Kripke’s World
An introduction to modal logics via tableau systems

O. Gasquet, A. Herzig, B. Saïd, F. Schwarzentruber

Institut de Recherche en Informatique de Toulouse - IRIT
Université de Toulouse
http://www.irit.fr/Lotrec

UNILOG 2010
Background: logic and reasoning

- Classical propositional logic (CPL)
  - satisfiability problem decidable: NP-complete
  - reasoning:
    - Hilbert-style axiomatics, natural deduction
    - Gentzen sequent systems, tableaux
    - resolution
    - heuristic search: many SAT solvers

- Classical predicate logic
  - satisfiability problem semi-decidable
  - reasoning:
    - ... resolution [OTTER, SPASS, etc.]

- Higher-order logic
  - undecidable
  - reasoning:
    - Proof assistants [Isabelle, Coq, etc.]
Background and motivation

Modal logics
- variant: description logics (semantic web)
- infinitely many logics
- ‘surprisingly often decidable’
  - $\text{NP} < \text{PSPACE} < \text{EXPTIME} < \text{NEXPTIME} < \text{EXPSPACE}$
- reasoning:
  - Hilbert-style axiomatics, natural deduction
  - Gentzen sequent systems
  - resolution [Fariñas 83]
  - translation to FOL and resolution [Fariñas and Herzig 88, Ohlbach 88; MSPASS]
  - methods based on SAT solvers for CPL [K-SAT, etc.]
  - Tableaux

Idea: step-by-step introduction to modal logics via tableaux
From Tarski’s World to Kripke’s World

- Tarski’s World: introduction to predicate logic
  - examples = scenarios from geometry
  - book + program
- Kripke’s World: introduction to modal logics
  - examples = modal logics
  - reasoning = try to construct models = tableaux
  - program: LoTREC, http://www.irit.fr/Lotrec
  - book to come
Early history: les tableaux de Monsieur Toulouse-LauTREC
Outline

Part 1: Theory
1. Modal logics
2. Reasoning problems

Part 2: Practice
3. LoTREC
4. Implementing logics
Part 1: Theory

1 Modal logics
   - Possible worlds models
   - Classes of models
   - Language
   - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
Outline

1 Modal logics
   - Possible worlds models
     - Classes of models
     - Language
     - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
What is a Kripke model?

- Possible worlds
  - node
  - states

- Valuation
  - labeling function
  - interpretation

- Accessibility relation
  - labeled edges
  - transitions

- Model
  - labeled graph
  - transition system
Kripke Model

Given: a set $\mathcal{P}$ (propositional variables) and a set $\mathcal{I}$ (indexes):

- $M = (W, R, V)$
  - $W$: nonempty set
  - $R: \mathcal{I} \rightarrow 2^{W \times W}$
  - $V: W \rightarrow 2^{\mathcal{P}}$

- Pointed model $(M, w)$
  where $w \in W$ is the actual world

set of possible worlds
accessibility relation
valuation function
Readings of $R$

- Alethic:
  $wRu$ iff $u$ is possible given the actual world $w$

- Temporal:
  $wRu$ iff $u$ is in the future of $w$

- Epistemic:
  $wR_I u$ iff $u$ is possible for agent $I$ at actual world $w$

- Deontic:
  $wRu$ iff $u$ is an ideal counterpart of the actual world $w$

- Dynamic:
  $wR_I u$ iff $u$ is a possible result of the execution of the program / action $I$ in $w$
Readings of $R$

- Alethic:
  \[ wRu \iff u \text{ is possible given the actual world } w \]

- Temporal:
  \[ wRu \iff u \text{ is in the future of } w \]

- Epistemic:
  \[ wR_Iu \iff u \text{ is possible for agent } I \text{ at actual world } w \]

- Deontic:
  \[ wRu \iff u \text{ is an ideal counterpart of the actual world } w \]

- Dynamic:
  \[ wR_Iu \iff u \text{ is a possible result of the execution of the program / action } I \text{ in } w \]

Readings of $R \implies$ Properties of $R$
Defining a model in LoTREC

How to build a graph with two nodes:

- open a new logic (menu ‘Logic’)
- add a new rule (‘Rules’ tab):
  - no conditions
  - in the action part:
    - createNewNode w
    - createNewNode u
    - link w u R
    - add w P
- edit the default strategy (‘Strategies’ tab):
  - call the new rule (double click)
Outline

1 Modal logics
   ▪ Possible worlds models
   ▪ Classes of models
   ▪ Language
   ▪ Semantics

2 Reasoning problems
   ▪ Validity and satisfiability in a class of models
   ▪ Outline of the tableaux method
Classes of models

- A class of models can be defined by
  - constraints on the accessibility relation
  - constraints on the valuation
- Applications?
- Mathematical properties?
Constraints on a single relation $R$

- Singleton models: \( \{ M : \text{card}(W) = 1 \} \)
- Serial
  - "there is always a future"
    - for all \( w \) exists \( u \) s.th. \( wRu \)
- Reflexive
  - "knowledge implies truth"
- Transitive
  - "future of future is future"
    - "I know what I know"
- Symmetric
- Euclidian
  - "I know what I don’t know"
- Confluent (Church-Rosser)
- Equivalence
- Universal
- ...
Constraints involving several relations

- $R_I$ included in $R_J$
- $R_I = R_J \cup R_K$
- $R_J = (R_I)^{-1}$
- $R_J = (R_I)^*$ \hspace{2cm} \textit{(reflexive and transitive closure)}
- $R_I \circ R_J = R_J \circ R_I$ \hspace{2cm} \textit{(permutation)}
- Confluent
- ...
Constraints on the valuation $V$

- names for worlds ('nominals '):
  
  If $N \in V(w)$ and $N \in V(u)$ then $w = u$
  
  $\implies$ hybrid logic

- $R$ is hereditary (atomic propositions persist)
  
  If $P \in V(w)$ and $wR u$ then $P \in V(u)$
  
  $\implies$ intuitionistic logic
Closing under constraints in LoTREC

- Closing under reflexivity:
  - condition: `isNewNode w`
  - action: `link w w R`

- Observe:
  - capital first letter $\rightarrow$ constant
  - small first letter $\rightarrow$ variable

- Exercise: make $R$ hereditary
Outline

1 Modal logics
   - Possible worlds models
   - Classes of models
   - Language
   - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
Boolean formulas

- atomic formulas = elements of $\mathcal{P}$ (propositional variables)
- complex formulas: built using the Boolean connectors
  - $\neg A$ = “not $A$”
  - $A \land B$ = “$A$ and $B$”
  - $A \lor B$ = “$A$ or $B$”
  - $A \rightarrow B$ = “if $A$ then $B$”
  - $A \leftrightarrow B$ = “$A$ if and only if $B$”
  - $A \oplus B$ = “either $A$ or $B$”
  - $\oplus(A, B, C)$ = “either $A$, or $B$, or $C$”
  ...

Modal formulas

- **Temporal logic**
  - $XA = \ "A \text{ will be true at the next time point}\"$
  - $FA = \ "A \text{ will be true at some time point in the future}\"$
  - $A = \ "A \text{ will eventually be true}\"$
  - $GA = \ "A \text{ will be true at every time point in the future}\"$
  - $AUB = \ "A \text{ until } B\”$
  - $ASB = \ "A \text{ since } B\”$

- **Dynamic logic**
  - $\text{After}_I A = \ "A \text{ will be true after every possible execution of program } I\”$
  - $\text{Feasible}_I A = \ "A \text{ will be true after some execution of program } I\”$

(programs may be nondeterministic)
Modal formulas (ctd.)

- **Epistemic and doxastic logic**
  \[ \text{Bel}_I A = \text{“agent } I \text{ believes that } A\” \]
  \[ \text{K}_I A = \text{“agent } I \text{ knows that } A\” \]
  \[ \hat{\text{Bel}}_I A = \text{“it is (doxastically) possible for agent } I \text{ that } A\” \]
  \[ \hat{\text{K}}_I A = \text{“it is (epistemically) possible for agent } I \text{ that } A\” \]

- **Deontic logic**
  \[ \text{O}_I B = \text{“} A \text{ is obligatory for } I \text{”} \]
  \[ \text{P}_I B = \text{“} A \text{ is permitted for } I \text{”} \]

- **Intuitionistic logic**
  \[ A \Rightarrow B = \text{“} A \text{ implies } B\” \text{ (like } \rightarrow \text{, but no excluded middle)} \]

- **Conditional logic**
  \[ A \Rightarrow B = \text{“} A \text{ implies } B\” \text{ (} \Rightarrow \text{ ‘stronger’ than } \rightarrow \text{)} \]

- ...
“Un pour tous, tous pour un” [A. Dumas]

- An abstraction: necessity and possibility
  \[ \Diamond A = MA = \text{“}A\text{ is possible”} \]
  \[ \Box A = LA = \text{“}A\text{ is necessary”} \]

- Multimodal version:
  \[ \Diamond_I A = \langle I \rangle A = \text{“}A\text{ is possible w.r.t. } I\text{”} \]
  \[ \Box_I A = [I]A = \ldots \]
  where \( I \in \mathcal{I} \) (set of parameters)

- Common feature: Not truth-functional
  - no \( f \) s.th. \( \text{truthvalue}(\Diamond A) = f(\text{truthvalue}(A)) \)
Duality

- Intuitively:
  \[ \hat{K}_IA \leftrightarrow \neg K_I \neg A \]
  \[ P_I A \leftrightarrow \neg O_I \neg A \]
  \[ FA \leftrightarrow \neg G \neg A \]
  \[ After_I A \leftrightarrow \neg \text{Feasible}_I \neg A \]
  \[ \ldots \]

- Abstracting:
  \[ \Diamond A \leftrightarrow \neg \Box \neg A \]
  \[ \Box A \leftrightarrow \neg \Diamond \neg A \]

- Options:
  - take both \( \Diamond \) and \( \Box \) as primitive
  - take \( \Diamond \) as primitive, and set \( \Box A \overset{\text{def}}{=} \neg \Diamond \neg A \)
  - take \( \Box \) as primitive, and set \( \Diamond A \overset{\text{def}}{=} \neg \Box \neg A \)
How define a language?

- Examples
  - $\text{CardRed} \land K_{\text{Ann}} \text{CardRed} \land K_{\text{Ann}} \lnot K_{\text{Bob}} \text{CardRed}$
  - $\text{DoorClosed} \land [\text{Open}]\text{DoorOpen}$
  - $P \land \lnot Q \land \Box Q \land \Diamond (P \land \Box \lnot Q)$

- Language = set of formulas
- Language is defined by BNF:

  \[ A ::= P \mid \lnot A \mid A \land A \mid A \lor A \mid \Diamond A \mid \Box A \mid \langle I \rangle A \mid [I] A \mid K_I A \mid \ldots \]

  where $P$ ranges over $\mathcal{P}$ and $I$ ranges over $\mathcal{I}$
How define a language in LoTREC?

- Formulas in LoTREC: prenex form
  $\implies$ General schema: $op(A_1, \ldots, A_n)$

  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, B)$
  $\ldots$
  $\neg A = not(A)$
  $A \land B = and(A, B)$
  $A \lor B = or(A, b)
Outline

1 Modal logics
   - Possible worlds models
   - Classes of models
   - Language
   - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
Truth conditions

- **Atoms**
  - $M, w \models P$ iff $P \in V(w)$

- **Classical connectors**
  - $M, w \models A \land B$ iff $M, w \models A$ and $M, w \models B$
  - $M, w \models A \lor B$ iff . . .
  - . . .

- **Modal operators**
  - $M, w \models \Diamond A$ iff there exists $u$ s.th. $wRu$ and $M, u \models A$
  - $M, w \models \Box A$ iff for all $u$, if $wRu$ then $M, u \models A$
Truth conditions

- Multi-modal operators
  - $M, w \models \langle I \rangle A$ iff there exists $u$ s.th. $w R_I u$ and $M, u \models A$
  - ... 

- Relation algebra operators
  - $M, w \models \Diamond^{-1} A$ iff there exists $u$ s.th. $w R^{-1} u$ and $M, u \models A$
  - $M, w \models \langle I \cup J \rangle A$ iff there exists $u$ s.th. $w (R_I \cup R_J) u$ and $M, u \models A$
  - $M, w \models \langle I^* \rangle A$ iff there exists $u$ s.th. $w (R_I)^* u$ and $M, u \models A$
Truth conditions

Temporal operators (linear time)

- \( M, w \models XA \) iff there exists \( u \) s.th. \( wRu \) and \( M, u \models A \)
- \( M, w \models FA \) iff there exists \( n, u \) s.th. \( wR^n u \) and \( M, u \models A \)

\[
\begin{align*}
A & \xrightarrow{w} A & & \cdots & & \xrightarrow{u} A & \xrightarrow{} B \\
A & \xrightarrow{} A & & \cdots & & \xrightarrow{} A & \xrightarrow{} B
\end{align*}
\]

- \( M, w \models A \cup B \) iff there exists \( u \) s.th. \( wR^* u \) and \( M, u \models B \) and \( M, v \models A \) for all \( v \) s.th. \( (wR^* v \) and \( vR^+ u) \)

- \( \ldots \)
Model checking

Given $M$, $w$, and $A$, do we have $M, w \models A$?

- Model checking problem
  - can be solved in polynomial time for most modal logics
- Model checking in LoTREC
  - requires more LoTREC primitives $\Rightarrow$ later
Part 1: Theory

1 Modal logics
   - Possible worlds models
   - Classes of models
   - Language
   - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
Outline

1 Modal logics
   - Possible worlds models
   - Classes of models
   - Language
   - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
Validity and satisfiability in the set of all models

$K = \text{the set of all possible worlds models (Kripke)}$

- $A$ is valid in $K$ iff for all $M$ in $K$ and all $w$ in $M$: $M, w \models A$

Example

- $\square(P \lor \neg P)$
- $\square P \land \square Q \rightarrow \square(P \land Q)$

- $A$ is satisfiable in $K$ iff for some $M$ in $K$ and some $w$ in $M$: $M, w \not\models A$

Example

- $P$
- $P \land \neg \square P$
- $P \land \square \neg P$
- $\square P \land \neg \square \square P$
Validity and satisfiability in the set of all models

$K$ = the set of all possible worlds models ($K_{\text{Kripke}}$)

- $A$ is valid in $K$ iff for all $M$ in $K$ and all $w$ in $M$: $M, w \models A$

Example

- $\Box(P \lor \neg P)$
- $\Box P \land \Box Q \rightarrow \Box(P \land Q)$

- $A$ is satisfiable in $K$ iff for some $M$ in $K$ and some $w$ in $M$: $M, w \not\models A$

Example

- $P$
- $P \land \neg \Box P$
- $P \land \Box \neg P$
- $\Box P \land \neg \Box \Box P$
Validity and satisfiability in some class of models

\[ C = \text{some subset of } K \]

- **A is valid in } C \text{ iff for all } M \text{ in } C \text{ and all } w \text{ in } M: M, w \models A**

**Example**

- \( \Box P \rightarrow P \) invalid in \( K \)
- \( \Box P, \neg P \rightarrow P \)
- \( \Box P \rightarrow P \) valid in the class of reflexive models
- \( \Diamond \Diamond P \rightarrow \Diamond P \) valid in transitive models

- **A is satisfiable in } C \text{ iff for some } M \text{ in } C \text{ and some } w \text{ in } M: M, w \not\models A**

**Example**

- \( P \land \Box \neg P \) is satisfiable in \( K \)
- \( P \land \Box \neg P \) is unsatisfiable in the class of reflexive models

- **A is valid in } C \text{ iff } \neg A \text{ is unsatisfiable in } C**


Validity and satisfiability in some class of models

\( \mathcal{C} = \) some subset of \( \mathcal{K} \)

- \( A \) is valid in \( \mathcal{C} \) iff for all \( M \) in \( \mathcal{C} \) and all \( w \) in \( M \): \( M, w \models A \)

**Example**

- \( \Box P \to P \) invalid in \( \mathcal{K} \)
- \( \Box P, \neg P \leadsto P \)

- \( \Box P \to P \) valid in the class of reflexive models
- \( \Diamond \Diamond P \to \Diamond P \) valid in transitive models

- \( A \) is satisfiable in \( \mathcal{C} \) iff for some \( M \) in \( \mathcal{C} \) and some \( w \) in \( M \): \( M, w \not\models A \)

**Example**

- \( P \land \Box \neg P \) is satisfiable in \( \mathcal{K} \)
- \( P \land \Box \neg P \) is unsatisfiable in the class of reflexive models

\[ A \text{ is valid in } \mathcal{C} \text{ iff } \neg A \text{ is unsatisfiable in } \mathcal{C} \]
Validity and satisfiability in some class of models

\[ \mathcal{C} = \text{some subset of } K \]

- \( A \) is valid in \( \mathcal{C} \) iff for all \( M \) in \( \mathcal{C} \) and all \( w \) in \( M \): \( M, w \models A \)

**Example**

- \( \Box P \rightarrow P \) invalid in \( K \)
- \( \Box P, \neg P \rightarrow P \)
- \( \Box P \rightarrow P \) valid in the class of reflexive models
- \( \Diamond \Diamond P \rightarrow \Diamond P \) valid in transitive models

- \( A \) is satisfiable in \( \mathcal{C} \) iff for some \( M \) in \( \mathcal{C} \) and some \( w \) in \( M \): \( M, w \not\models A \)

**Example**

- \( P \land \Box \neg P \) is satisfiable in \( K \)
- \( P \land \Box \neg P \) is unsatisfiable in the class of reflexive models

\[ A \text{ is valid in } \mathcal{C} \text{ iff } \neg A \text{ is unsatisfiable in } \mathcal{C} \]
Examples

- Singleton models: \( \{ M : \text{card}(W) = 1 \} \)
  valid: \( \Diamond A \rightarrow \Box A \)

- Reflexive models: KT
  valid: \( \Box A \rightarrow A \)

- Transitive models: K4
  valid: \( \Diamond \Diamond A \rightarrow \Diamond A \)

- Reflexive and transitive models: S4
  valid: \( \ldots \)

- Equivalence relation: S5
  valid: \( A \rightarrow \Box \Diamond A, \ldots \)

- \( \ldots \)
Validity and satisfiability in a class of models

Reasoning problems

- Model checking
  Given $M$, $w$, and $A$ do we have $M, w \models A$?

- Validity
  Given $A$ and $C$ is $A$ valid in $C$?

- Satisfiability
  Given $A$ and $C$ does there exist $M$ in $C$ and $w$ in $M$: $M, w \models A$?

- Model construction
  Given $A$ and $C$ compute $M$ in $C$ and $w$ in $M$: $M, w \models A$

How can we solve them automatically?
Outline

1 Modal logics
   - Possible worlds models
   - Classes of models
   - Language
   - Semantics

2 Reasoning problems
   - Validity and satisfiability in a class of models
   - Outline of the tableaux method
Classical logic [Beth]

Checking the satisfiability of a given formula $A$:

1. Try to find $M$ and $w$ by applying truth conditions
   - $M, w \vDash A_1 \land A_2 \implies$ add $M, w \vDash A_1$, and add $M, w \vDash A_2$
   - $M, w \vDash A_1 \lor A_2 \implies$ either add $M, w \vDash A_1$, or add $M, w \vDash A_2$
     (nondeterministic)
   - $M, w \vDash \neg A_1 \implies$ don’t add $M, w \vDash A_1$ !!!
     - $M, w \vDash \neg \neg A_1 \implies$ add $M, w \vDash A_1$
     - $M, w \vDash \neg (A_1 \lor A_2) \implies$ add $M, w \vDash \neg A_1$ and add $M, w \vDash \neg A_2$
     - $M, w \vDash \neg (A_1 \land A_2) \implies$ add $M, w \vDash \neg A_1$ or add $M, w \vDash \neg A_2$

$\implies$ tableau rules

2. apply while possible (saturation)

3. is $M$ a model?
   - NO if both $M, w \vDash B$ and $M, w \vDash \neg B$ (closed tableau)
   - ELSE $M$ is a model for $A$ (open tableau)
     $W = \{w\}, R = \emptyset, V(w) = \{P : M, w \vDash P\}$
Classical logic [Beth]

Checking the satisfiability of a given formula $A$:

1. Try to find $M$ and $w$ by applying truth conditions
   - $M, w \models A_1 \land A_2 \implies$ add $M, w \models A_1$, and add $M, w \models A_2$
   - $M, w \models A_1 \lor A_2 \implies$ either add $M, w \models A_1$, or add $M, w \models A_2$
     (nondeterministic)
   - $M, w \models \neg A_1 \implies$ don’t add $M, w \models A_1$ !!
     - $M, w \models \neg \neg A_1 \implies$ add $M, w \models A_1$
     - $M, w \models \neg (A_1 \lor A_2) \implies$ add $M, w \models \neg A_1$ and add $M, w \models \neg A_2$
     - $M, w \models \neg (A_1 \land A_2) \implies$ add $M, w \models \neg A_1$ or add $M, w \models \neg A_2$

2. apply while possible (saturation)

3. is $M$ a model?
   - NO if both $M, w \models B$ and $M, w \models \neg B$ (closed tableau)
   - ELSE $M$ is a model for $A$ (open tableau)
     - $W = \{w\}$, $R = \emptyset$, $V(w) = \{P : M, w \models P\}$
Modal logic [Fitting]

Basic cases

- **$M, w \models \Diamond A$**
  - $\implies$ add some new node $u$, add $wRu$, add $M, u \models A$

- **$M, w \models \Box A$**
  - $\implies$ for all node $u$ s.th. $wRu$, add $M, u \models A$

Apply truth conditions $\equiv$ build a labeled graph

- create nodes
- add links
- add formulas to nodes
Example

A node with the input formula

\[ \square P \land \leftrightarrow Q \land \leftrightarrow (R \lor \neg P) \]
Example

\[ M, w \models A \land B \text{ iff } M, w \models A \text{ and } M, w \models B \]

- \( A \) is \( \Box P \)
- \( B \) is \( \Diamond Q \land \Diamond (R \lor \neg P) \)

\[ \square P \land \leftrightarrow Q \land \leftrightarrow (R \lor \neg P) \]
Example

\[ M, w \vDash A \land B \iff M, w \vDash A \text{ and } M, w \vDash B \]

- \( A \) is \( \Box P \)
- \( B \) is \( \Diamond Q \land \Diamond (R \lor \neg P) \)

\[
\begin{align*}
[] P & \land <> Q & <> (R \lor \neg P) \\
[] P & \\
<> Q & <> (R \lor \neg P)
\end{align*}
\]
Example

\( M, w \models A \land B \iff M, w \models A \text{ and } M, w \models B \)

\[
\begin{array}{c}
[] \ P & \leftrightarrow Q & \leftrightarrow (R \lor \neg P) \\
[] \ P \\
\leftrightarrow \ Q & \leftrightarrow (R \lor \neg P) \\
\leftrightarrow \ Q \\
\leftrightarrow \ (R \lor \neg P)
\end{array}
\]
Example

\[ M, w \models \Diamond A \text{ iff there is } u \text{ s.th. } wRu \text{ and } M, u \models A \]

\[
\begin{array}{c}
\Box P & \land & \lnot Q & \land & \lnot (R v \lnot P) \\
\Box P \\
\lnot Q & \land & \lnot (R v \lnot P) \\
\lnot Q \\
\lnot (R v \lnot P) \\
\end{array}
\]

R

R

R

R v \lnot P

Q
$M, w \models \square A$ iff for all $u$: if $wRu$ then $M, u \models A$

```
[] P & <> Q & <> (R v ~ P)
    [] P
    <> Q & <> (R v ~ P)
        <> Q
        <> (R v ~ P)
```

```
R         R
```

```
R v ~ P
P
```

```
Q
P
```
Validity and satisfiability in a class of models

Outline of the tableaux method

Example

\[ M, w \models A \vee B \iff M, w \models A \text{ or } M, w \models B \]

premodel 1

\[
\begin{aligned}
\Box P & \land \leftrightarrow Q & & \leftrightarrow (R \vee \neg P) \\
\Box P \\
\leftrightarrow Q & & \leftrightarrow (R \vee \neg P) \\
\leftrightarrow Q \\
\leftrightarrow (R \vee \neg P)
\end{aligned}
\]

premodel 2

\[
\begin{aligned}
\Box P & \land \leftrightarrow Q & & \leftrightarrow (R \vee \neg P) \\
\Box P \\
\leftrightarrow Q & & \leftrightarrow (R \vee \neg P) \\
\leftrightarrow Q \\
\leftrightarrow (R \vee \neg P)
\end{aligned}
\]

41 / 97
Validity and satisfiability in a class of models

Outline of the tableaux method

Example

Premodel 1

Premodel 2
Validity and satisfiability in a class of models

Example

Outline of the tableaux method

\[ \begin{align*}
\Box P & \land \langle\rangle Q & \land \langle\rangle (R \lor \neg P) \\
\Box P \\
\langle\rangle Q & \land \langle\rangle (R \lor \neg P) \\
\langle\rangle Q \\
\langle\rangle (R \lor \neg P)
\end{align*} \]

premodel 1

\[ \begin{align*}
R & \lor R \\
R \lor \neg P \\
P & \lor R
\end{align*} \]

\[ M, w \models P \text{ then } P \in V(w) \]

\[ \begin{align*}
\neg P & \lor P
\\
\neg Q & \lor Q
\\
\neg R & \lor R
\end{align*} \]

Model

\[ \text{extraction} \]
A short history of tableaux

Handwritten proofs since 1950’s

■ ... Sequent calculi [Beth, Gentzen]
■ Tableaux calculi
  (tableau proof = sequent proof backwards)
■ Kripke: explicit accessibility relation
■ Smullyan, Fitting: uniform notation
■ Single-step tableaux [Massacci]
  \[ \sigma : \Diamond A \implies \sigma, n : A \]
■ Tableaux by graph rewriting [Castilho et al.]

Nowadays: automated provers

■ fast: FaCT [Horrocks], LWB [Heuerding, Jäger et col.], K-SAT [Giunchiglia&Sebastiani]
■ generic: TWB [Abate&Goré], LoTREC
Part 2: Practice

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
A short history of LoTREC

- before 2000: theoretical bases (Luis Fariñas del Cerro, Olivier Gasquet, Andreas Herzig)
- David Fauthoux [2000]
  - rewriting kernel
  - event-based implementation
  - K, KT, KB
- Mohamad Sahade [2002-2005]
  - loopchecking
  - more logics: S4, K4, …
  - general completeness and termination proofs
- Bilal Saïd [2006-2010]
  - LTL, PDL…
  - Confluence & commutative patterns
  - Model checking
  - graph rewriting basis & their theoretical properties
  - GUI, full web accessibility, step-by-step run,…
  - …
The black box

Logic Definition

Input Formula
- Partial Premodel

LoTREC

Graphs
- Extensible to models
- Not Extensible to models
Outline

3 LoTREC
  - Language
  - Rules
  - Strategies
  - Tableau notation
  - Do the algorithms do the right thing?

4 Implementing logics
  - Classical logic
  - Modal logic K
  - Multi-modal logic $K_n$
  - KT
  - KD
  - S4
  - Intuitionistic logic LJ
  - Model checking in LoTREC
  - PDL
  - Suggestions
User-defined language

Atomic propositions

- Any constant symbol = Capital_1st_letter_words

Formulas

- Prefix notation (but can be displayed in infix form)
- Priority and associativity to avoid printing parentheses

### Example (definition)

<table>
<thead>
<tr>
<th>name</th>
<th>arity</th>
<th>display</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>1</td>
<td>~ _</td>
</tr>
<tr>
<td>and</td>
<td>2</td>
<td>_ &amp; _</td>
</tr>
<tr>
<td>nec</td>
<td>1</td>
<td>[] _</td>
</tr>
<tr>
<td>pos</td>
<td>1</td>
<td>&lt;&gt; _</td>
</tr>
</tbody>
</table>

### Example (usage)

- pos P
  - displayed: <> P
- and not Q not P
  - displayed: ~ Q & ~ P
Outline

3 LoTREC
   - Language
   - Rules
   - Strategies
   - Tableau notation
   - Do the algorithms do the right thing?

4 Implementing logics
   - Classical logic
   - Modal logic K
   - Multi-modal logic $K_n$
   - KT
   - KD
   - S4
   - Intuitionistic logic LJ
   - Model checking in LoTREC
   - PDL
   - Suggestions
Truth conditions
+ Structural constraints

\[ M, w \models A \land B \text{ iff } \]
\[ M, w \models A \text{ and } M, w \models B \]

as Graph rewriting rules

\[ A \land B \]
\[ A \]
\[ B \]
On paper

Truth conditions

+ Structural constraints

as Graph rewriting rules

\[ M, w \vdash \Diamond A \iff \exists u \text{ s.th. } wRu \text{ and } M, u \vdash A \]
Truth conditions

+ Structural constraints

as Graph rewriting rules

Model is reflexive
Graph rewriting rule as “if Conditions ... then Actions”

Rule And
  hasElement node and variable A variable B
  add node variable A
  add node variable B
End
Graph rewriting rule as “if Conditions ... then Actions”

**Rule Pos**

- `hasElement node1 pos variable A`
- `createNewNode node2`
- `link node1 node2 R`
- `add node2 variable A`

**End**
In LoTREC

Graph rewriting rule as “if Conditions ... then Actions”

Rule ReflexiveEdges
   isNewNode node
   link node node R
End
Semantics of rules: the basic idea

Apply rule to a graph $G = \text{apply to every formula in every node}$

$\implies$ strategies get more declarative

$\implies$ proofs get easier

Tableau rules expand directed graphs by

- adding links
- adding nodes
- adding formulas
- duplicating the graph

$$\text{rule}(G) = \{G_1, \ldots, G_n\}$$

$$\text{rule}(\{G_1, \ldots, G_n\}) = \text{rule}(G_1) \cup \ldots \cup \text{rule}(G_n)$$
Managing graph copies: depth-first
Managing graph copies: depth-first
Managing graph copies: depth-first
Managing graph copies: depth-first
Managing graph copies: depth-first

Premodel$_1$  Premodel$_2$

...  ...

Premodel$_n$
Outline

3 LoTREC
   - Language
   - Rules
   - Strategies
   - Tableau notation
   - Do the algorithms do the right thing?

4 Implementing logics
   - Classical logic
   - Modal logic K
   - Multi-modal logic $K_n$
   - KT
   - KD
   - S4
   - Intuitionistic logic LJ
   - Model checking in LoTREC
   - PDL
   - Suggestions
Why a strategy?

- **Apply rules in order:**
  
  ```
  Strategy performOnce
  Stop
  And
  Or
  ...
  ```

- **Saturation:**
  
  ```
  Strategy CPL_strat
  repeat
  Stop
  NotNot
  And
  Or
  end
  ```

  ```
  Strategy K_strat
  repeat
  CPL
  Pos
  Nec
  end
  ```
Semantics of strategies

- block: rule1 ... rule_n ... anotherStrategy ...  
  apply all applicable rules in order then stop

Example

Strategy CPL
  Stop
  And
  Or
  Not_Not
  ...

Language  Rules  Strategies  Tableau notation  Do the algorithms do the right thing?
Semantics of strategies

- block: `rule1 ... rulen ... anotherStrategy ...`
  apply all applicable rules in **order** then **stop**

- **repeat** block **end**
  repeat until no rule applicable (**saturation**)

**Example**

Strategy **K**

**repeat**
  CPL
  Pos
  Nec
**end**

For simple logics: **repeat** and blocks are sufficient!
Semantics of strategies

- block: rule1 ... rulen ... anotherStrategy ... 
  apply all applicable rules in order then stop

- repeat block end
  repeat until no rule applicable (saturation)

- firstRule block end
  apply first applicable rule, then stop (unfair!) cf. higher-order proof assistants

Example

repeat
  firstRule
    rule1
    rule2  X
  end
end

rule1 is always applicable
rule2 is applicable
BUT never applied!
Semantics of strategies

- block: `rule1 ... rulen ... anotherStrategy ...`
  apply all applicable rules in `order` then `stop`

- `repeat` block `end`
  repeat until no rule applicable (saturation)

- `firstRule` block `end`
  apply first applicable rule, then stop (unfair!) cf. higher-order proof assistants

- `allRules` block `end`
  exactly as a “block”, but needed inside `firstRule`

**Example**
```
  firstRule
    rule1
    allRules
      rule2
      rule3
    end
  rule4
end
```
Semantics of strategies

- **block**: `rule1 ... rulen ... anotherStrategy ...`  
  apply all applicable rules in **order** then **stop**

- **repeat** block **end**  
  repeat until no rule applicable (**saturation**)  

- **firstRule** block **end**  
  apply first applicable rule, then stop (**unfair!**) cf. higher-order proof assistants

- **allRules** block **end**  
  exactly as a “**block**”, but needed inside **firstRule**

- **applyOnce** rule  
  apply the rule on **only one occurrence**
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
Tableau definition

The set of tableaux for formula $A$ with strategy $S$ is the set of graphs obtained by applying the strategy $S$ to an initial single-node graph whose root contains only $A$.

- Notation: $S(A)$

Remark
our tableau = “tableau branch” in the literature
(sounds odd to call a graph a branch)
Open or Closed?

- A **node** is closed iff it contains "**False**" (unless...)  
- A **tableau** is closed iff it has a **closed node**  
- A **set of tableaux** is closed iff all its elements are closed

An open tableau is a premodel  
\[\implies\text{ build a model}\]
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic $K$
- Multi-modal logic $K_n$
- $KT$
- $KD$
- $S4$
- Intuitionistic logic $LJ$
- Model checking in LoTREC
- PDL
- Suggestions
Formal properties

To be proved for each strategy $S$:

- **Termination**
  For every $A$, $S(A)$ terminates.

- **Soundness**
  If $S(A)$ is closed then $A$ is unsatisfiable.

- **Completeness**
  If $S(A)$ is open then $A$ is satisfiable.
Soundness proofs: easy (we just apply truth conditions)
Termination proofs: not so easy (case-by-case)
Completeness proofs...
  ... for fair strategies: standard techniques work “in most cases”
  but fair strategies do not terminate in general
  ... for terminating strategies: difficult
  rigorous proofs rare even for the basic modal logics!
  reason: strategy = imperative programming
In general. . .

BUT soundness + termination is practically sufficient (e.g. when experimenting with a logic):

- given: class of models $C$, strategy $S$, formula $A$
- apply strategy $S$ to $A$
- take an open tableau and build pointed model $(M, w)$
- check if $M$ in desired class of models
- check if $M, w \models A$
A general termination theorem

[O. Gasquet et al., AIML 2006]

- IF for every rule $\rho$:
  - the RHS of $\rho$ contains strict subformulas of its LHS
  - AND
  - some restriction on node creation

- THEN
  - for every formula $A$:
  - the tableaux construction terminates
Another general termination theorem

[O. Gasquet et al., AIML 2006]

- IF for every rule $\rho$:
  - the RHS of $\rho$ contains subformulas of its LHS
  - AND
  - some restriction on node creation
  - AND
  - some loop testing in the strategy
- THEN
  - for every formula $A$:
  - the tableaux construction terminates
Part 2: Practice

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
How to get LoTREC

http://www.irit.fr/Lotrec  (Capital “L”)

- Download ⇒ Executable to get LoTREC_2.0.zip
  - unzip
  - run file run.bat
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- $K_T$
- $K_D$
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
How to proceed

CPL: Classical Propositional Logic

1. From the task pane, open:
   - *Open Predefined logic* \(\Rightarrow\) *Others* \(\Rightarrow\) *CPL*

2. Run with
   - *Build Models*

3. Why these results?
   - Predefined formula
   - Predefined Main strategy

4. Review the logic definition: *Connectors, Rules*…

5. Change the formula

6. Re-run…
Adding “↔”

What about formulas with “↔” connector?

1. Save as CPL locally as “CPL_complete.xml”
2. Add to Connectors:
<table>
<thead>
<tr>
<th>name</th>
<th>arity</th>
<th>display</th>
<th>priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>equiv</td>
<td>2</td>
<td><em>&lt;-&gt;</em></td>
<td>0 (lowest)</td>
</tr>
</tbody>
</table>
3. Add to Rules:
   Equiv, and NotEquiv
4. Call them in the strategy
5. Try some formulas...
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
From CPL to K

- Here: minimal set of connectors $\neg$, $\land$, $\Box$ only
- Rules of CPL
  - Rule for $\neg\Box A$:
    - for every $\neg\Box A$ at every node $w$:
      - create a successor $u$ and add $\neg A$ to it
  - Rule for $\Box A$:
    - for every $\Box A$ at every $w$, and for every $R$-successor $u$ of $w$:
      - add $A$ to $u$
- Strategy: saturate with all the rules...
Rules

- **Rule NotNec**
  - `hasElement w pos variable a`
  - `createNewNode u`
  - `link w u R`
  - `add u variable a`

- **Rule Nec**
  - `hasElement w nec variable a`
  - `isLinked w u R`
  - `add u variable a`
Strategies

1. Continue with your “CPL_complete.xml”,
   or
   Open Predefined logic → Others → CPL_complete
2. Add the nec connector
3. Add the rules Nec and NotNec
4. Add a new strategy KStrategy which calls repeatedly CPLStrategy and then the rules Pos and Nec
5. Test with [] P & <> Q & <> (R ∨ ~ P)
   i.e. and nec P and pos Q pos or R not P
6. Test with other formulas...
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
From K To $K_n$

- Replace the connector □ by [\_-]
- Change all the predefined formulae 😞
- Change the modal rules: Nec and NotNec

**Rule Nec_K**

```
hasElement w nec variable a
isLinked w u R
add u variable a
```

**Rule Nec_Multimodal_K**

```
hasElement w nec variable r
variable a
isLinked w u variable r
add u variable a
```
How to proceed

1. From the task pane, open:
   
   *Open Predefined logic* $\implies$ *Others* $\implies$ *Multimodal-K*

2. Check $\neg[1]P \land \neg[2]\neg P$, …
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic Kₙ
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
Accessibility relation $R$ is reflexive

- **Aim:** close all tableaux for $\neg(\Box P \rightarrow P)$ (negation of axiom T)
- **Idea$_1$:** integrate reflexivity into the truth condition
  - $M, w \models \Box A$ iff $M, w \models A$, and $M, u \models A$ for every $u$ that is accessible from $w$ via $R$
- **Idea$_2$:** explicitly add reflexive edges to the graphs
From K to KT, ctd.

1. **Save** *Monomodal-K* as *Monomodal-KT*

2. **Idea**₁: add new rule
   
   **Rule** *NecT*
   
   *hasElement* `w` *nec* *variable* `a`
   
   *add* `w` *variable* `a`

3. **Idea**₂: add new rule
   
   **Rule** *Reflexive_edges_for_R*
   
   *isNewNode* `w`
   
   *link* `w` `w` *R*

4. Call new rule in the strategy

5. Check *P* ∧ □¬*P*, *P* ∧ □□¬*P*, ...
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
Accessibility relation $R$ is serial

- **Aim:** close all tableaux for $\Box P \land \Box \neg P$ (negation of axiom D)
- **Naive idea:** just add edges

  **Rule** `makeSerial`
  
  ```
  isNewNode w
  ```
  (match a node)

  ```
  createNewNode u
  link w u R
  ```

  $\implies$ will loop
From K to KD, ctd.

Accessibility relation $R$ is serial

- Idea: add edges only when needed and not created elsewhere
  - **Rule** `makeSerial`
    - `hasElement` $w$ nec variable $a$
    - `hasNotElement` $w$ not nec variable $b$
    - `createNewNode` $u$
    - `link` $w$ $u$ $R$

- Call **rule** `makeSerial` in the strategy

- **Check** $\Box P \land \Box \neg P$ ... $\Rightarrow$ sound but suboptimal

- avoid too many successor nodes: apply `makeSerial` only once
  - `applyOnce makeSerial`
From K to KD, ctd.

Accessibility relation $R$ is serial

- Idea: add edges only when needed and not created elsewhere
  
  **Rule** `makeSerial`
  
  `hasElement` $w$ nec variable $a$
  
  `hasNotElement` $w$ not nec variable $b$
  
  `createNewNode` $u$
  
  `link` $w$ $u$ $R$

- Call rule `makeSerial` in the strategy

- Check $\Box P \land \Box \neg P \ldots \Rightarrow$ sound but suboptimal

- Avoid too many successor nodes: apply `makeSerial` only once
  
  `applyOnce` `makeSerial`
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- $KT$
- $KD$
- $S4$
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
From KT to S4

- Accessibility relation $R$ is reflexive and transitive ($S4 = KT4$)
- Aim: close all tableaux for $\neg(\Box P \rightarrow \Box \Box P)$
  (negation of axiom 4)
- Idea$_1$: integrate reflexivity and transitivity into the truth condition
  - $M, w \vDash \Box A$ iff $M, w \vDash A$, and $M, u \vDash \Box A$ for every $u$ that is accessible from $w$ via $R$
- Idea$_2$: ...
From KT to S4, ctd.

1. Save Monomodal-KT as Monomodal-S4
2. Copy/Paste rule Nec, and rename it as Nec4
3. Idea$_1$:
   Rule Nec4
   hasElement node nec R variable a
   isLinked node node’ R

   add node’ nec R variable a
4. Check $\neg(\Box P \rightarrow \Box \Box P)$, i.e. $\Box P \land \neg\Box \Box P$
5. Test $\Box \neg \Box P$ ...
Taming S4

- Input formula $\square \neg \square P$ loops!
- Execute step-by-step (‘Step By Step’ instead of ‘Build Premodels’ button)
- Observe: if no clash wasn’t found after 2 nodes, there is no chance to find it later
  $\implies$ no need to create successors for nodes that are included in an ancestor!
  - hypothesis: nodes have been locally saturated before checking for loops
Add the rule loopTest (cf. predefined S4_Optimal)

Rule loopTest

isNewNode node’ (required only here)

isAncestor node node’

contains node node’ (or: haveSameFormulaSet)

mark node’ CONTAINED

link node’ node Loop (optional, marks the inclusion)

add condition to rule NotNec: IsNotMarked w CONTAINED

Call it in the strategy

guarantee that nodes are saturated before loopchecking:
call loopTest after the CPL rules and rule NecT

Run again...
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
From S4 to LJ

Accessibility relation $R$ is reflexive, transitive, and hereditary

- $M, w \models A \rightarrow A$ iff for all $u$ such that $wRu$, $M, u \nvdash A$ or $M, u \models A$

- Tableau method requires signed formulas
  - In LoTREC: define connectors mTrue and mFalse

- Rules:

  Rule mFalseImp
  
  hasElement w mFalse imp variable a variable b
  
  isNotMarked w CONTAINED

  createNewNode u
  
  link w u R
  
  add u mTrue variable a
  
  add u mFalse variable b

  ...
From S4 to LJ, ctd.

- Rule to propagate true atoms:

  Rule mTrueAtom
  
  hasElement w mTrue variable a
  isAtomic variable a
  isLinked w u R

  add u mTrue variable a

- Test:
  
  \(((P \to Q) \to P) \to P\)  \hfill (Pierce's formula)

- Add the other connectives...

- Test:
  
  $$\neg\neg P \to P$$
  $$P \to \neg\neg P$$
  $$P \lor \neg P$$
  ...
  
  Add the other connectives...
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- $KT$
- $KD$
- $S4$
- Intuitionistic logic $LJ$
- Model checking in LoTREC
- $PDL$
- Suggestions
Model checking

Given $M$, $w$, and $A$ do we have $M, w \models A$?

language: not, and, or, nec, pos

1. build model $M$ in LoTREC

   ```
   createNewNode w,  
   createNewNode u,  
   link w u R,  
   add u P,  
   ...
   ```

2. add formula $A$ to be checked to root note

   ```
   add w isItTrue nec P  
   ```
   (to be added as dummy connector)

3. top-down: decomposition of $A$

   ```
   hasElement w isItTrue not variable A  
   add w isItTrue variable A  
   ...
   ```

4. bottom-up: build truth value of $A$ ...
Model checking, ctd.

4. bottom-up: build truth value of $A$

hasElement $w$ isItTrue variable $A$

isAtomic variable $A$

hasElement $w$ variable $A$

markExpression $w$ isItTrue variable $A$ Yes

hasElement $w$ isItTrue nec variable $A$

isLinked $w$ $u$ $R$

isMarkedExpression $u$ isItTrue variable $A$ No

markExpression $w$ isItTrue nec variable $A$ No

hasElement $w$ isItTrue nec variable $A$

isLinked $w$ $u$ $R$

isMarkedExpressionInAllChildren $w$ isItTrue variable $A$ $R$ Yes

markExpression $w$ isItTrue nec variable $A$ Yes
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
PDL

program constructions: Kleene star, . . .
Outline

3 LoTREC
- Language
- Rules
- Strategies
- Tableau notation
- Do the algorithms do the right thing?

4 Implementing logics
- Classical logic
- Modal logic K
- Multi-modal logic $K_n$
- KT
- KD
- S4
- Intuitionistic logic LJ
- Model checking in LoTREC
- PDL
- Suggestions
It is up to you...

- S5; K +Universal operator
- Confluence
- LTL
- ...
Thank you!