Tableau-based Decision Procedures for Hybrid Logic

Gert Smolka
Saarland University

Joint work with Mark Kaminski

HyLo 2010
Edinburgh, July 10, 2010
Research Goals

- Design transparent and efficient decision procedures for expressive modal languages with nominals
- Advance the art of tableaux
- Develop efficient provers
Plan of Talk

1. Models, Formulas, Tableaux
2. Prefixed Tableaux
3. Clauses and Demos
4. Clausal Tableaux
5. Final Remarks
Graphs (nodes, edges)
Models

Graphs (nodes, edges)

Nodes are labelled with predicates ($p$, $q$, ...)

$2 \xrightarrow{x, p} 1 \xleftarrow{} 3 \xrightarrow{q} 4 \xrightarrow{p, q}$
Models

- Graphs (nodes, edges)
- Nodes are labelled with predicates \((p, q, \ldots)\)
- There are predicates called nominals that can label at most one node \((x, y, \ldots)\)
- NB: non-standard semantics of nominals
Modal Formulas

\[ s ::= p | \neg s | s \land s | \Diamond s | \Diamond^* s | Ds \]
\[ | s \lor s | \Box s | \Box^* s | \bar{D}s \]

- \( M, a \models s \) in model \( M \) node \( a \) satisfies formula \( s \)
- \( M, a \models \Diamond^* s \) there is a node reachable from \( a \) satisfying \( s \)
- \( M, a \models Ds \) there is a node different from \( a \) satisfying \( s \)

- \( \Diamond^* \) and \( \Box^* \) are called star modalities
- \( D \) and \( \bar{D} \) are called difference modalities
- Formulas containing nominals are called hybrid
- We mostly assume negation normal form (\( \neg p \))
Formulas of the form ♦∗s are called eventualities.

Eventualities cause non-compactness:
♦∗¬p, p, □p, □□p, ...

Difference modalities can express global modalities and nominals:
- Every node satisfies s: s ∧ Ds
- Some node satisfies s: s ∨ Ds
- At most one node satisfies s: D¬s ∨ D dbname ¬s
Complexity of Satisfiability

- Formula $s$ is satisfiable if $M, a \models s$ for some $M$ and $a$
- $K$ is PSPACE-complete
- $H$ is PSPACE-complete
- $K$ with $\Box^*$ is EXP-complete ($\approx$ ALC)
- $H$ with $\Box^*$ and $\Diamond^*$ is EXP-complete (hybrid $\mu$-calculus)
Wanted: Constructive Decision Procedures

Given a formula $s$,
- return a finite model of $s$ if $s$ is satisfiable
- return “unsatisfiable” if $s$ is unsatisfiable

Procedures should
- elegant (e.g., transparent correctness proof)
- be practical (goal-directed, incremental),
  see reasoners for description logics
Method: Tableau Systems

- A branch is a finite, nonempty set $\Gamma$ of formulas
- $M \models \Gamma$ iff $\forall s \in \Gamma \exists a. M, a \models s$
Method: Tableau Systems

- A **branch** is a finite, nonempty set $\Gamma$ of formulas
- $M \models \Gamma$ iff $\forall s \in \Gamma \exists a. \ M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
Method: Tableau Systems

A branch is a finite, nonempty set $\Gamma$ of formulas

$M \models \Gamma$ iff $\forall s \in \Gamma \exists a. M, a \models s$

Expansion rules add formulas to branch such that satisfiability is preserved

Closing rules identify unsatisfiable branches
A branch is a finite, nonempty set $\Gamma$ of formulas

$M \models \Gamma$ iff $\forall s \in \Gamma \exists a. M, a \models s$

Expansion rules add formulas to branch such that satisfiability is preserved

Closing rules identify unsatisfiable branches
A **branch** is a finite, nonempty set $\Gamma$ of formulas

$M \models \Gamma$ iff $\forall s \in \Gamma \exists a. M, a \models s$

Expansion rules add formulas to branch such that satisfiability is preserved

Closing rules identify unsatisfiable branches
Method: Tableau Systems

- A branch is a finite, nonempty set $\Gamma$ of formulas
- $M \models \Gamma$ iff $\forall s \in \Gamma \exists a. M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
- Closing rules identify unsatisfiable branches
Method: Tableau Systems

- A branch is a finite, nonempty set $\Gamma$ of formulas
- $M \models \Gamma$ iff $\forall s \in \Gamma \exists a. M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
- Closing rules identify unsatisfiable branches
- A branch is evident if no rules applies to it
Correctness of Tableau Systems

- **Termination**   Tableau construction terminates
- **Soundness**   Satisfiable branches are either evident or have a satisfiable expansion
- **Completeness**   Evident branches are finitely satisfiable
- **Correct tableau system** describes a tableau construction procedure that yields a constructive decision procedure
- **Nondeterminism**   There may be many complete tableaux for a given initial branch; may differ in size; each of them decides satisfiability of initial branch
Which formulas?
Which notion of evidence?
Which rules?
II Prefixed Tableaux

- Originated with Kripke 1963

- Previous work on prefixed tableaux for hybrid logic
  - Horrocks and Sattler, JAR 2007

- Our work (Kaminski and Smolka) considers hybrid logic with difference modalities, graded modalities, star modalities and transitive relations
  - Spartacus prover for H with global modalities: M4M 2009, ENTCS 2010

- Here: H with □*, D, ŠD
Prefixed Tableaux

Prefixed Formulas

\[ x : s \]

- \( x \) is a prefix, \( s \) is a modal formula
- Prefixes name the nodes of the model to be constructed
- We represent prefixes as nominals
- \( M \models x : s \) iff \( M \) has a node labeled with \( x \) that satisfies \( s \)
- Invariant for tableau expansion: All modal formulas are subformulas of the initial modal formulas
- Prefixed tableau system terminates if number of prefixes can be bounded
Prefixed Tableaux

Four Kinds Prefixed Formulas

\[ x : s \]
\[ rxy \]
\[ x = y \]
\[ x \neq y \]

- Branch is a set of prefixed formulas
- A model satisfies a branch if it satisfies every formula of the branch
- A model satisfies a modal formula if it has a node that satisfies the formula
Prefixed Tableaux

Four Kinds Prefixed Formulas

\[ x : s \leadsto x \land s \]
\[ r_{xy} \leadsto x \land \Diamond y \]
\[ x = y \leadsto x \land y \]
\[ x \neq y \leadsto x \land \neg y, \ y \land \neg x \]

- Branch is a set of prefixed formulas
- A model satisfies a branch if it satisfies every formula of the branch
- A model satisfies a modal formula if it has a node that satisfies the formula
- Hybrid logic can internalize prefixed formulas
Four Kinds Prefixed Formulas

\[
\begin{align*}
x : s & \leadsto x \land s \\
rxy & \leadsto x \land \Diamond y \\
x = y & \leadsto x \land y \\
x \neq y & \leadsto x \land \neg y, y \land \neg x
\end{align*}
\]

- Branch is a set of prefixed formulas
- A model satisfies a branch if it satisfies every formula of the branch
- A model satisfies a modal formula if it has a node that satisfies the formula
- Hybrid logic can internalize prefixed formulas
- Prefixes simplify formulation and analysis of tableau system
# Tableau Rules for K with \( \square^* \)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Premise 1</th>
<th>Premise 2</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x : s ), ( x : \neg s )</td>
<td>closed</td>
<td>( x : s \land t )</td>
<td>( x : s \lor t )</td>
</tr>
<tr>
<td>( x : \square s ), ( rxy )</td>
<td>( y : s )</td>
<td>( x : \square^* s )</td>
<td>( x : \diamond s )</td>
</tr>
<tr>
<td>( x : s ), ( x : \square^* s )</td>
<td>( rxy ), ( y : s )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( y \) fresh
Tableau Rules for K with $\Box^*$

\[
\begin{align*}
\frac{x : s, \ x : \neg s}{\text{closed}} \\
\frac{x : s \land t}{x : s, \ x : t} \\
\frac{x : s \lor t}{x : s \mid x : t}
\end{align*}
\]

\[
\begin{align*}
\frac{x : \Box s, \ rxy}{y : s} \\
\frac{x : \Box^* s}{x : s, \ x : \Box \Box^* s} \\
\frac{x : \Diamond s}{rxy, \ y : s}
\end{align*}
\]

- Diamond rule is blocked if evidence condition for $x : s$ is satisfied
Prefixed Tableaux

Tableau Rules for K with $\Box^*$

\[
\begin{align*}
\frac{x : s \land \neg s}{\text{closed}} & \quad \frac{x : s \land t}{x : s, x : t} & \frac{x : s \lor t}{x : s \mid x : t} \\
\frac{x : \Box s \land rxy}{y : s} & \quad \frac{x : \Box^* s}{x : s, x : \Box \Box^* s} & \frac{x : \Diamond s}{rxy, y : s} \quad \text{y fresh}
\end{align*}
\]

- Diamond rule is blocked if evidence condition for $x : s$ is satisfied

\[
\frac{x : \Diamond s \land \Box s_1, \ldots, x : \Box s_n}{r y z, z : s, y : \Box s_1, \ldots, y : \Box s_n}
\]
Prefixed Tableaux

**Tableau Rules for K with □**

\[
\begin{align*}
x : s, \ x : \neg s & \quad \text{closed} \\
x : s \land t & \quad x : s, \ x : t \\
x : s \lor t & \quad x : s \mid x : t \\
x : \Box s, \ rxy & \quad y : s \\
x : \Box^* s & \quad x : s, \ x : \Box \Box^* s \\
x : \Diamond s & \quad rxy, \ y : s \quad \text{y fresh}
\end{align*}
\]

- Diamond rule is blocked if evidence condition for \( x : s \) is satisfied

\[
\begin{align*}
x : \Diamond s, \ x : \Box s_1, \ldots, \ x : \Box s_n & \quad ryz, \ z : s, \ y : \Box s_1, \ldots, \ y : \Box s_n
\end{align*}
\]

- Ensures termination since there are only finitely many patterns \( \Diamond s, \ \Box s_1, \ldots, \ \Box s_n \)
Tableau Rules for K with □*

- $x : s$, $x : \neg s$
  - closed

- $x : s \land t$
  - $x : s$, $x : t$

- $x : s \lor t$
  - $x : s \mid x : t$

- $x : \Box s$, $rxy$
  - $y : s$

- $x : \Box^* s$
  - $x : s$, $x : \Box \Box^* s$

- $x : \Diamond s$
  - $rxy$, $y : s$

  y fresh

- **Diamond rule is blocked if evidence condition for $x : s$ is satisfied**

  - $x : \Diamond s$, $x : \Box s_1$, \ldots, $x : \Box s_n$
  - $rzy$, $z : s$, $y : \Box s_1$, \ldots, $y : \Box s_n$

  - Ensures termination since there are only finitely many patterns $\Diamond s, \Box s_1, \ldots, \Box s_n$

  - **Pattern-based blocking** [HyLo 2007], implemented in Spartacus
Model Construction

- Construct model for evident branch

\[
\begin{align*}
  x : \Diamond s, & \quad x : \square s_1, \ldots, \quad x : \square s_n \\
  ryz, & \quad z : s, \quad y : \square s_1, \ldots, \quad y : \square s_n \\
  \hline
  x : \square s, & \quad rxy \\
  \hline
  y : s
\end{align*}
\]

- Nodes = prefixes of evident branch
- Edges = pairs \((x, y)\) such that \(\forall s. \ x : \square s \ \Rightarrow \ y : s\) (i.e., all edges that respect box formulas of branch)
Extension to Nominals

- A prefixed formula $x : y$ is an equational constraint $x = y$
- Work with nominal equivalence, that is, least equivalence relation $\sim$ such that $x \sim y$ if $x : y$ or $x = y$ on the branch
- Lift tableau rules to equivalence classes

\[
\begin{array}{c}
\begin{array}{c}
\tilde{x} : s, \tilde{x} : \neg s \\
\text{closed}
\end{array} & \\
\begin{array}{c}
\tilde{x} : s \land t \\
x : s, x : t
\end{array} & \ldots
\end{array}
\]

- One additional rule $\tilde{x} : \neg x$
- closed

- Model construction
  - Nodes = equivalence classes of prefixes
  - Edges = $(\tilde{x}, \tilde{y})$ such that $\forall s. \tilde{x} : \Box s \Rightarrow \tilde{y} : s$
- Straightforward implementation, see Spartacus
Rules for Difference Modalities

\[
\begin{align*}
\frac{x : Ds}{y : s, y \neq x} & \quad \text{y fresh} \\
\frac{x : Ds}{y : s, y \sim x} & \quad \text{closed} \\
x \not= y & \quad x \sim y
\end{align*}
\]

\[
\frac{x : \bar{D}s}{y = x \mid y : s} \quad \text{forall prefixes y on branch}
\]

- Nominal equivalence $\sim$ essential for evidence condition for D
- Disequations $y \neq x$ are essential for termination
- At most two fresh prefixes per formula Ds
- Equations $y = x$ are essential for soundness
III Clauses and Demos

- Foundation for prefix-free decision procedures [IJCAR 2010]
- Here we consider $H^*$ ($H$ with $\Box^*$ and $\Diamond^*$).
- Extends to hybrid PDL and difference modalities
DNF

\[ s \equiv \bigvee \left( \bigwedge \text{literal} \right) \]

\[ \text{literal} := p \mid \neg p \mid \lozenge s \mid \Box s \]

\[ \lozenge^* s \equiv s \lor \lozenge \lozenge^* s \]

\[ \Box^* s \equiv s \land \Box \Box^* s \]

- **Clause**: set of literals, no complementary pair \( p, \neg p \)
- Every formula can be represented as a set of clauses
- NB: Clauses are interpreted conjunctively
We assume a **DNF procedure** $\mathcal{D}$ that, given a set of formulas $A$, yields a set of clauses $\mathcal{D}A$ such that

$$\bigwedge_{s \in A} s \equiv \bigvee_{C \in \mathcal{D}A} \bigwedge_{s \in C} s$$

**DNF procedure provides local propositional reasoning**
Request of a Clause

\[ \mathcal{RC} := \{ s \mid \Box s \in C \} \]

- If a node satisfies \( C \),
  then every successor of the node must satisfy \( \mathcal{RC} \).
- If a node satisfies \( C \) and \( \Diamond s \in C \),
  then the node must have a successor that satisfies a clause \( D \in D(\mathcal{RC}; s) \).
Demos

- Demos are syntactic models
- Nodes of demos are clauses such that $\Delta, C \models C$
- Edges of demos are described as links $CsD$ that identify the literal $\Diamond s \in C$ they satisfy
Example: Construction of a Demo

◊◊∗p, □¬p, □□¬p
Example: Construction of a Demo

Note: \( \diamondsuit * p \equiv p \lor \diamondsuit \diamondsuit * p \)
Example: Construction of a Demo

Note: $\Diamond^* p \equiv p \lor \Diamond \Diamond^* p$
Example: Construction of a Demo

Note: \( \Diamond^* p \equiv p \lor \Diamond \Diamond^* p \)
Links

- **Minimal link**: Triple $CsD$ such that 
  $\diamond s \in C$ and $D \in D(RC; s)$

- **Lifted Link**: Triple $CsD$ such that 
  $CsD'$ is minimal link for some $D' \subsetneq D$

- Lifted links are needed to accommodate nominals
Definition of Demos

- A demo is a finite, nonempty set of clauses and links such that

\[
\begin{align*}
\Box s \in C & \quad \frac{CsD}{C,D} & \quad x \in C, x \in D & \quad \frac{C = D}{\Box \Box s \in C} \\
\end{align*}
\]

\[
\Box \Box \Box s \in C
\]

\[
\Box \Box s \text{-path from } C \text{ to } D \text{ such that } D \supset s
\]

- \( D \supset s :\Leftrightarrow \exists C \in D\{s\}. \ C \subseteq D \quad D \text{ supports } s \)
- A demo is a model (nodes = clauses, edges = links)
- A demo \( \Delta \) satisfies \( \Delta, C \models C \) for all nodes / clauses
Finite Supply of Literals

- When we construct a demo for a formula $s$, it suffices to consider a finite set $\mathcal{L}s$ of literals that can be computed in linear time; this leaves us with a finite search space.
- A literal base is a finite set $\mathcal{L}$ of literals closed under taking minimal links:
  \[
  \forall C \subseteq \mathcal{L} \ \forall \diamond s \in C \ \forall D \in \mathcal{D}(RC; s). \ D \subseteq \mathcal{L}
  \]
- For every formula $s$ one can obtain in linear time a literal base $\mathcal{L}s$ containing the clauses of $\mathcal{D}\{s\}$.
- $\mathcal{L}s$ basically consists of the literals occurring as subformulas in $s$. 
Demo Theorem

For every satisfiable formula \( s \) there exists a demo satisfying \( s \) that employs only literals from \( \mathcal{L}s \).

- Small model theorem
- Yields naive decision procedure
- Proof for \( K^* \)
  - Let \( M \) be model of \( s \)
  - All clauses \( C \subseteq \mathcal{L}s \) satisfied by \( M \)
  - All links between these clauses
IV Clausal Tableaux

- Take clauses and links as formulas
- Construct demos
- Here: Clausal decision procedure for H* [IJCAR 2010]
- Extends to hybrid PDL

- The term “clausal tableaux” has been used before for a rather different approach by Nguyen and Goré [1999, 2009]
Clausal Tableaux for K*

- A **branch** is a finite, nonempty set of clauses and links such that:

\[
\frac{CsD}{C, D}
\]

\[
\frac{CsD, CsD'}{D = D'}
\]
Clausal Tableaux for $K^*$

- A **branch** is a finite, nonempty set of clauses and links such that:

  \[
  \frac{CsD}{C, D}
  \]

  \[
  \frac{CsD, CsD'}{D = D'}
  \]

- Tableaux rules

  \[
  \frac{\Diamond s \in C}{CsD, D | \cdots}
  \]

  \[
  D \in \mathcal{D}(RC; s)
  \]

  \[
  \Diamond s \in C \quad \mathcal{D}(RC; s) = \emptyset
  \]

  \[
  \frac{\operatorname{closed}}{}
  \]
Clausal Tableaux for K*

- A **branch** is a finite, nonempty set of clauses and links such that:

\[
\frac{CsD}{C, D}
\]

\[
\frac{CsD, CsD'}{D = D'}
\]

- **Tableaux rules**

\[
\frac{\diamond s \in C}{CsD, D | \cdots}
\]

\[
D \in \mathcal{D}(RC; s)
\]

\[
\frac{\diamond s \in C}{\text{closed}}
\]

\[
\mathcal{D}(RC; s) = \emptyset
\]

**Bad loop rule**

\[
C_1 \rightarrow^s \cdots \rightarrow^s C_n \rightarrow^s C_1
\]

\[
\forall i \in [1, n]. \ C_i \not\models s
\]

\[
\text{closed}
\]

where \( C \rightarrow^s D \) means that \( CsD \) is on branch
Correctness ($K^*$)

- Termination straightforward since all clauses are subsets of initial literal base
- Completeness straightforward since evident branches are demos (bad loop rule guarantees satisfaction of eventualities)
- **Soundness challenging** since one needs a semantics for star links that justifies bad loop rule

**Example**
- $C = \{\Box \Diamond^* p\}$ is satisfiable clause
- $\{C, C(\Diamond^* p)C\}$ is closed branch
- Link $C(\Diamond^* p)C$ must be unsatisfiable
Minimal Distance Semantics for Star Links

\[ \delta_M As := \text{minimal distance from a node satisfying } A \]
\[ \text{to a node satisfying } s \]

- \( M \) satisfies \( C(\diamond^*s)D \) if
  - \( \delta_M Cs > 0 \) \( \Rightarrow \) \( \delta_M Cs > \delta_M Ds \)
  - \( \delta_M Ds = 0 \) \( \Rightarrow \) \( D \triangleright s \)

- Link must reduce minimal distance to \( s \)
- Link must deliver (i.e., \( D \triangleright s \)) if minimal distance is 0
- Minimal distance idea appears in [Baader 1990]
Clausal Tableaux

Clausal Tree Tableaux for H*, Example

\[\Diamond\Diamond^* p, \neg p, \Box (x \land \neg p), \Diamond \Box \neg p\]
Clausal Tree Tableaux for H*, Example

$x, ♦♦*p, ¬p$

♦♦*p, ¬p, □(x ∧ ¬p), ♦□¬p
Clausal Tree Tableaux for $H^*$, Example

$\Diamond \Diamond^* p, \neg p, \Box(x \land \neg p), \Diamond \Box \neg p$

$\Diamond^* p$

$x, \Diamond \Diamond^* p, \neg p$

$\Diamond^* p$

$p$
Clausal Tableaux

Clausal Tree Tableaux for H*, Example

\[ \Diamond \Diamond^* p, \neg p, \Box (x \land \neg p), \Diamond \Box \neg p \]

\[ x, \Diamond \Diamond^* p, \neg p \]

\[ x, \Diamond \Diamond^* p, \neg p, \Box \neg p \]

\[ p \]
Clausal Tree Tableaux for $H^*$, Example

\[\Diamond \Diamond \Diamond \Diamond \Diamond p, \neg p, \Box (x \land \neg p), \Diamond \Box \neg p\]

\[x, \Diamond \Diamond \Diamond \Diamond p, \neg p\]

\[\neg p\]

\[\Diamond \Diamond \Diamond \Diamond p, \neg p, \Box \neg p\]
Clausal Tree Tableaux for $H^*$, Example

\[ \Diamond \Diamond \Diamond \Diamond p, \neg, \Box(x \land \neg \ p), \Diamond \Box \neg \ p \]

\[ x, \Diamond \Diamond \Diamond \Diamond p, \neg \]

\[ p \]

\[ x, \Diamond \Diamond \Diamond \Diamond p, \neg p, \Box \neg \]

\[ \Diamond \Diamond \Diamond \Diamond p, \neg \]

\[ \Diamond \Diamond \Diamond \Diamond \Diamond p, \neg \]
Clausal Tableaux for H*, Example
Clausal Tableaux

Clausal Tree Tableaux for H*, Example

Demo consists of nominally maximal clauses
Clausal Tree Tableaux for H*

- Nominal completion

\[ C^\Gamma := C \cup \{ s \mid \exists x \in C \ \exists D \in \Gamma. \ x \in D \land s \in D \} \]

- Require branches to be nominally coherent

\[ \frac{C}{C^\Gamma} \]

- Ignore clauses that aren’t nominally maximal (i.e., \( C = C^\Gamma \))

- See link \( CsD \) as link \( CsD^\Gamma \) (link lifting)

\[ C \xrightarrow{s} D :\Leftrightarrow \exists E. \ CsE \in \Gamma \land E^\Gamma = D \]
Tableau Rules for $H^*$

\[
\begin{align*}
\Diamond s \in C & \quad \Rightarrow \quad CsD^{\Gamma}, D^{\Gamma} \mid \ldots \\
\forall D \in \mathcal{D}(\mathcal{R}C; s). \quad D^{\Gamma} \text{ not a clause} & \\
C_1 \rightarrow^{s} \ldots \rightarrow^{s} C_n \rightarrow^{s} C_1 & \quad \Rightarrow \quad \forall i \in [1, n]. \quad C_i \not\models s
\end{align*}
\]
Correctness (H*)

- Termination: As for K*
- Soundness: As for K*, we have $\delta_M Cs = \delta_M C^\Gamma s$
- Completeness: Take clauses $C$ with $C = C^\Gamma$
V Final Remarks
Complexity

\[ n \] : size of initial formula

\[ n \] : number of literals to be considered

\[ 2^n \] : number of clauses to be considered

\[ 2^{2n} \] : number of branches to be considered

- \( H^* \) satisfiability is in Exp
- Must not construct complete tableaux in tree representation
- Must avoid recomputation at clause level
- Switch to graph representation to stay in EXP
  - [Pratt 1980] PDL
  - [Goré and Widmann, IJCAR 2010] PDL with converse
Graph Representation and Nominals

- Graph representation is straightforward for K* if eventuality checking is done at end.
- Yields EXPTIME decision procedure.
- Nominals cause severe complications, no good solution so far.
- Satisfiability of clause must be determined under nominal assumptions and may depend on nominal assumptions.
Main Contributions

- Pattern-based blocking for prefixed tableaux
- Terminating prefixed tableaux for difference modalities
- Clauses and demos
- Decision procedure for $H^*$