ modality. Artemov's extension accordingly allows us to extend LP so that the extension incorporates various multi-modal logics with unary modalities $\Box_i$ (indexed by the subscript i) and corresponding interpreting relations R_i, each of which satisfies $R_i \subseteq R_e$. For example, $\mathsf{S5}_n\mathsf{LP}$ is a logic which has $\mathsf{S5}$ modalities $\Box_1, \ldots, \Box_n$, each of which satisfies the Connection Principle $t{:}\varphi \supset \Box_i\varphi$. Similarly, we have logics $\mathsf{T}_n\mathsf{LP}$, $\mathsf{S4}_n\mathsf{LP}$, and various mixed logics such as $\mathsf{S4S5LP}$ (where $\Box_1$ is $\mathsf{S4}$, $\Box_2$ is $\mathsf{S5}$, and each satisfies the Connection Principle).

4 The tableau systems

In a technical report (Renne 2004), the author defined a tableau system for LP.[3] This was extended by Fitting in (Fitting 2004) to $\mathsf{S4LP}$ and proved complete with respect to the class

[3]The author's LP tableau system is essentially a reformulation of Artemov's Gentzen-style system for LP (Artemov 2001).

of models with $R = R_e$, though Fitting's system is not cut-free. By a slight modification of Fitting's system, we obtain a system that is cut-free.

We first recall the author's LP tableau system and then show how it is extended to a cut-free system for $\mathsf{S4LP}$. We will then prove that the $\mathsf{S4LP}$ system is sound and complete with respect to the class of Fitting models where $R \subseteq R_e$. That $\mathsf{S4LP}$ is cut-free follows as a consequence of completeness, but more on this later.

4.1 The LP tableau system

A *tableau* for (or beginning with) φ is a tree with φ at the root constructed by non-deterministically applying a branch extension rule, called a *tableau rule*.[4] The tableau rules for LP are of three basic types: non-branching (otherwise known as α), branching (otherwise known as β), and ψ-branching (otherwise known as β^ψ, where ψ is an arbitrary LP formula). Each of these types will be described shortly. A branch of a tableau is said to be *closed* if it satisfies at least one of the following three conditions:

1. the branch contains both φ and $\neg\varphi$ for some formula φ,
2. the branch contains $\bot$, or
3. the branch contains $\neg(c{:}A)$ for c a proof constant and A an axiom.

If every branch of a tableau is closed, the tableau itself is said to be *closed.* A branch or tableau that is not closed is called *open.* A *tableau proof* of a formula φ is a closed tableau beginning with $\neg\varphi$. We now give the tableau rules for LP.

The classical tableau rules are given in Figure 1.1. In this figure, the leftmost rule is an α. For convenience, we follow Smullyan's naming convention of (Smullyan 1963): the formula above the line is referred to as α and the formulas below the line are called α_1 (the top formula) and α_2. An α rule allows any branch on which α appears to be extended by adding either α_1 or α_2 to the end of the branch.

$$\frac{\neg(\varphi \supset \psi)}{\begin{array}{c}\varphi\\ \neg\psi\end{array}} \qquad \frac{\neg\neg\varphi}{\varphi} \qquad \frac{\varphi \supset \psi}{\neg\varphi \mid \psi}$$

Figure 1.1: Tableau rules for classical logic.

The middle rule in Figure 1.1 is an α rule that only produces one formula α_1, so we will thus adopt the convention that α_1 and α_2 name the same formula in such a situation.

[4] It will also be convenient to allow for the case that a tableau is for (or begins with) a set of formulas. In this case, the tableau is constructed from the single-branched tree consisting of that set of formulas (appearing in any order).

The rightmost rule in Figure 1.1 is a β. Following a similar naming convention as in the α case, the formula above the (horizontal) line is referred to as β and the formulas below the line are called β_1 (the left formula) and β_2. A β rule allows any branch on which β appears to be extended by splitting at the end so that both β_1 and β_2 are new leaves.

Following the diagrammatic conventions suggested in Figure 1.1 for designating α rules, the tableau rules for the LP evidence operations are given in Figure 1.2.

$$\frac{t\!:\!\varphi}{\varphi} \qquad \frac{\neg(!t\!:\!(t\!:\!\varphi))}{\neg(t\!:\!\varphi)} \qquad \frac{\neg((s+t)\!:\!\varphi)}{\begin{array}{c}\neg(s\!:\!\varphi)\\ \neg(t\!:\!\varphi)\end{array}} \qquad \frac{\neg((s\cdot t)\!:\!\varphi)}{\neg(s\!:\!(\psi\supset\varphi)) \;\big|\; \neg(t\!:\!\psi)}$$

Figure 1.2: Tableau rules for the LP evidence operations.

The rule at the far right in Figure 1.2 is rather odd, since the formula ψ is arbitrary and, in particular, need not even appear as a subformula of $\neg((s\cdot t)\!:\!\varphi)$. So, while this rule has the diagrammatic form of a β rule, a formula ψ must be given as a parameter in order to specify the formulas below the line. This rule is thus called a β^ψ rule, where ψ may be any formula, and the formulas below the line are then called β_1^ψ (the left formula) and β_2^ψ. As is the case for β formulas, a branch on which a β^ψ formula occurs may be extended by splitting at the end so that both β_1^ψ and β_2^ψ are new leaves.

This completes the specification of the tableau system for LP. This system notably omits the cut rule, which is given in Figure 1.3. For any formula φ, the cut rule allows any branch to be extended by splitting at the end so that both φ and $\neg\varphi$ are new leaves. While the LP tableau system is cut-free (that is, without cut), the system does not have the subformula property due to the presence of the β^ψ rule.[5]

$$\frac{}{\varphi \;\big|\; \neg\varphi}$$

Figure 1.3: The (tableau) cut rule. This rule is not a part of the LP tableau system.

4.2 The S4LP tableau system

The tableau system for S4LP is obtained from the system for LP by adding a rule corresponding to the Connection Principle along with rules to handle formulas of the form $\Box\varphi$ and of the form $\neg\Box\varphi$. These rules are given in Figure 1.4. In this figure, the rightmost rule is a new rule type, known as a *destructive rule*. A destructive rule modifies the tableau,

[5]The *subformula property* is the property whereby each of the formulas below the line in a rule diagram is a subformula of the formula above the line.

in this case deleting a branch S containing the formula $\neg\Box\varphi$ and adding to the tableau a new branch consisting of those formulas in $S^{\#}$ along with the formula $\neg\varphi$. This deletion operation can be implemented in various ways. A simple implementation is to mark all formulas on the branch (including $\neg\Box\varphi$) as "deleted" for this branch and then extend the branch by adding $\neg\varphi$ and each of the formulas in $S^{\#}$ at the branch end (in any order).[6] Tableau rules are then restricted so that a tableau rule may be applied to a formula ψ appearing on a branch θ only if ψ is not marked as "deleted" for θ. And, as the reader might expect, a branch θ is closed only if one of the closure conditions applies to formulas not marked as "deleted" for θ.

$$\frac{t\!:\!\varphi}{\Box\varphi} \qquad \frac{\Box\varphi}{\varphi} \qquad \frac{S, \neg\Box\varphi}{S^{\#}, \neg\varphi}$$

$$S^{\#} := \{\Box\psi \mid \Box\psi \in S\} \cup \{t\!:\!\psi \mid t\!:\!\psi \in S\}$$

Figure 1.4: Additional tableau rules for S4LP.

5 Correctness of the tableau systems

We have presented two tableau systems, one for LP and another for S4LP. We will show the correctness results (soundness and completeness) for the latter system with respect to the class of models satisfying $R \subseteq R_e$. We indicate later how these correctness results may be modified to handle the LP system. So let us proceed with the proof of correctness of the S4LP system.

The proof of soundness is facilitated by a couple of definitions and a standard lemma. In particular, an open branch θ on a tableau is said to be *satisfiable* if there is a world Γ of a model M such that $M, \Gamma \models \varphi$ for every $\varphi \in \theta$ that is not "deleted" for θ. A tableau is then said to be *satisfiable* if it has a satisfiable open branch. The following lemma, whose proof is standard and is thus omitted, then leads easily to the proof of soundness.

Lemma 1. If τ is a satisfiable tableau, any tableau produced from τ by application of a tableau rule is also satisfiable.

Theorem 1 (Soundness)**.** If a formula φ has a tableau proof, then φ is valid.

Proof. If φ is not valid, there is a world Γ of a model M such that $M, \Gamma \not\models \varphi$. Thus $M, \Gamma \models \neg\varphi$ and so $\neg\varphi$ is satisfiable. By Lemma 1, no tableau beginning with $\neg\varphi$ can close, so φ does not have a tableau proof. □

[6]This is Fitting's approach in (Fitting 1999).

Completeness of the tableau system uses a canonical model construction, with maximal consistency defined relative to the tableau system. Care is taken to avoid implicit use of the cut rule (which is not present in the system), but more on this later.

Since all of the sets mentioned in the remainder of the paper are sets of formulas, a *set* is assumed to be a set of formulas. Now let S be a set. S is said to be *consistent* if for no finite subset S' does a tableau beginning with S' close. If S is not consistent, it is called *inconsistent*. A consistent S is then *maximal consistent* if adding any formula to S that is not already present produces an inconsistent set. It is a well-known fact that every consistent set can be extended to a maximal consistent set. What's more, any maximal consistent set S satisfies each of the following properties:

- $\alpha \in S$ implies both $\alpha_1 \in S$ and $\alpha_2 \in S$,
- $\beta \in S$ implies $\beta_1 \in S$ or $\beta_2 \in S$, and
- $\beta^\psi \in S$ implies β_1^ψ or $\beta_2^\psi \in S$.

A set satisfying each of these properties is called *downward saturated*, so every maximal consistent set is downward saturated.

We now begin the proof of completeness. After constructing the canonical model (and verifying that it is in fact a model satisfying $R \subseteq R_e$), we verify a useful lemma known as the Truth Lemma. Completeness of the tableau system follows almost immediately from the Truth Lemma. Proceeding, we construct the *canonical model* $M = (G, R, R_e, \mathcal{E}, V)$ as follows:

- G is the collection of all maximal consistent sets
- $\Gamma R_e \Delta$ holds if and only if both $\{t{:}\psi \mid t{:}\psi \in \Gamma\} \subseteq \Delta$ and $\{\neg t{:}\psi \mid \neg t{:}\psi \in \Delta\} \subseteq \Gamma$
- $\Gamma R \Delta$ holds if and only if both $\Gamma R_e \Delta$ and $\{\Box\psi \mid \Box\psi \in \Gamma\} \subseteq \Delta$
- $\varphi \in \mathcal{E}(\Gamma, t)$ holds if and only if $\neg t{:}\varphi \notin \Gamma$
- $p \in V(\Gamma)$ holds if and only if $p \in \Gamma$

Lemma 2. The canonical model is a model.

Proof. It's clear that R and R_e are transitive and reflexive and G is nonempty, so (G, R, V) is an $\mathsf{S4}$ Kripke model. It is also obvious that $R \subseteq R_e$. What remains is to show that $\mathcal{E}$ is an evidence function; that is, $\mathcal{E}$ satisfies Evidence Closure and Evidence Monotonicity.

The Evidence Closure conditions follow from the tableau rules and the fact that each world is maximal consistent. As an example, we check Application. So suppose that $\varphi \supset \psi \in \mathcal{E}(\Gamma, t)$ and that $\varphi \in \mathcal{E}(\Gamma, s)$. We then have $\neg t{:}(\varphi \supset \psi) \notin \Gamma$ and $\neg s{:}\varphi \notin \Gamma$ by the definition of $\mathcal{E}$. Note that $\neg t{:}(\varphi \supset \psi)$ has the form of a β_1^φ and that $\neg s{:}\varphi$ has the form of a β_2^φ. Now, since Γ is downward saturated, it cannot be the case that $\neg(t \cdot s){:}\psi \in \Gamma$, for otherwise the β^φ rule applies and $\beta_1^\varphi \in \Gamma$ or $\beta_2^\varphi \in \Gamma$, a contradiction. Hence $\neg(t \cdot s){:}\psi \notin \Gamma$ and thus $\psi \in \mathcal{E}(\Gamma, t \cdot s)$ by the definition of $\mathcal{E}$.

We now show $\mathcal{E}$ satisfies Evidence Monotonicity. So assume $\Gamma R_e \Delta$ and $\varphi \in \mathcal{E}(\Gamma, t)$. From the definition of $\mathcal{E}$, we have $\neg t{:}\varphi \notin \Gamma$. It then follows from the meaning of $\Gamma R_e \Delta$ that $\neg t{:}\varphi \notin \Delta$, and thus $\varphi \in \mathcal{E}(\Delta, t)$, as desired. □

Lemma 3 (Truth Lemma)**.** Let $M = (G, R, R_e, \mathcal{E}, V)$ be the canonical model and Γ be a world of M. Then each of the following holds:

- $\varphi \in \Gamma$ implies $M, \Gamma \models \varphi$, and
- $\neg\varphi \in \Gamma$ implies $M, \Gamma \not\models \varphi$.

Proof. By induction on φ. The base cases and propositional inductive case are standard, so we restrict our attention to the other inductive cases. We use without mention the fact that worlds of M are downward saturated (and thus closed under α-rule applications).

- Case $t{:}\varphi \in \Gamma$.

  Since Γ is consistent, $\neg t{:}\varphi \notin \Gamma$ and thus $\varphi \in \mathcal{E}(\Gamma, t)$. Now let Δ be an arbitrary world satisfying $\Gamma R_e \Delta$. By the definition of R_e, we have $t{:}\varphi \in \Delta$ and thus $\varphi \in \Delta$ by an α rule. By the induction hypothesis, we have $M, \Delta \models \varphi$. Since Δ was arbitrary, we have shown that φ is knowable at Γ.

- Case $\neg t{:}\varphi \in \Gamma$.

  By the definition of $\mathcal{E}$, we have $\varphi \notin \mathcal{E}(\Gamma, t)$ and thus $M, \Gamma \not\models t{:}\varphi$.

- Case $\Box\varphi \in \Gamma$.

  Let Δ be an arbitrary world satisfying $\Gamma R \Delta$. By the definition of R, we have $\Box\varphi \in \Delta$ and thus $\varphi \in \Delta$ by an α rule. By the induction hypothesis, we have $M, \Delta \models \varphi$. Since Δ was arbitrary, we have shown that φ is knowable at Γ.

- Case $\neg\Box\varphi \in \Gamma$.

  By the destructive tableau rule, we have that $\Gamma^{\#} \cup \{\neg\varphi\}$ is consistent. We may thus extend this union to a maximal consistent set Δ and we then have $\Delta \in G$. Since $\neg\varphi \in \Delta$, it follows from the induction hypothesis that $M, \Delta \not\models \varphi$. We have shown that φ is not knowable at Γ. □

Theorem 2 (Completeness)**.** A valid formula φ has a tableau proof.

Proof. Suppose φ does not have a tableau proof, so $\{\neg\varphi\}$ is consistent and may thus be extended to a maximal consistent set Γ. This set Γ is a world of the canonical model M and we thus have $M, \Gamma \not\models \varphi$ by the Truth Lemma. □

In (Artemov 2006), it is shown that the Hilbert-style theory of S4LP presented in Section 2 is sound and complete with respect to the class of Fitting models satisfying $R \subseteq R_e$. Since the same is true of the tableau system (with respect to the same semantics), the tableau system describes the same theory as do the Hilbert-style axiom and rule schemas.

Call a set S *downward closed* if it is downward saturated and, in addition, S contains either φ or its negation $\neg\varphi$ for every formula φ. In a tableau system with cut, every maximal consistent set S is downward closed. In such a system with cut, it can be shown that for any world Γ of the canonical model M, we have $\varphi \in \Gamma$ if and only if $M, \Gamma \models \varphi$. The Truth Lemma may thus take a sharper form and is a bit easier to prove. However, the added difficulty of working in a cut-free tableau system provides us with an additional payoff: not only do we obtain completeness but we also verify the admissibility of cut in the S4LP tableau system.

Theorem 3. Cut is an admissible rule.

Proof. Cut is certainly a sound rule. Since the S4LP tableau system without cut is complete, every formula provable in the S4LP tableau system with cut is also provable without. But this is what it means to say that cut is an admissible rule. □

That the tableau system for LP is correct follows by an appropriate restriction of the above completeness argument (soundness is identical).

Theorem 4 (Completeness)**.** A valid formula φ in the language of LP has an LP tableau proof.

Proof. Construct the canonical model as above, except set $R = R_e$ (and thus omit the condition $\{\Box\psi \mid \Box\psi \in \Gamma\} \subseteq \Delta$). The verification that this canonical model is a model is as before. In the proof of the Truth Lemma, the cases $\Box\varphi$ and $\neg\Box\varphi$ are omitted. Completeness is as before. □

Theorem 5. Cut is an admissible rule in the LP tableau system.

Proof. As in the S4LP case. □

In (Artemov 2001), Artemov proved Theorem 5 via syntactic means. This has the added advantage of providing a procedure for converting proofs with cut to those without.

6 Conclusion

While our tableau systems are cut-free, they do not have the subformula property. It is unknown at present whether there is a cut-free tableau system for LP (or for S4LP) that does have this property. The troublemaker, of course, is the β^{ψ} rule. Since LP is a decidable language, perhaps there is an efficient procedure for computing ψ, which would give the system a sort of computable subformula property. This might be a reasonable compromise if we otherwise wish to use the system we presented above.

7 Acknowledgments

The author extends his thanks to two anonymous referees, whose comments improved the paper greatly. The author is also grateful for Melvin Fitting's tireless editing of previous versions of this paper. Evan Goris also provided useful feedback and assistance on an earlier version. Additionally, the author thanks Sergei Artemov for his support, advice, and teaching.

Bibliography

Alberucci, L. and G. Jäger (2005). About cut elimination for logics of common knowledge. *Annals of Pure and Applied Logic 133*, 73–99.

Artemov, S. N. (2001). Explicit provability and constructive semantics. *The Bulletin of Symbolic Logic 7*(1), 1–36.

Artemov, S. N. (2004). Evidence-based common knowledge. Technical Report TR-2004018, CUNY Graduate Center Ph.D. Program in Computer Science.

Artemov, S. N. (2006). Justified common knowledge. *Theoretical Computer Science*. Article in Press, doi:10.1016/j.tcs.2006.03.009.

Fitting, M. (1999). A simple propostional S5 tableau system. *Annals of Pure and Applied Logic 96*, 107–115.

Fitting, M. (2003). A semantics for the Logic of Proofs. Technical Report TR-2003012, CUNY Graduate Center Ph.D. Program in Computer Science.

Fitting, M. (2004). Semantics and tableaus for LPS4. Technical Report TR-2004016, CUNY Graduate Center Ph.D. Program in Computer Science.

Fitting, M. (2005). The logic of proofs, semantically. *Annals of Pure and Applied Logic 132*(1), 1–25.

Mkrtychev, A. (1997). Models for the logic of proofs. In S. I. Adian and A. Nerode (Eds.), *Logical Foundations of Computer Science, Proceedings of the 4th International Symposium*, Volume 1234 of *Lecture Notes in Computer Science*, pp. 266–275. Springer.

Renne, B. (2004). Tableaux for the logic of proofs. Technical Report TR-2004001, CUNY Graduate Center Ph.D. Program in Computer Science.

Renne, B. (2006). Propositional games with explicit strategies. In *Proceedings of the 13th Workshop on Logic, Language, and Computation (WoLLIC)*.

Smullyan, R. M. (1963). A unifying principal in quantification theory. In *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, Volume 49, pp. 828–832.