Tableau-based Decision Procedures for Hybrid Logic

Gert Smolka
Saarland University

Joint work with Mark Kaminski

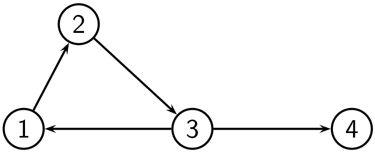
HyLo 2010
Edinburgh, July 10, 2010

Research Goals

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

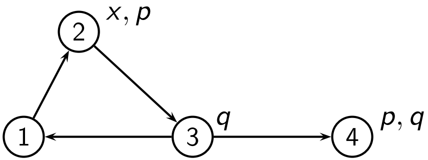
Plan of Talk

Models



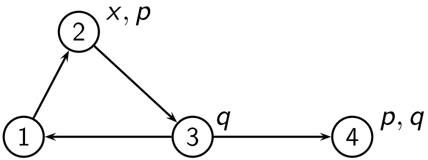
- Graphs (nodes, edges)

Models



- Graphs (nodes, edges)
- Nodes are labelled with predicates (p, q, ...)

Models



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

Modal Formulas

$$s \ ::= \ p \mid \neg s \mid s \wedge s \mid \Diamond s \mid \Diamond^* s \mid \mathrm{D} s$$
$$\mid s \vee s \mid \Box s \mid \Box^* s \mid \bar{\mathrm{D}} s$$

- $M, a \models s$ in model M node a satisfies formula s
- $M, a \models \Diamond^* s$ there is a node reachable from a satisfying s
- $M, a \models \mathrm{D} s$ there is a node different from a satisfying s
- $\Diamond^*$ and $\Box^*$ are called star modalities
- D and $\bar{\mathrm{D}}$ are called difference modalities
- Formulas containing nominals are called hybrid
- We mostly assume negation normal form $(\neg p)$

- Formulas of the form $\Diamond^* s$ are called eventualities
- Eventualities cause non-compactness:
  $\Diamond^* \neg p,\ p,\ \Box p,\ \Box\Box p,\ \ldots$
- Difference modalities can express global modalities and nominals
  - Every node satisfies s: $s \wedge \bar{D}s$
  - Some node satisfies s: $s \vee Ds$
  - At most one node satisfies s: $\bar{D}\neg s \vee D\bar{D}\neg s$

Complexity of Satisfiabilty

- 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 μ-calculus)

Wanted: Constructive Decision Procedures

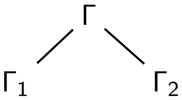
- Given a formula *s*,
  - return a finite model of *s* if *s* is satisfiable
  - return "unsatisfiable" if *s* is unsatisfiable
- Procedures should
  - elegant (e.g., transparent correctness proof)
  - be practical (goal-directed, incremental), see reasoners for description logics

Method: Tableau Systems

Γ

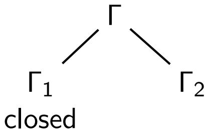
- A branch is a finite, nonempty set Γ of formulas
- $M \models \Gamma$ iff $\forall\, s \in \Gamma\ \exists\, a.\ M, a \models s$

Method: Tableau Systems



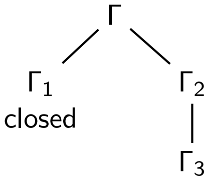
- A branch is a finite, nonempty set Γ of formulas
- $M \models \Gamma$ iff $\forall\, s \in \Gamma\ \exists\, a.\ M, a \models s$
- Expansion rules add formulas to branch
  such that satisfiability is preserved

Method: Tableau Systems



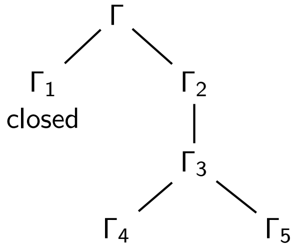
- A branch is a finite, nonempty set Γ of formulas
- $M \models \Gamma$ iff $\forall\, s \in \Gamma\ \exists\, a.\ M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
- Closing rules identify unsatisfiable branches

Method: Tableau Systems



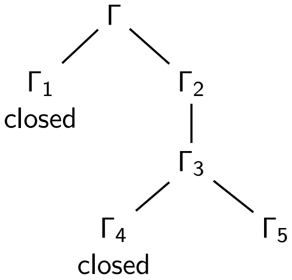
- A branch is a finite, nonempty set Γ of formulas
- $M \models \Gamma$ iff $\forall\, s \in \Gamma\ \exists\, a.\ M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
- Closing rules identify unsatisfiable branches

Method: Tableau Systems



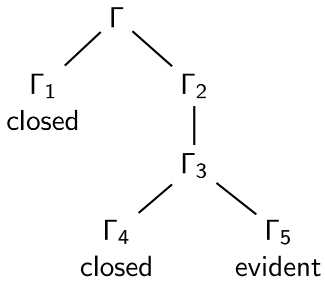
- A branch is a finite, nonempty set Γ of formulas
- $M \models \Gamma$ iff $\forall\, s \in \Gamma\ \exists\, a.\ M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
- Closing rules identify unsatisfiable branches

Method: Tableau Systems



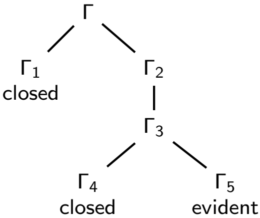
- A branch is a finite, nonempty set Γ of formulas
- $M \models \Gamma$ iff $\forall\, s \in \Gamma\ \exists\, a.\ M, a \models s$
- Expansion rules add formulas to branch such that satisfiability is preserved
- Closing rules identify unsatisfiable branches

Method: Tableau Systems



- A branch is a finite, nonempty set Γ 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

Design Space for Tableau Systems

- Which formulas?
- Which notion of evidence?
- Which rules?

II Prefixed Tableaux

- Originated with Kripke 1963
- Previous work on prefixed tableaux for hybrid logic
  - Bolander and Braüner, J. Log. Comput. 2006
  - Bolander and Blackburn, J. Log. Comput. 2007
  - Horrocks and Sattler, JAR 2007
- Our work (Kaminski and Smolka) considers hybrid logic with difference modalities, graded modalities, star modalities and transitive relations
  - HyLo 2007, M4M 2007, IJCAR 2008, JoLLI 2009, Tableaux 2009, TCS 2010
  - Spartacus prover for H with global modalities: M4M 2009, ENTCS 2010
- Here: H with $\Box^*$, D, $\bar{\mathrm{D}}$

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

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

Four Kinds Prefixed Formulas

$$x : s \quad \rightsquigarrow \quad x \wedge s$$
$$rxy \quad \rightsquigarrow \quad x \wedge \Diamond y$$
$$x = y \quad \rightsquigarrow \quad x \wedge y$$
$$x \neq y \quad \rightsquigarrow \quad x \wedge \neg y, \; y \wedge \neg x$$

- 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

$$
\begin{aligned}
x : s &\quad\rightsquigarrow\quad x \wedge s \\
rxy &\quad\rightsquigarrow\quad x \wedge \Diamond y \\
x = y &\quad\rightsquigarrow\quad x \wedge y \\
x \neq y &\quad\rightsquigarrow\quad x \wedge \neg y,\; y \wedge \neg x
\end{aligned}
$$

- 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 $\Box^*$

$$\frac{x : s \;,\; x : \neg s}{\text{closed}} \qquad \frac{x : s \wedge t}{x : s \;,\; x : t} \qquad \frac{x : s \vee t}{x : s \;\mid\; x : t}$$

$$\frac{x : \Box s \;,\; rxy}{y : s} \qquad \frac{x : \Box^* s}{x : s \;,\; x : \Box\Box^* s} \qquad \frac{x : \Diamond s}{rxy \;,\; y : s}\; y \text{ fresh}$$

Tableau Rules for K with $\Box^*$

$$\frac{x : s \;,\; x : \neg s}{\text{closed}} \qquad \frac{x : s \wedge t}{x : s \;,\; x : t} \qquad \frac{x : s \vee t}{x : s \;\mid\; x : t}$$

$$\frac{x : \Box s \;,\; rxy}{y : s} \qquad \frac{x : \Box^* s}{x : s \;,\; x : \Box\Box^* s} \qquad \frac{x : \Diamond s}{rxy \;,\; y : s} \; y \text{ fresh}$$

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

Tableau Rules for K with $\Box^*$

$$\frac{x : s \;,\; x : \neg s}{\text{closed}} \qquad \frac{x : s \wedge t}{x : s \;,\; x : t} \qquad \frac{x : s \vee t}{x : s \;\mid\; x : t}$$

$$\frac{x : \Box s \;,\; rxy}{y : s} \qquad \frac{x : \Box^* s}{x : s \;,\; x : \Box\Box^* s} \qquad \frac{x : \Diamond s}{rxy \;,\; y : s} \;\; y \text{ fresh}$$

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

$$\frac{x : \Diamond s \;,\; x : \Box s_1 \;,\ldots,\; x : \Box s_n}{ryz \;,\; z : s \;,\; y : \Box s_1 \;,\ldots,\; y : \Box s_n}$$

Tableau Rules for K with $\Box^*$

$$\frac{x : s \;,\; x : \neg s}{\text{closed}} \qquad \frac{x : s \wedge t}{x : s \;,\; x : t} \qquad \frac{x : s \vee t}{x : s \;\mid\; x : t}$$

$$\frac{x : \Box s \;,\; rxy}{y : s} \qquad \frac{x : \Box^* s}{x : s \;,\; x : \Box\Box^* s} \qquad \frac{x : \Diamond s}{rxy \;,\; y : s} \quad y \text{ fresh}$$

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

$$\frac{x : \Diamond s \;,\; x : \Box s_1 \;,\; \ldots,\; x : \Box s_n}{\overline{ryz \;,\; z : s \;,\; y : \Box s_1 \;,\; \ldots,\; y : \Box s_n}}$$

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

Tableau Rules for K with $\Box^*$

$$\frac{x : s \;,\; x : \neg s}{\text{closed}} \qquad \frac{x : s \wedge t}{x : s \;,\; x : t} \qquad \frac{x : s \vee t}{x : s \;\mid\; x : t}$$

$$\frac{x : \Box s \;,\; rxy}{y : s} \qquad \frac{x : \Box^* s}{x : s \;,\; x : \Box\Box^* s} \qquad \frac{x : \Diamond s}{rxy \;,\; y : s} \;\; y \text{ fresh}$$

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

$$\frac{x : \Diamond s \;,\; x : \Box s_1 \;,\ldots,\; x : \Box s_n}{ryz \;,\; z : s \;,\; y : \Box s_1 \;,\ldots,\; y : \Box s_n}$$

- 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

$$\frac{x : \Diamond s \;,\; x : \Box s_1 \;,\ldots,\; x : \Box s_n}{ryz \;,\; z : s \;,\; y : \Box s_1 \;,\ldots,\; y : \Box s_n} \qquad \frac{x : \Box s \;,\; rxy}{y : s}$$

- Nodes = prefixes of evident branch
- Edges = pairs (x, y) such that $\forall s.\ x : \Box s \;\Rightarrow\; y : s$
  (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

$$\frac{\tilde{x} : s \ , \ \tilde{x} : \neg s}{\text{closed}} \qquad \frac{\tilde{x} : s \wedge t}{x : s \ , \ x : t} \qquad \ldots$$

- One additional rule $\dfrac{\tilde{x} : \neg x}{\text{closed}}$
- Model construction
  - Nodes = equivalence classes of prefixes
  - Edges = $(\tilde{x}, \tilde{y})$ such that $\forall s.\ \tilde{x} : \Box s \ \Rightarrow \ \tilde{y} : s$
- Straigthforward implementation, see Spartacus

Rules for Difference Modalities

$$\frac{x : \mathsf{D}s}{y : s \;\;,\;\; y \neq x}\; y \text{ fresh} \qquad \frac{x : \mathsf{D}s}{\overline{\overline{y : s \;\;,\;\; y \nsim x}}} \qquad \frac{x \neq y}{\text{closed}}\; x \sim y$$

$$\frac{x : \bar{\mathsf{D}}s}{y = x \;\mid\; y : s}\; \text{forall prefixes } y \text{ on branch}$$

- 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

$$s \equiv \bigvee \Big(\bigwedge \text{literal} \Big)$$

$$\text{literal} \;:=\; p \mid \neg p \mid \Diamond s \mid \Box s$$

$$\Diamond^* s \equiv s \vee \Diamond\Diamond^* s$$

$$\Box^* s \equiv s \wedge \Box\Box^* s$$

- 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

DNF Procedure

- 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

$$\mathcal{R}C \;:=\; \{\, s \mid \Box s \in C \,\}$$

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

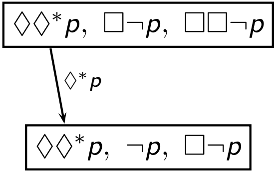
Demos

- 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

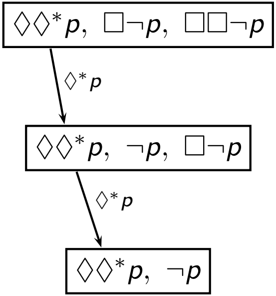
$$\boxed{\Diamond\Diamond^* p,\ \Box\neg p,\ \Box\Box\neg p}$$

Example: Construction of a Demo



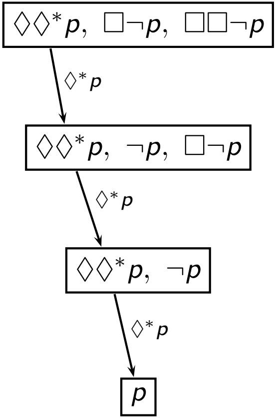
Note: $\Diamond^* p \equiv p \vee \Diamond\Diamond^* p$

Example: Construction of a Demo



Note: $\Diamond^* p \equiv p \vee \Diamond\Diamond^* p$

Example: Construction of a Demo



Note: $\Diamond^* p \equiv p \vee \Diamond\Diamond^* p$

Links

- Minimal link: Triple CsD such that
  $\Diamond s \in C$ and $D \in \mathcal{D}(\mathcal{R}C; s)$
- 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

$$\frac{\Diamond s \in C}{CsD} \qquad \frac{CsD}{C\,,D} \qquad \frac{x \in C\,, x \in D}{C = D}$$

$$\frac{\Diamond\Diamond^* s \in C}{\Diamond^* s\text{-path from } C \text{ to } D \text{ such that } D \triangleright s}$$

- $D \triangleright s \;:\Leftrightarrow\; \exists C \in \mathcal{D}\{s\}.\ C \subseteq D$ $\qquad$ D supports s
- A demo is a model (nodes = clauses, edges = links)
- A demo Δ 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 finite set $\mathcal{L}$ of literals closed under taking minimal links:

$$\forall\, C \subseteq \mathcal{L} \;\; \forall\, \Diamond s \in C \;\; \forall\, D \in \mathcal{D}(\mathcal{R}C; s). \;\; D \subseteq \mathcal{L}$$

- 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:

$$\frac{CsD}{C\,,\,D} \qquad\qquad \frac{CsD\,,\,CsD'}{D = D'}$$

- Tableaux rules

$$\frac{\Diamond s \in C}{CsD\,,\,D \;\;\big|\;\; \cdots}\;\; D \in \mathcal{D}(\mathcal{R}C; s) \qquad\qquad \frac{\Diamond s \in C}{\text{closed}}\;\; \mathcal{D}(\mathcal{R}C; s) = \emptyset$$

Clausal Tableaux for K*

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

$$\frac{CsD}{\overline{C\,,\,D}} \qquad\qquad \frac{CsD\,,\,CsD'}{\overline{D = D'}}$$

- Tableaux rules

$$\frac{\Diamond s \in C}{CsD\,,\,D \;\mid\; \cdots}\; D \in \mathcal{D}(\mathcal{R}C; s) \qquad\qquad \frac{\Diamond s \in C}{\text{closed}}\; \mathcal{D}(\mathcal{R}C; s) = \emptyset$$

Bad loop rule $\dfrac{C_1 \overset{\Diamond^* s}{\rightarrow} \cdots \overset{\Diamond^* s}{\rightarrow} C_n \overset{\Diamond^* s}{\rightarrow} C_1}{\text{closed}}\; \forall i \in [1, n].\; C_i \not\rhd s$

where $C \overset{s}{\rightarrow} D$ 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
  - $C = \{\Diamond\Diamond^* p\}$ 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

$\delta_M As$:= minimal distance from a node satisfying A to a node satisfying s

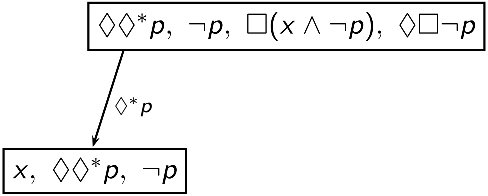
- M satisfies $C(\Diamond^* s)D$ if
  - $\delta_M Cs > 0 \;\Rightarrow\; \delta_M Cs > \delta_M Ds$
  - $\delta_M Ds = 0 \;\Rightarrow\; D \triangleright s$

- 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]

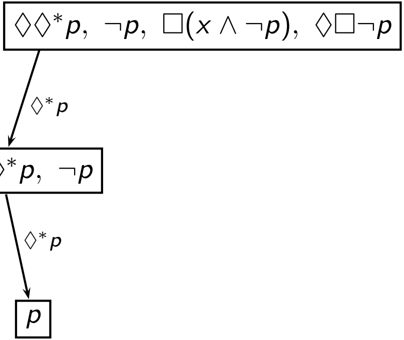
Clausal Tree Tableaux for H*, Example

$$\boxed{\Diamond\Diamond^* p,\ \neg p,\ \Box(x \wedge \neg p),\ \Diamond\Box\neg p}$$

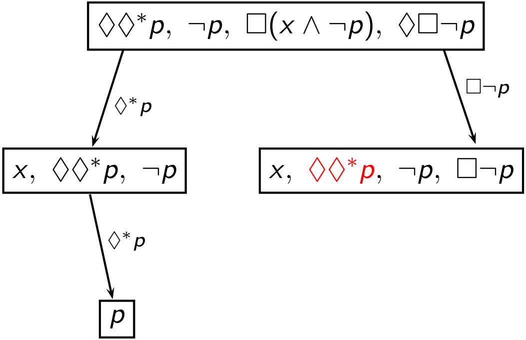
Clausal Tree Tableaux for H*, Example



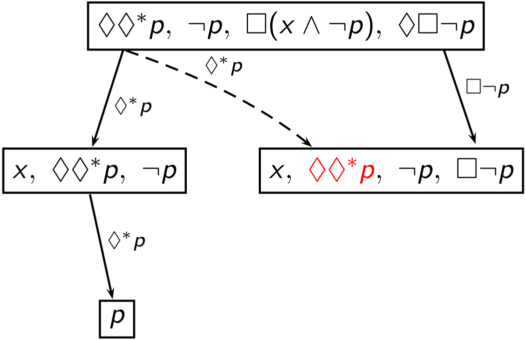
Clausal Tree Tableaux for H*, Example



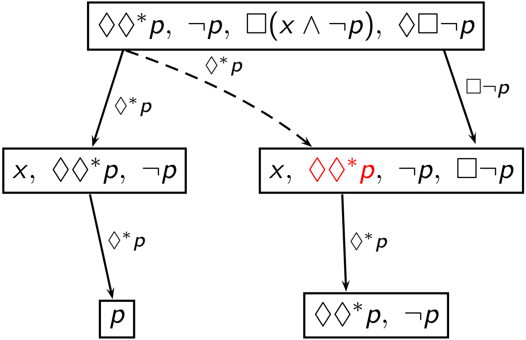
Clausal Tree Tableaux for H*, Example



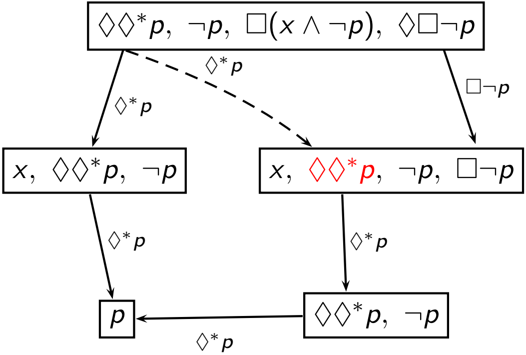
Clausal Tree Tableaux for H*, Example



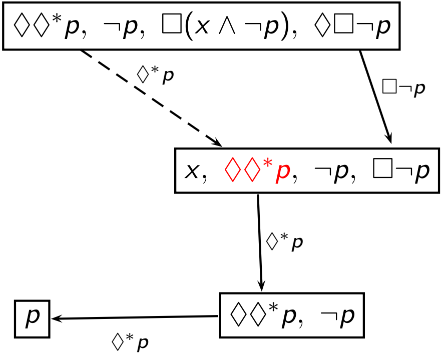
Clausal Tree Tableaux for H*, Example



Clausal Tree Tableaux for H*, Example



Clausal Tree Tableaux for H*, Example



Demo consists of nominally maximal clauses

Clausal Tree Tableaux for H*

- Nominal completion

$$C^{\Gamma} := C \cup \{\, s \mid \exists x \in C \; \exists D \in \Gamma.\; x \in D \wedge s \in D \,\}$$

- Require branches to be nominally coherent

$$\frac{C}{\overline{C^{\Gamma}}}$$

- Ignore clauses that aren't nominally maximal (i.e, $C = C^{\Gamma}$)
- See link CsD as link CsD^{Γ} (link lifting)

$$C \xrightarrow{s} D \;:\Leftrightarrow\; \exists E.\; CsE \in \Gamma \wedge E^{\Gamma} = D$$

Tableau Rules for H*

$$\frac{\Diamond s \in C}{CsD^{\Gamma}, D^{\Gamma} \;\mid\; \cdots} \quad C = C^{\Gamma},\ D \in \mathcal{D}(\mathcal{R}C; s),\ D^{\Gamma} \text{ clause}$$

$$\frac{\Diamond s \in C}{\text{closed}} \quad \forall D \in \mathcal{D}(\mathcal{R}C; s).\ D^{\Gamma} \text{ not a clause}$$

$$\frac{C_1 \overset{\Diamond^* s}{\rightarrow} \cdots \overset{\Diamond^* s}{\rightarrow} C_n \overset{\Diamond^* s}{\rightarrow} C_1}{\text{closed}} \quad \forall i \in [1, n].\ C_i \not\triangleright s$$

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^{2^n}	:	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*