Isabelle Proof Assistant and Hybrid Logic

Formalizing Seligman-Style Tableaux

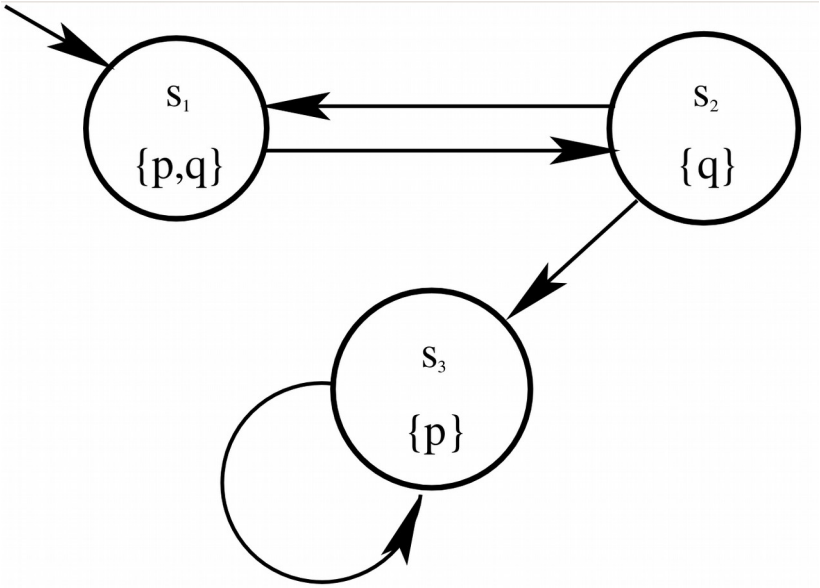
Andreas Halkjær From, DTU Compute

Background

- Masters thesis: Hybrid Logic
- Current plan: Formalize in Isabelle/HOL the paper
  - Klaus Frovin Jørgensen, Patrick Blackburn, Thomas Bolander and Torben Braüner. Synthetic Completeness Proofs for Seligman-style Tableau Systems. Advances in Modal Logic 11:302-321 2016.
  - [Patrick Blackburn, Thomas Bolander, Torben Braüner and Klaus Frovin Jørgensen. Completeness and Termination for a Seligman-style Tableau System. Journal of Logic and Computation 27(1):81-107 2017.]
- Dates: 19/08 2019 – 19/01 2020
- Supervisors:
  - Jørgen Villadsen
  - Alexander Birch Jensen
  - Patrick Blackburn

Modal Logic

- Translates into a (decidable) fragment of classical (first-order) logic
- We think of models as graphs / relational structures
- *modalities are simply macros that handle quantification over accessible states*
- Local perspective
  (we are inside the graph)



Hybrid Logic

- Still translates into a (decidable) fragment of first-order logic
- Adds nominals that stand for specific states
- If nominal *i* stands for Wednesday noon and propositional symbol *p* stands for ”it is raining” we can now say *@i p*, ”*at* Wednesday noon it is raining”.
- Nominals along with the satisfaction operator @ can witness diamonds based on accessibility:

  $\Diamond i$ and *@i p* implies $\Diamond p$

Isabelle

- Isabelle is a generic proof assistant.
- It allows mathematical formulas to be expressed in a formal language and provides tools for proving those formulas in a logical calculus.
- The main application is the formalization of mathematical proofs and in particular formal verification, which includes proving the correctness of computer hardware or software and proving properties of computer languages and protocols.

http://isabelle.in.tum.de/overview.html

- That is, **machine-checked proofs**.

Syntax

- Encode the syntax as a datatype.
- Automatically generates induction principle, disjointness lemmas and more.

$$\varphi ::= i \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi \mid @_i\varphi.$$

```
datatype ('a, 'b) fm
  = Pro 'a
  | Nom 'b
  | Neg ‹('a, 'b) fm› ("¬ _" [40] 40)
  | Dis ‹('a, 'b) fm› ‹('a, 'b) fm› (infixr "∨" 30)
  | Dia ‹('a, 'b) fm› ("◇ _" 10)
  | Sat 'b ‹('a, 'b) fm› ("@ _ _" 10)
```

Semantics I

- ”We interpret the language in models based on frames (W, R), where W is a non-empty set (we call its elements worlds) and R is a binary relation on W (the accessibility relation).”
- ”A model is a triple (W, R, V) where (W, R) is a frame and V (the valuation) maps propositional symbols p to arbitrary subsets of W, and nominals i to singleton subsets of W.”

$$
\begin{aligned}
\mathfrak{M}, w &\models a && \text{iff } a \text{ is atomic and } w \in V(a)\\
\mathfrak{M}, w &\models \neg\varphi && \text{iff } \mathfrak{M}, w \not\models \varphi\\
\mathfrak{M}, w &\models \varphi \vee \psi && \text{iff } \mathfrak{M}, w \models \varphi \text{ or } \mathfrak{M}, w \models \psi\\
\mathfrak{M}, w &\models \Diamond\varphi && \text{iff for some } w',\ wRw' \text{ and } \mathfrak{M}, w' \models \varphi\\
\mathfrak{M}, w &\models @_i\varphi && \text{iff } \mathfrak{M}, w' \models \varphi \text{ and } w' \in V(i).
\end{aligned}
$$

Semantics II

- 'w is the non-empty type of worlds.
- R is the accessibility relation, V is the valuation on propositions, g maps nominals to worlds.

```
datatype ('w, 'a) model =
  Model (R: ‹'w ⇒ 'w set›) (V: ‹'w ⇒ 'a ⇒ bool›)

primrec semantics
  :: ‹('w, 'a) model ⇒ ('b ⇒ 'w) ⇒ 'w ⇒ ('a, 'b) fm ⇒ bool›
  ("_, _, _ ⊨ _" [50, 50, 50] 50) where
  ‹(M, _, w ⊨ Pro x) = V M w x›
| ‹(_, g, w ⊨ Nom i) = (w = g i)›
| ‹(M, g, w ⊨ ¬ p) = (¬ M, g, w ⊨ p)›
| ‹(M, g, w ⊨ (p ∨ q)) = ((M, g, w ⊨ p) ∨ (M, g, w ⊨ q))›
| ‹(M, g, w ⊨ ◇ p) = (∃v ∈ R M w. M, g, v ⊨ p)›
| ‹(M, g, _ ⊨ @ i p) = (M, g, g i ⊨ p)›
```

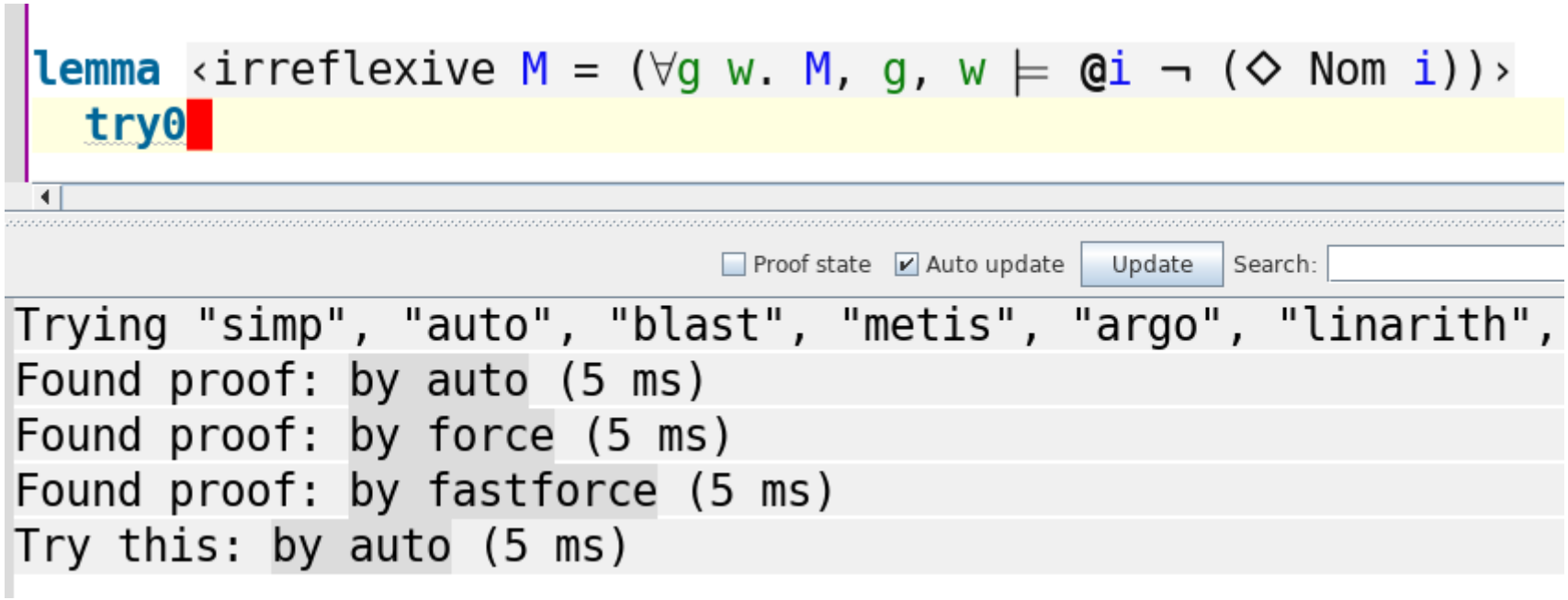
Example Proof I

```
abbreviation irreflexive :: ‹('w, 'b) model ⇒ bool› where
  ‹irreflexive M ≡ ∀w. w ∉ R M w›


lemma ‹irreflexive M ⟹ M, g, w ⊨ @i ¬ (◇ Nom i)›
proof -
  assume ‹irreflexive M›
  then have ‹g i ∉ R M (g i)›
    by simp
  then have ‹¬ (∃v ∈ R M (g i). g i = v)›
    by simp
  then have ‹¬ M, g, g i ⊨ ◇ Nom i›
    by simp
  then have ‹M, g, g i ⊨ ¬ (◇ Nom i)›
    by simp
  then show ‹M, g, w ⊨ @i ¬ (◇ Nom i)›
    by simp
qed
```

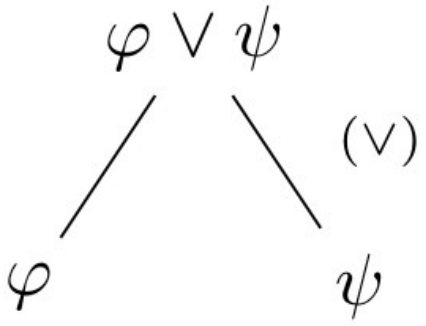
Example Proof II

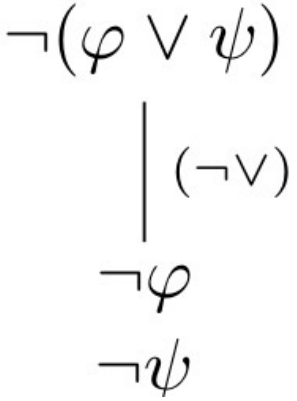
- Isabelle has powerful proof search.

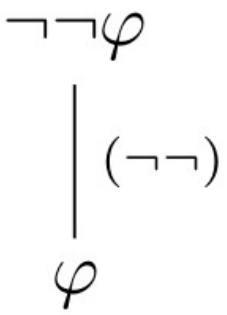


Calculus I

- Tableau system
- Satisfiability search
- Close branch if we reach a contradiction
- Original formula is valid if all branches close for negated formula







Calculus II

$$\dfrac{\Diamond\varphi \qquad (\Diamond)^1}{\Diamond i}\qquad \dfrac{\neg\Diamond\varphi \qquad \Diamond i}{\neg @_i\varphi}\,(\neg\Diamond)\qquad \dfrac{i \qquad @_i\varphi}{\varphi}\,(@)\qquad \dfrac{i \qquad \neg @_i\varphi}{\neg\varphi}\,(\neg @)$$

$$\dfrac{}{i}\,\mathsf{Name}^2 \qquad \dfrac{\varphi}{i}\,\mathsf{GoTo}^3 \qquad \dfrac{\varphi \qquad i}{i}\,\mathsf{Nom}^4$$

Calculus Example

1	$\neg(@_i j \wedge @_j \varphi \rightarrow @_i \varphi)$	
2	$@_i j \wedge @_j \varphi$	$(\neg \rightarrow)$ on 1
3	$\neg @_i \varphi$	$(\neg \rightarrow)$ on 1
4	$@_i j$	$(\wedge)$ on 2
5	$@_j \varphi$	$(\wedge)$ on 2
	———	
6	i	GoTo
7	j	(@) on 4,6
8	φ	(@) on 5,7
9	$\neg \varphi$	$(\neg @)$ on 3,6
	$\times$	

Calculus Embedding

- We inductively define the tableau rules.
- Definition over a single branch. Whole tree is implicit.

```
inductive ST :: ‹('a, 'b) branch ⇒ bool› ("⊢ _" [50] 50) where
```

- *a branch closes either by having φ and ¬φ inside a block, or inside two distinct blocks with the same opening nominal*

```
Close:
‹(ps, i) ∈ set branch ⟹ (qs, i) ∈ set branch ⟹
  p on (ps, i) ⟹ (¬ p) on (qs, i) ⟹
  ⊢ branch›
```

GoTo Considered Harmful

- System in paper has an initial block without an opening nominal
- I have removed this (along with the Name rule)
- Results in a simpler, more uniform system
- Prevents getting stuck due to silly errors:

$$\frac{\Diamond @_i \varphi \wedge \neg @_i \varphi}{i} \quad \text{GoTo}$$

- (Stronger weakening result)

Soundness

- Derivation of negation implies validity of original.
- Drawback of removing initial segment:
  We have to start in a fresh world

```
theorem soundness_fresh:
  assumes ‹⊢ [([¬ p], i)]› ‹i ∉ nominals p›
  shows ‹M, g, w ⊨ p›
```

- Around 150 lines of proof code. Four lines in the paper:

Now, the contrapositive of soundness follows from the observation that if a tableau T of the calculus ST has a branch which is block-wise satisfiable, then the tableau obtained by applying a rule to T also has a branch which is block-wise satisfiable. This can be seen simply by inspecting each rule in ST. ■

Completeness Overview

- Hintikka definition for sets of named blocks
- Model existence for any formula on a named block in a Hintikka set
- Lindenbaum-Henkin inspired construction of a maximally consistent set of named blocks
  - Consistency means there is no closed tableau
  - Includes witness blocks for diamonds
- Smullyan-Fitting inspired block lemma: A maximally consistent set of blocks is Hintikka

Completeness I

```
definition hintikka :: ‹('a, 'b) block set ⇒ bool› where
  ‹hintikka H =
   ((∀x i j. (∃ps. (ps, i) ∈ H ∧ Nom j on (ps, i)) ⟶ (∃qs. (qs, j) ∈ H ∧ Pro x on (qs, j)) ⟶
     (∄rs. (rs, i) ∈ H ∧ (¬ Pro x) on (rs, i))) ∧
    (∀a i. (∃ps. (ps, i) ∈ H ∧ Nom a on (ps, i)) ⟶ (∄qs. (qs, i) ∈ H ∧ (¬ Nom a) on (qs, i))) ∧
    (∀i j. (∃ps. (ps, i) ∈ H ∧ (◇ Nom j) on (ps, i)) ⟶
     (∄qs. (qs, i) ∈ H ∧ (¬ (◇ Nom j)) on (qs, i))) ∧
    (∀p i. i ∈ nominals p ∧ (∃block ∈ H. p on block) ⟶ (∃qs. (qs, i) ∈ H)) ∧
    (∀i j. (∃ps. (ps, i) ∈ H ∧ Nom j on (ps, i)) ⟶ (∃qs. (qs, j) ∈ H ∧ Nom i on (qs, j))) ∧
    (∀i j k. (∃ps. (ps, i) ∈ H ∧ Nom j on (ps, i)) ⟶ (∃qs. (qs, j) ∈ H ∧ Nom k on (qs, j)) ⟶
     (∃rs. (rs, i) ∈ H ∧ Nom k on (rs, i))) ∧
    (∀i j k. (∃ps. (ps, i) ∈ H ∧ (◇ Nom j) on (ps, i)) ⟶
     (∃qs. (qs, j) ∈ H ∧ Nom k on (qs, j)) ⟶ (∃rs. (rs, i) ∈ H ∧ (◇ Nom k) on (rs, i))) ∧
    (∀i j k. (∃ps. (ps, i) ∈ H ∧ (◇ Nom j) on (ps, i)) ⟶
     (∃qs. (qs, i) ∈ H ∧ Nom k on (qs, i)) ⟶ (∃rs. (rs, k) ∈ H ∧ (◇ Nom j) on (rs, k))) ∧
    (∀p q i. (∃ps. (ps, i) ∈ H ∧ (p ∨ q) on (ps, i)) ⟶
     (∃qs. (qs, i) ∈ H ∧ (p on (qs, i) ∨ q on (qs, i)))) ∧
    (∀p q i. (∃ps. (ps, i) ∈ H ∧ (¬ (p ∨ q)) on (ps, i)) ⟶
     (∃qs. (qs, i) ∈ H ∧ (¬ p) on (qs, i) ∧ (¬ q) on (qs, i))) ∧
    (∀p i. (∃ps. (ps, i) ∈ H ∧ (¬ ¬ p) on (ps, i)) ⟶ (∃qs. (qs, i) ∈ H ∧ p on (qs, i))) ∧
    (∀p i. (∃block ∈ H. (@ i p) on block) ⟶ (∃qs. (qs, i) ∈ H ∧ p on (qs, i))) ∧
    (∀p i. (∃block ∈ H. (¬ (@ i p)) on block) ⟶ (∃qs. (qs, i) ∈ H ∧ (¬ p) on (qs, i))) ∧
    (∀p i. (∄a. p = Nom a) ⟶ (∃ps. (ps, i) ∈ H ∧ (◇ p) on (ps, i)) ⟶
     (∃j. (∃qs. (qs, i) ∈ H ∧ (◇ Nom j) on (qs, i)) ∧ (∃rs. (rs, i) ∈ H ∧ (@ j p) on (rs, i)))) ∧
    (∀p i j. (∃ps. (ps, i) ∈ H ∧ (¬ (◇ p)) on (ps, i)) ⟶
     (∃qs. (qs, i) ∈ H ∧ (◇ Nom j) on (qs, i)) ⟶
     (∃rs. (rs, i) ∈ H ∧ (¬ (@ j p)) on (rs, i))))›
```

Completeness I’

- Error in Hintikka definition in paper:

(i) If there is an i-block in H with an atomic formula a on it, then there is no i-block in H with $\neg a$ on it.

- Should be something like:

  If there is an *i*-block in *H* with *j* on it and a *j*-block in *H* with an atomic formula *a* on it, then there is no *i*-block in H with ~a on it.

- Thank you, Isabelle

Completeness II

Lindenbaum-Henkin-construction then goes like this: Let S_1 be S. Suppose S_n has been constructed. Then:

$$S_{n+1} = \begin{cases} S_n, & \text{if } S_n \cup \{B_n\} \text{ is inconsistent,} \\ S_n \cup \{B_n\}, & \text{if } S_n \cup \{B_n\} \text{ is consistent, and on } B_n \text{ there is no } \Diamond\varphi, \\ S_n \cup \{B_n\} \cup \{B'\}, & \text{if } S_n \cup \{B_n\} \text{ is consistent, and on } B_n \text{ there is at least one } \Diamond\varphi, \text{ with } \varphi \text{ not a nominal and } B' \text{ is the } \Diamond\text{-witness for } B_n. \end{cases}$$

Finally, we'll say that a set S of finite named blocks is $\Diamond$-*saturated*, if for any $\Diamond\varphi$ occurring on any i-block $B \in S$, there are (possibly identical) i-blocks B_1 and B_2 with $\Diamond j$ and $@_j\varphi$ on them.

Lemma 3.4 (Lindenbaum-Henkin) *Any* `STB`*-consistent set of finite named blocks can be extended into a* $\Diamond$*-saturated maximally* `STB`*-consistent set of finite named blocks.*

Completeness III

```
lemma hintikka_Extend:
  fixes S :: ‹('a, 'b) block set›
  assumes inf: ‹infinite (UNIV :: 'b set)› and
    ‹maximal S› ‹consistent S› ‹saturated S›
  shows ‹hintikka S›

theorem main:
  assumes ‹i ∉ nominals p›
  shows ‹valid p ⟷ ⊢ [([¬ p], i)]›
```

- Almost! Missing a single case in the top lemma.

Bridge Elimination

- Needed for sixth Hintikka property. Current work.
- Transformation of tableau that is not just syntax-driven but relies on tableau context. Very tricky.

Lemma 4.2 (Elimination lemma) *Suppose B_1 is the i-block consisting of i and $\Diamond j$, B_2 is the j-block consisting of j and k, and B_3 is the i-block consisting of i and $\Diamond k$. Suppose furthermore that S is any finite set of finite named blocks. Given a finite* ST*-tableau T for $S \cup \{B_3\}$ we can construct another finite* ST*-tableau T' for $S \cup \{B_1\} \cup \{B_2\}$ such that there is a correspondence between the branches of T and T' in such a way, that given any branch Θ of T, the following holds for any formula φ occurring on any l-block in Θ:*

(i) *If φ does not descend from $\Diamond k$ in B_3, then φ occurs on an l-block of the corresponding Θ' in T'.*

(ii) *If φ descends from $\Diamond k$ in B_3, then φ^j occurs on an l-block of the corresponding Θ' in T'.*

Possible Future Work

- Showing termination of the system
  - … or a suitably modified system
  - Current termination proof is via translation to a different tableau system
- Extension to hybrid logic with binders
  - But then you lose decidability (so termination)
- Code-generating a verified prover
  - Requires showing termination
  - Decision procedure

References

- Blackburn, Patrick: "Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto", Logic Journal of the IGPL 8(3):339-365 (2000)
- Braüner, Torben: "Hybrid Logic", The Stanford Encyclopedia of Philosophy (Summer 2017 Edition), Edward N. Zalta (ed.)

All screenshots are from the two papers referenced in the beginning.