Certification of prefixed tableau proofs for modal logic

Tomer Libal and Marco Volpe

INRIA, Parsifal Team
Can we trust provers?

Complex software is rarely free of bugs.
Automated theorem provers are complex software - can we trust them?
Current provers can rarely share each other’s proofs

Work has been done for building bridges between two specific provers (but even a change in the version number of one prover can cause that bridge to collapse)
Towards proof certification

Motivating questions

1. Can we trust provers?
2. Can provers talk a common language?

Goal

Provide a flexible framework for defining the semantics of a wide range of proof evidences in such a way that:

- provers would define the meaning of their own proof evidence;
- trusted proof checkers would be able to interpret that meaning and check its formal correctness.
An analogy

**Structural Operational Semantics**

1. There are many programming languages.
2. SOS can define the semantics of many of them.
3. Compilers can be built based on the semantics.

**Foundational Proof Certificates (FPCs)**

1. There are many forms of proof evidence.
2. FPC can define the semantics of many of them.
3. Checkers can be built based on the semantics.
- **Proof evidence**: The proof output from a prover.
- **Pretty printer**: Some program for properly formatting the proof evidence.
- **FPC specification**: Specification of predicates used to interpret the proof evidence in order to guide the kernel proof search.
- **Kernel**: A trustable low-level calculus, with additional control predicates.
- **Proof evidence**: The proof output from a prover.
- **Pretty printer**: Some (typically OCaml) program for properly formatting the proof evidence (as a λProlog file).
- **FPC specification**: λProlog specification of predicates used to interpret the proof evidence in order to guide the kernel proof search.
- **Kernel**: An encoding of (focused) sequent calculus (LKF + control predicates) as a λProlog program.
Let's consider a **sequent calculus** for classical first-order logic (LK).

- Reduce the **search space**.
- Better organize the **structure** of derivations.
- Emphasis on: non-invertible vs. invertible rules.
- Propositional connectives have:
  - a positive version;
  - a negative version.

\[
\frac{\Gamma, \Theta, B_i}{\Gamma, \Theta, B_1 \lor B_2} \quad \frac{\Gamma, \Theta, B_1, B_2}{\Gamma, \Theta, B_1 \lor B_2}
\]

\[\vdash_{F} \quad \vdash_{F}
\]
Let’s consider a **sequent calculus** for classical first-order logic (LK).

- Reduce the **search space**.
- Better organize the **structure** of derivations.
- Emphasis on: **non-invertible** vs. **invertible** rules.
- Propositional connectives have:
  - a **positive** version;
  - a **negative** version.
- Polarization of a formula does not affect its **provability**.
Focused proof systems

store  (a positive formula to possibly focus on later)

\[ \Gamma \vdash \Theta \uparrow \Gamma \quad t^-, f^-, \forall^-, \land^-, \forall \]

release

\[ \Gamma \vdash \Theta \downarrow \Delta \quad t^+, f^+, \forall^+, \land^+, \exists \]

decide  (on a positive formula to focus on)
Focused proof systems

store  (a positive formula to possibly focus on later)

$\vdash \Theta \uparrow \Gamma$  NEGATIVE PHASE (invertible)

release  (change of phase)

$\vdash \Theta \downarrow A$  POSITIVE PHASE (non-invertible)

decide  (on a positive formula to focus on)
Focused proof systems

\( \vdash \Theta \uparrow \Gamma \)  (a positive formula to possibly focus on later)

\( \vdash \Theta \downarrow A \)  (on a positive formula to focus on later)

By the way,

release

this is a BIPOLE

\( t^-, f^-, \lor-, \land-, \forall \)

\( t^+, f^+, \lor^+, \land^+, \exists \)

decide
A focused proof system for classical logic (LKF)

**Negative introduction rules**

\[
\frac{\vdash \Theta \uparrow t^{-}, \Gamma}{\vdash \Theta \uparrow t^{-}} \\
\frac{\vdash \Theta ^{\uparrow} A, \Gamma}{\vdash \Theta ^{\uparrow} A \wedge^{-} B, \Gamma} \\
\frac{\vdash \Theta ^{\uparrow} \Gamma}{\vdash \Theta ^{\uparrow} f^{-}, \Gamma} \\
\frac{\vdash \Theta \uparrow A, B, \Gamma}{\vdash \Theta \uparrow A \vee^{-} B, \Gamma}
\]

\[
\frac{\vdash \Theta \uparrow [y/x]B, \Gamma}{\vdash \Theta \uparrow \forall x.B, \Gamma}
\]

**Positive introduction rules**

\[
\frac{\vdash \Theta \downarrow t^{+}, \Gamma}{\vdash \Theta \downarrow t^{+}} \\
\frac{\vdash \Theta \downarrow B_1, \Theta \downarrow B_2}{\vdash \Theta \downarrow B_1 \wedge^{+} B_2} \\
\frac{\vdash \Theta \downarrow B_i}{\vdash \Theta \downarrow B_1 \vee^{+} B_2} \\
\frac{\vdash \Theta \downarrow B_i, \Theta \downarrow B_j}{\vdash \Theta \downarrow B_i \vee^{+}, i \in \{1, 2\}} \\
\frac{\vdash \Theta \downarrow [t/x]B}{\vdash \Theta \downarrow \exists x.B}
\]

**Identity rules**

\[
\vdash \neg P_a, \Theta \downarrow P_a
\]

\[
\frac{\vdash \Theta \uparrow B}{\vdash \Theta \uparrow B} \\
\frac{\vdash \Theta \uparrow \neg B}{\vdash \Theta \uparrow \neg B}
\]

\[
\text{init}
\]

\[
\text{cut}
\]

**Structural rules**

\[
\frac{\vdash \Theta, C \uparrow \Gamma}{\vdash \Theta \uparrow C, \Gamma} \\
\frac{\vdash \Theta \uparrow N}{\vdash \Theta \downarrow N}
\]

\[
\text{store}
\]

\[
\frac{\vdash P, \Theta \downarrow P}{\vdash P, \Theta \uparrow P}
\]

\[
\text{release}
\]

\[
\text{decide}
\]

T. Libal, M. Volpe  Certification of prefixed tableau proofs for modal logic
**Labeled deduction** approach: we encode in the syntax additional information (e.g. of a semantic nature).

**Labels denoting worlds**

- Two classes of **formulas**:
  
  1. Labeled logical formulas, e.g. $x : A$
  
  2. Relational formulas, e.g. $xRy$

- The **basic idea** is:
  
  - each label $y$ refers to a world $\bar{y}$ in the Kripke semantics
  
  - the relational symbol $R$ refers to the accessibility relation
A labeled sequent system for modal logic

**Classical rules**

\[
\frac{x : P, \Gamma \vdash \Delta}{x : P, \Gamma \vdash \Delta, x : P} \quad \text{init} \quad \frac{x : A, x : B, \Gamma \vdash \Delta}{x : A \land B, \Gamma \vdash \Delta} \quad L\land \quad \frac{\Gamma \vdash \Delta, x : A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, x : A \land B} \quad R\land
\]

\[
\frac{x : A, \Gamma \vdash \Delta}{x : A \lor B, \Gamma \vdash \Delta} \quad L\lor \quad \frac{\Gamma \vdash \Delta, x : A, x : B}{\Gamma \vdash \Delta, x : A \lor B} \quad R\lor
\]

**Modal rules**

\[
\frac{y : A, x : \Box A, xRy, \Gamma \vdash \Delta}{x : \Box A, xRy, \Gamma \vdash \Delta} \quad L\Box \quad \frac{xRy, \Gamma \vdash \Delta, y : A}{\Gamma \vdash \Delta, x : \Box A} \quad R\Box
\]

\[
\frac{xRy, y : A, \Gamma \vdash \Delta}{x : \Diamond A, \Gamma \vdash \Delta} \quad L\Diamond \quad \frac{xRy, \Gamma \vdash \Delta, x : \Diamond A, y : A}{xRy, \Gamma \vdash \Delta, x : \Diamond A} \quad R\Diamond
\]

In \( R\Box \) and \( L\Diamond \), \( y \) does not occur in the conclusion.
A prefixed tableau system for modal logic

**Classical rules**

\[
\frac{\sigma : A \land B}{\sigma : A, \sigma : B} \quad \land_F \\
\frac{\sigma : A \lor B}{\sigma : A \mid \sigma : B} \quad \lor_F
\]

**Modal rules**

\[
\frac{\sigma : \Box A}{\sigma.n : A} \quad \Box_F \\
\frac{\sigma : \Diamond A}{\sigma.n : A} \quad \Diamond_F
\]

In \( \Box_F \), \( \sigma.n \) is used. In \( \Diamond_F \), \( \sigma.n \) is new.

Plus branch **closure rules**, of course.
Labeling and focusing

**PROPOSITIONAL MODAL LOGIC** → **STANDARD TRANSLATION** → **FIRST-ORDER CLASSICAL LOGIC**

**LABELED PROOF SYSTEM**

**FOCUSED PROOF SYSTEM**
Labeling and focusing

PROPOSITIONAL MODAL LOGIC → STANDARD TRANSLATION → FIRST-ORDER CLASSICAL LOGIC

PROPOSITIONAL MODAL LOGIC

STANDARD TRANSLATION

FIRST-ORDER CLASSICAL LOGIC

LABELED PROOF SYSTEM → LESS STANDARD TRANSLATION → FOCUSED PROOF SYSTEM

T. Libal, M. Volpe  
Certification of prefixed tableau proofs for modal logic
Labeling and focusing

PROPOSITIONAL MODAL LOGIC \[\rightarrow\] STANDARD TRANSLATION \[\rightarrow\] FIRST-ORDER CLASSICAL LOGIC

LABELED PROOF SYSTEM \[\downarrow\] LESS STANDARD TRANSLATION \[\downarrow\] FOCUSED PROOF SYSTEM

inference rule \[\rightarrow\] bipole
The standard translation

**Modal language**  \(\Rightarrow\) **FO language** with:
- a binary predicate \(R\)
- a unary predicate \(P\) for each \(P \in \mathcal{P}\)

\[
\begin{align*}
ST_x(P) &= P(x) \\
ST_x(A \land B) &= ST_x(A) \land ST_x(B) \\
ST_x(\Box A) &= \forall y(\neg R(x, y) \lor ST_y(A)) \\
ST_x(\Diamond A) &= \exists y(R(x, y) \land ST_y(A))
\end{align*}
\]

where \(x\) is a free variable.

For any modal formula \(A\), any model \(\mathcal{M}\) and any world \(w\):

\[\mathcal{M}, w \models A\] iff \[\mathcal{M} \models ST_x(A)[x \leftarrow w]\]
### Our translation \[ ... \]

\[
\begin{align*}
ST_x(P) & = P(x) \\
[x : P] & = P(x) \\

ST_x(A \land B) & = ST_x(A) \land ST_x(B) \\
[x : A \land B] & = \partial^+ ([x : A]) \land \lnot \partial^+ ([x : B]) \\

ST_x(\Box A) & = \forall y (\lnot R(x, y) \lor ST_y(A)) \\
[x : \Box A] & = \forall y (\lnot R(x, y) \lor \lnot \partial^+ ([y : A])) \\

ST_x(\Diamond A) & = \exists y (R(x, y) \land ST_y(A)) \\
[x : \Diamond A] & = \exists y (R(x, y) \land \partial^+ \partial^- (\partial^+ ([y : A])))
\end{align*}
\]

Delay operators \((\partial^+, \partial^-)\) force a formula to be **positive** or **negative**.
Theorem of adequacy

PROPOSITIONAL MODAL LOGIC \quad \text{STANDARD TRANSLATION} \quad \text{FIRST-ORDER CLASSICAL LOGIC}

Labeled Proof System

LESS STANDARD TRANSLATION

inference rule

bipole

T. Libal, M. Volpe Certification of prefixed tableau proofs for modal logic
Typically, in an FPC specification, the information about $t$ will be contained in $\Xi$.

- e.g., $\Xi = \{t, t_1, \ldots, t_n\}$ and $\Xi' = \{t_1, \ldots, t_n\}$. 
The augmented focused system \( LKF^a \)

**Invertible Rules**

\[
\begin{align*}
\Xi' \vdash \Theta \uparrow A, \Gamma & \quad \Xi'' \vdash \Theta \uparrow B, \Gamma \quad \text{andNeg}_c(\Xi, \Xi', \Xi'') \\
\Xi \vdash \Theta \uparrow A \wedge \neg B, \Gamma \\
\Xi' \vdash \Theta \uparrow A, B, \Gamma & \quad \text{orNeg}_c(\Xi, \Xi') \\
\Xi \vdash \Theta \uparrow A \vee \neg B, \Gamma \\
(\Xi'y) \vdash \Theta \uparrow [y/x]B, \Gamma & \quad \text{all}_c(\Xi, \Xi') \\
\Xi \vdash \Theta \uparrow \forall x.B, \Gamma
\end{align*}
\]

**Focused Rules**

\[
\begin{align*}
\Xi' \vdash \Theta \downarrow B_1 & \quad \Xi'' \vdash \Theta \downarrow B_2 \quad \text{andPos}_e(\Xi, \Xi', \Xi'') \\
\Xi \vdash \Theta \downarrow B_1 \wedge^+ B_2 \\
\Xi' \vdash \Theta \downarrow B_i & \quad \text{orPos}_e(\Xi, \Xi', i) \\
\Xi \vdash \Theta \downarrow B_1 \vee^+ B_2 \\
\Xi' \vdash \Theta \downarrow [t/x]B & \quad \exists t.e(\Xi, t, \Xi') \\
\Xi \vdash \Theta \downarrow \exists x.B
\end{align*}
\]

**Identity rules**

\[
\begin{align*}
\Xi' \vdash \Theta \uparrow B & \quad \Xi'' \vdash \Theta \uparrow \neg B \quad \text{cut}_e(\Xi, \Xi', \Xi'', B) \\
\Xi \vdash \Theta \uparrow \cdot
\end{align*}
\]

**Structural rules**

\[
\begin{align*}
\Xi' \vdash \Theta \uparrow N & \quad \text{release}_e(\Xi, \Xi') \\
\Xi \vdash \Theta \downarrow N \\
\Xi' \vdash \Theta \downarrow P & \quad \langle l, P \rangle \in \Theta \quad \text{decide}_e(\Xi, l, \Xi') \\
\Xi \vdash \Theta \uparrow \cdot
\end{align*}
\]

T. Libal, M. Volpe  
Certification of prefixed tableau proofs for modal logic
A proof is punctually represented by specifying:

1. at each step, on which formula we apply a rule (\textit{decide}-predicate);
2. in the case of a ♦-formula, with respect to which label (∃-predicate);
3. in the case of an initial, with respect to which complementary literal (\textit{init}-predicate).

This gives rise to a punctual FPC specification:

- it allows for reconstructing the proof in a very faithful way;
- it might be not very concise.
We can only require some essential information:

1. a mapping between □- and ◊- formulas (∃-predicate);
2. a mapping between complementary literals (init-predicate).

This gives rise to an essential FPC specification:

- it leaves the checker free of doing some not-driven reconstruction;
- it is less faithful but also more concise.
The essential specification can be used also to check proofs in, e.g., free variable (FV) tableaux (where punctual is not possible).

\[
\begin{array}{ccc}
1 : \Box \neg p \lor \Box \neg q & & 1 : \Box \neg p \lor \Box \neg q \\
1 : \Box(p \land q) & & 1 : \Box(p \land q) \\
1 : \Box \neg p & & 1 : \Box \neg q \\
1.1 : \neg p & & 1.2 : \neg q \\
1.1 : p \land q & & 1.2 : p \land q \\
1.1 : p & & 1.2 : p \\
1.1 : q & & 1.2 : q \\
\end{array}
\]
<table>
<thead>
<tr>
<th>Formalism</th>
<th>Prover</th>
<th>Punctual FPC</th>
<th>Essential FPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labeled sequents</td>
<td>by hand</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prefixed tableaux</td>
<td>ModLEAN-TAP</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FV-tableaux</td>
<td>ModLEAN-TAP</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>
In this paper

- Application of the use of a general framework for proof checking/certification to modal logics.
- Two different specifications for prefixed tableau proofs.

Current and future work

- Extension to modal logics represented by geometric frame properties.
- Extension to other formalisms:
  - “unlabeled” sequent systems;
  - nested sequent systems;
  - hypersequent systems;
  - resolution methods.
Thank you!