A comparison of automata and tableau methods for modal satisfiability*

Franz Baader
Theoretical Computer Science
RWTH Aachen

- Advantages and disadvantages of both methods
- Intuitive comparison for K with global axioms
- Formal connection between the automata approach and the inverse tableau method for K

* Joint work with Stephan Tobies (supported by DFG under grant Gr 1324/3-2)

Methods for modal satisfiability

- Semantic tableaux and related methods.
- Translation into classical first-order logic and application of general theorem provers.
- Translation into quantified Boolean formulae and application of QBF solvers.
- Reduction to the emptiness problem for certain tree automata.
- Others: show finite model property; mosaic method; ...

Decidability/complexity results vs. implementable/practical algorithms

Advantages and disadvantages

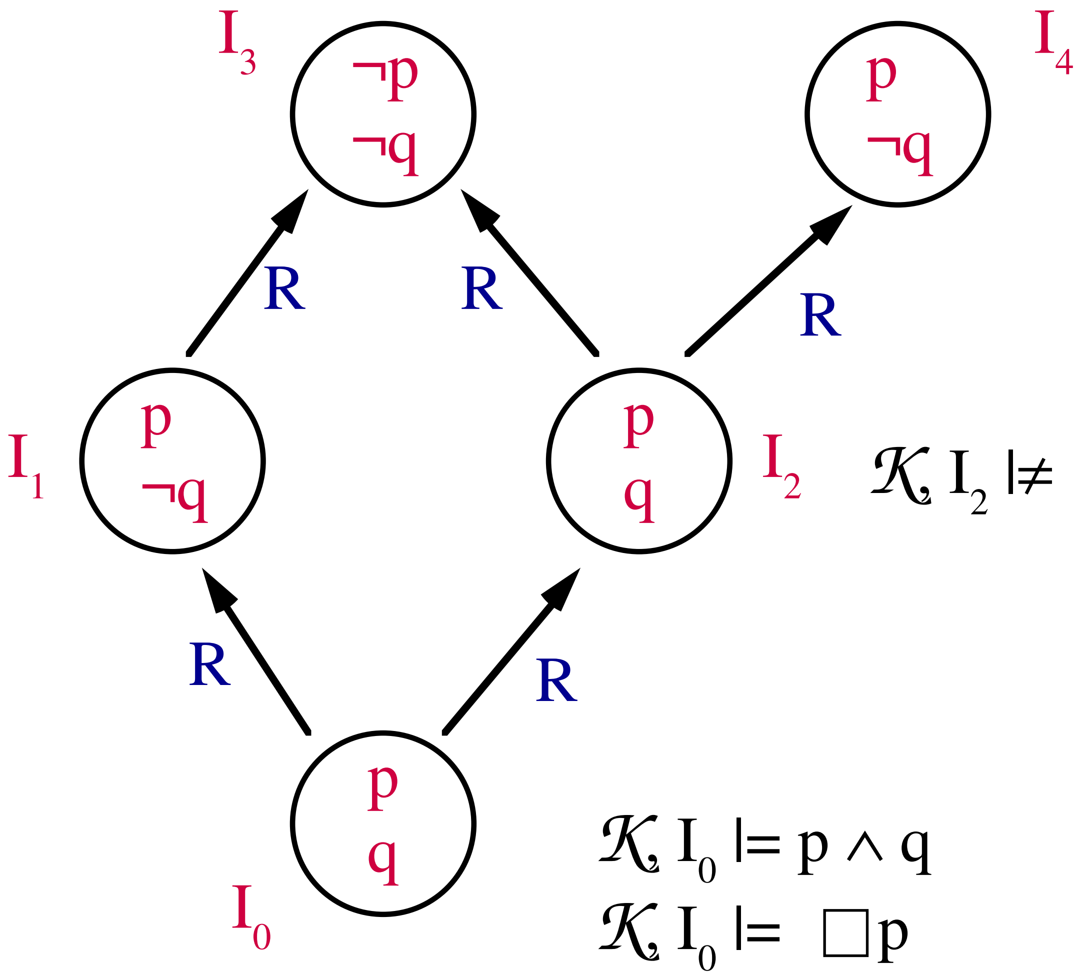
for the case of
K with global axioms

tableau-based	tree automata-based
+ optimized implementations	– implementations?
+ behave well in practice	– best-case exponential
– hard to obtain exact worst-case complexity upper-bound: - "natural" tableau algorithm is NExpTime - problem is ExpTime-complete	+ easy to obtain exact worst-case complexity upper-bound: + "natural" approach yields ExpTime algorithm

Approach that combines the advantages of both?

The basic modal logic K

- Extends propositional logic by a pair of unary modal operators box $\Box$ and diamond $\Diamond$.
- Semantics is defined via Kripke structures, i.e., sets of propositional interpretations linked by an accessibility relation.
  - box: $\Box G$ means that G holds in all accessible worlds.
  - diamond: $\Diamond G$ means that G holds in some accessible worlds.



$\mathcal{K}, I_2 \not\models \Diamond q$

$\mathcal{K}, I_0 \models p \wedge q$

$\mathcal{K}, I_0 \models \Box p$

$\mathcal{K}, I_0 \models \Diamond q$ and $\mathcal{K}, I_0 \models \Diamond \neg q$

$\mathcal{K}, I_0 \not\models \Box q$

$\mathcal{K}, I_0 \models \Diamond \Box (\neg p \wedge \neg q)$

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

The basic modal logic K

- Extends propositional logic by a pair of unary modal operators box $\Box$ and diamond $\Diamond$.
- Semantics is defined via Kripke structures, i.e., sets of propositional interpretations linked by an accessibility relation.
  - box: $\Box G$ means that G holds in all accessible worlds.
  - diamond: $\Diamond G$ means that G holds in some accessible worlds.

The modal formula G is satisfiable	iff	G is satisfiable in some world of some Kripke structure

finite tree structures suffice

- Satisfiability in K is PSpace-complete.

Tableau approach

for K without global axioms

- Tries to generate a finite tree structure satisfying G (where G is without loss of generality in NNF).
- Generates an initial world labeled with G, and
- then applies tableau rules:
  - propositional rules expand the label of the given world; rule for disjunction is nondeterministic.
  - diamond rule generates new accessible worlds
  - box rule extends the label of accessible worlds
  - clash rule detects obvious contradictions (p and ¬p for propositional variable p)

Tableau approach

Example:

Satisfiability of $\Diamond p \wedge \Box(\neg p \vee q)$

$\Diamond p \wedge \Box(\neg p \vee q)$

$\Diamond p, \quad \Box(\neg p \vee q)$

p

$\neg p \vee q$

$\cancel{\neg p} \quad q$

Satisfiability in K w.r.t. global axioms

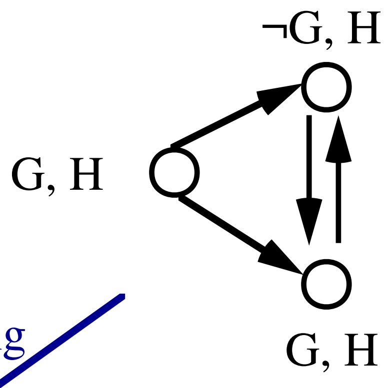
ExpTime-complete

[Spaan 93]

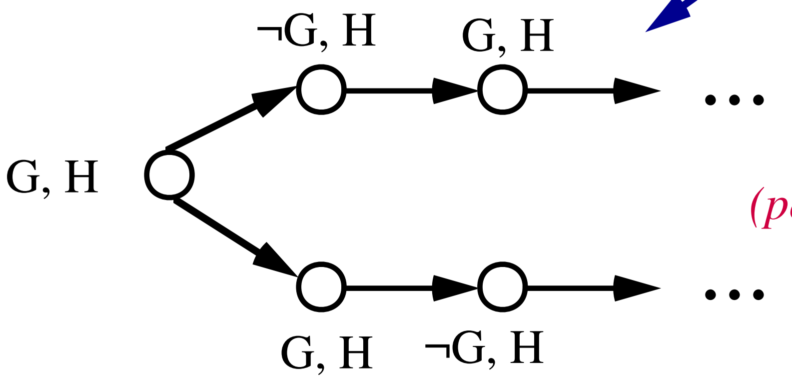
G is satisfiable w.r.t. the global axiom H | iff | G is satisfiable in a Kripke structure in which all worlds satisfy H

finite structures suffice

tableau approach tries to generate one



unraveling



(possibly infinite) tree structures suffice

automata approach test its existence

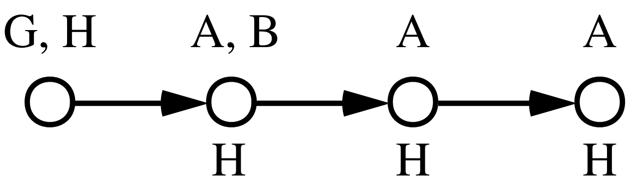
Tableau-approach

for K with global axioms

- Tries to generate a finite structure satisfying G w.r.t. H.
- The axiom H is added to every world generated by the algorithm.
- Blocking required to ensure termination (cyclic structures).

$G = A \wedge \Box B$

$H = \Diamond A$



- Length of paths: may become exponential before blocking occurs.
- Nondeterminism: treatment of disjunction.

NExpTime complexity

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

Automata approach

reduction to the emptiness problem for automata on infinite trees

- Tests for the existence of an (infinite) tree structure satisfying G w.r.t. H.
- States of the automaton: propositionally expanded sets of subformulae of G and H that contain H.
- Initial state: contains G.
- Transitions: look for the existence of appropriate sons if they are required by diamond formulae (otherwise: "dummy" sons).
  No transition from states containing a clash.
- Looping tree automaton: accepts if there is an infinite run.
- Run looks like an infinite tableau (no blocking required).
- Automaton is nondeterministic due to presence of disjunction.

Automata approach

Example:

$H = \Diamond p$ and $G = \Box(\neg p \lor q)$

successful run on

$\bigcirc$ $\Diamond p, \;\; \Box(\neg p \lor q), \;\; \neg p \lor q, \;\; \neg p$

$\downarrow$

$\bigcirc$ $\Diamond p, \;\; p, \;\; \neg p \lor q, \;\; q$

$\downarrow$

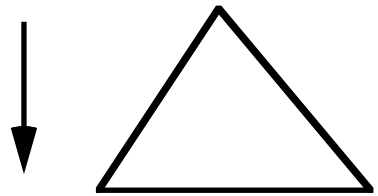
$\bigcirc$ $\Diamond p, \;\; p, \;\; q$

$\downarrow$

$\vdots$

Emptiness test

naive top-down approach



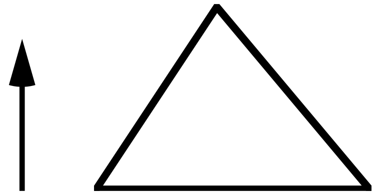
- Tries to construct an infinite tree and an infinite run on this tree.
- Starts with an initial state at the root, and then generates son nodes labeled according to the transition function.
- Looks for state repetition on paths to ensure termination.
- Very similar to tableau-approach with blocking.
- Complexity: NP in size of automaton if the automaton is nondeterministic.

Since the constructed automaton is exponential in the size of the formula, this still leaves us with a NExpTime procedure.

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

Emptiness test

improved bottom-up approach
(dynamic programming)



- Computes inactive states, i.e., states that cannot occur on an infinite run of the automaton:
  - Starts with obviously inactive states, i.e., states that do not have successors states w.r.t. the transition function.
  - Propagates inactiveness along the transition function.
- Naive implementation already polynomial.
- Using appropriate data structures, the set of inactive states can be computed in linear time.

Since the constructed automaton is exponential in the size of the formula, this provides us with an ExpTime procedure.

Emptiness test

bottom-up approach in more detail
(naive implementation)

states containing p, ¬p
in constructed automaton

Obviously inactive states:

Q_0 := {q | q is state s.t. there is not transition (q,.) –> (...)}

Propagation of inactiveness:

$Q \succ Q \cup \{q\}$ iff all transitions (q,.) –> $(q_1,\ldots,q_n)$ are such that some $q_i \in Q$

Set of inactive states:

$Q_0^{\succ} := \bigcup\{Q \mid Q_0 \succ^* Q\}$ (propagation closure)

states containing the
the formula G in
constructed automaton

Emptiness test:

$L(\mathcal{A}) = \emptyset$ iff all initial states belong to $Q_0^{\succ}$

The inverse calculus

for modal K [Voronkov 01]
generates unsatisfiable formulae

Rules

$$\frac{\Gamma, A}{\Gamma, A \wedge B} \qquad \frac{\Gamma, B}{\Gamma, A \wedge B} \qquad \frac{\Gamma, A \mid \Delta, B}{\Gamma, \Delta, A \vee B}$$

$$\frac{\Gamma, A}{\Box\Gamma, \Diamond A} \qquad \frac{\Gamma}{\Box\Gamma, \Diamond A}$$

Axioms

$\{p, \neg p\}$

for propositional variables p

To test for satisfiability of the formula G,
restrict rules and axioms to subformulae of G.

Satisfiability test:

$\mathcal{S}_0 := \{\Gamma \mid \Gamma \text{ is axiom}\}$ and $\mathcal{S}_0^{\vdash} := \bigcup\{\mathcal{S} \mid \mathcal{S}_0 \vdash^* \mathcal{S}\}$ (inference closure)

G unsatisfiable iff $\{G\} \in \mathcal{S}_0^{\vdash}$

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

The inverse calculus

for modal K [Voronkov 01] with global axioms

Rules			Axioms
$\dfrac{\Gamma, A}{\Gamma, A \wedge B}$	$\dfrac{\Gamma, B}{\Gamma, A \wedge B}$	$\dfrac{\Gamma, A \mid \Delta, B}{\Gamma, \Delta, A \vee B}$	$\{p, \neg p\}$
$\dfrac{\Gamma, A}{\Box\Gamma, \Diamond A}$	$\dfrac{\Gamma}{\Box\Gamma, \Diamond A}$	$\dfrac{\Gamma, H}{\Gamma}$	for propositional variables p

To test for satisfiability of the formula G w.r.t. H,
restrict rules and axioms to subformulae of G and H.

Satisfiability test:

$\mathcal{S}_0 := \{\Gamma \mid \Gamma \text{ is axiom}\}$ and $\mathcal{S}_0^{\vdash} := \bigcup\{\mathcal{S} \mid \mathcal{S}_0 \vdash^* \mathcal{S}\}$ (inference closure)

G unsatisfiable w.r.t. H iff $\{G\} \in \mathcal{S}_0^{\vdash}$ or $\emptyset \in \mathcal{S}_0^{\vdash}$

Connecting the two approaches

main technical result

$|[\Gamma]| := \{q \mid q$ is state containing $\Gamma\}$ and $|[\mathcal{S}]| := \bigcup\{|[\Gamma]| \mid \Gamma$ in $\mathcal{S}\}$

Theorem The automata approach and the inverse method can simulate each other:

- If $Q_0 \succ^* Q$, then there exists a set of sequents $\mathcal{S}$ such that $\mathcal{S}_0 \vdash^* \mathcal{S}$ and $Q \subseteq |[\mathcal{S}]|$.
- If $\mathcal{S}_0 \vdash^* \mathcal{S}$, then there exists a set of states Q such that $Q_0 \succ^* Q$ and $|[\mathcal{S}]| \subseteq Q$.

Consequences

of the theorem

- The propagation closure and the inference closure "agree":

  $Q_0^{\succ} = |[\mathcal{S}_0^{\vdash}]|$

  - ➽ The inverse calculus yields an "on-the-fly" implementation of the emptiness test for the constructed automaton.
  - ➽ One sequent represents several states (states containing this sequent).

- The inverse calculus yields an ExpTime decision procedure for satisfiability w.r.t. global axioms in K.

  - ➽ This "on-the-fly" implementation of the emptiness test yields an a procedure that is optimal w.r.t. worst-case complexity.

Further results

concerning optimizations of the procedure

Voronkov introduces optimizations of the inverse calculus for K without global axioms:

- redundant sequents (corresponding to unreachable states)
- redundant inferences avoided by imposing an ordering-restriction on the application of the diamond inference rules.

Completeness of both optimizations can be shown within the automata framework (which yields simpler proofs).

Future work

- Can the inverse method be used to obtain a PSPACE-algorithm for satisfiability in K w/o global axioms?
- Can Voronkov's optimizations be adapted to the calculus dealing with global axioms?
- Can additional optimizations considered by Voronkov (eg. prefix subsumption) also be justified using the automata approach?
- Can the results be transferred to other modal/description logics?
- What can we say about alternating automata?
  How can we get a "direct" algorithm that tests emptiness for alternating (looping) tree automata?

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH