

Complexity of decision problems for mixed and modal specifications

Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman,
and Andrzej Wąsowski.

April 2, 2008



Outline

Background

Contributions of paper

Conclusions

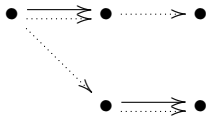
Future work



Background



- ▶ Modal transition systems generalize labeled transition systems:



- ▶ Either “may” (dashed) or “must” transitions (solid lines)
- ▶ Can model allowed (“may”) and required (“must”) behavior
- ▶ But anything that is required is also allowed: “must \subseteq may”



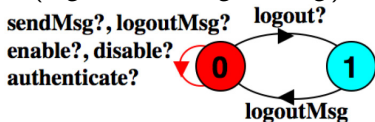
Refinement

- ▶ Refinement gives an information ordering upon states.
- ▶ \preceq is a refinement relation if $a \preceq b$ implies
 - ▶ For every must transition from a to a' , there is a matching must transition from b to a b' such that $a' \preceq b'$
 - ▶ For every may transition from b to b' , there is a matching may transition from a to a a' such that $a' \preceq b'$
- ▶ Thus transitions that must happen still must happen in refinements, and transitions that may happen in refinements must have been possible to happen before.

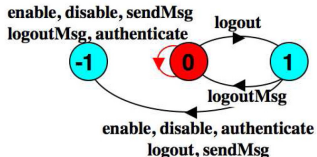


Webmail example [Uchitel et al., ICSE 2007]

- ▶ Modal transition system synthesis of $G(\text{logout} \rightarrow X \text{logoutMsg})$, may-transitions have a “?”:

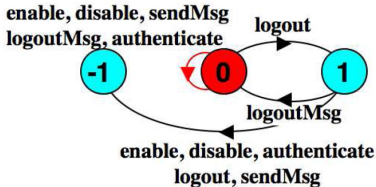


- ▶ Labeled transition system synthesis of same LTL formula, refines that modal transition system:

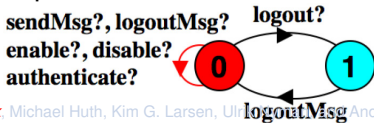


Implementations

- ▶ Implementation = modal transition system for which “must” equals “may”, correspond to labeled transition systems
- ▶ Every modal transition system has an implementation, e.g.



implements



Mixed transition systems

- ▶ Mixed transition systems = modal transition systems without consistency condition that must \subseteq may, e.g.:



- ▶ Not all mixed transition systems have an implementation.
- ▶ Those that do are called consistent, e.g.:



Contributions of paper

Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wąsowski.
Complexity of decision problems for mixed and modal specifications



Common implementation (**CI**) & Consistency (**C**)

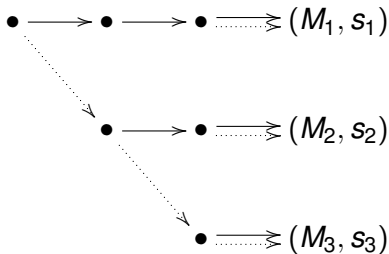
CI: Given a set of modal or mixed transition systems, is there an implementation that refines all systems of that set?

- ▶ Can a set of differing specifications be reconciled?
 - ▶ E.g. systems may specify scenarios, features or faults.
 - ▶ E.g. systems may specify hard requirements.
- ▶ For modal transitions systems, **CI** is PTIME-complete [Huth & Hussain 2005] if the cardinality of the set is fixed.
- ▶ **C**: Does a mixed transition system have an implementation?
- ▶ **C** is **CI** for cardinality 1 and mixed systems.



CI reduces to C for mixed systems

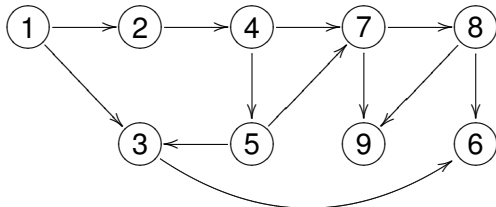
- ▶ We can reduce the question CI for a set $\{(M_i, s_i)\}$ of cardinality n to a question C of a set of cardinality 1 of one **mixed** transition system:



- ▶ Thus the important question is that of CI for $n > 1$ modal transition systems



Generalized geography (**GG**)



- ▶ Plays start at given node, two players move in strict alternation.
- ▶ Players choose a not-yet-visited successor state.
- ▶ If a player has no valid move, she loses.
- ▶ Determining if player has winning strategy is **PSPACE-complete**.



Reducing GG to CI

For any instance of GG , construct set of modal systems that has common implementation iff player 0 has a winning strategy for that instance of GG :

- ▶ Winning strategy has to work no matter what Player 1 plays.
- ▶ Encode Player 1 choices as must-transitions, forcing an implementation to consider every choice of Player 1.
- ▶ Use may-transitions for Player 0, allowing an implementation to choose the move of Player 0.
- ▶ Add further models to ensure that at least one may-transition is used.



Upper bound for CI

Given set of models $S = \{(M_i, s_i)\}$

- ▶ There is alternating tree automata $A_{(M_i, s_i)}$ accepting exactly the implementations of (M_i, s_i) [Bruns & Godefroid 2000]
- ▶ We can check non-emptiness of intersection of these automata in EXPTIME in sum of sizes of the M_i .



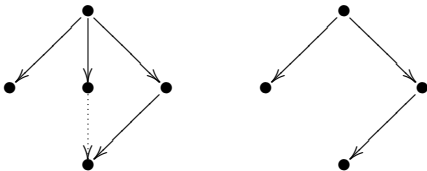
Results for common implementation

	Modal TS	Mixed TS
Consistency	Trivial	PSPACE-hard in EXPTIME
Fixed card	PTIME-complete	PSPACE-hard in EXPTIME
Card n	PSPACE-hard, in EXPTIME	PSPACE-hard in EXPTIME



Thorough refinement (TR)

Modal refinement is “incomplete”: all implementations of (M, s) may also be implementations of (N, j) , although $(N, j) \not\leq (M, s)$:

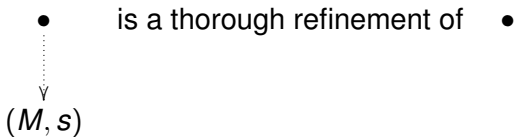


- ▶ We define thorough refinement (TR) to be this relation of inclusion of implementations.
- ▶ TR cannot be easily reduced to CI , as there is no way to “complement” a mixed or modal transition system.



Lower bounds for TR

- ▶ For mixed transition systems, we can reduce C to TR : A model (M, s) is **not** consistent iff



- ▶ For modal transition systems, we need a different approach.



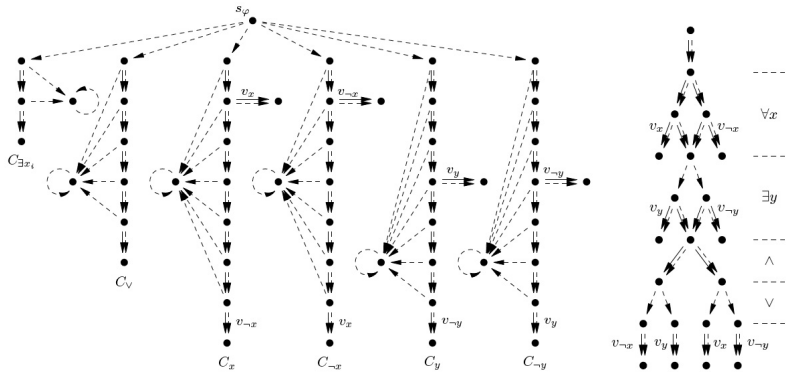
Lower bound for “modal” TR

We reduce QCNF, a variant of Quantified Boolean Formulae, to TR for modal transition systems:

- ▶ For sentence ϕ , we create two models
 - ▶ M_ϕ : “contains” all attempted proofs of the truth of ϕ .
 - ▶ N_ϕ : “contains” all wrong proofs of the truth of ϕ .
- ▶ Then ϕ is false iff all implementations of M_ϕ are implementations of N_ϕ , i.e., every attempted proof is wrong.



Illustration of (M_ϕ, s_ϕ) and (N_ϕ, t_ϕ) for
 $\phi = \forall x \exists y (\neg x \vee y) \wedge (x \vee \neg y)$



Upper bounds for TR

- ▶ We construct alternating tree automata $A_{(M,s)}$ and $\bar{A}_{(N,t)}$, the complement of $A_{(N,t)}$.
- ▶ Exploits that alternating tree automata are more expressive than mixed transition systems: there is (in general) no mixed TS (\bar{N}, \bar{t}) having exactly those implementations accepted by $\bar{A}_{(N,j)}$.
- ▶ We perform a non-emptiness intersection test on $A_{(M,s)}$ and $\bar{A}_{(N,j)}$, doable in EXPTIME.



Conclusions



	Modal TS	Mixed TS
Fixed card CI	PSPACE-complete	PSPACE-hard in EXPTIME
Card n CI	PSPACE-hard in EXPTIME	PSPACE-hard in EXPTIME
TR	PSPACE-hard in EXPTIME	PSPACE-hard in EXPTIME



Future work



Reduce gap between upper and lower bounds

We conjecture:

- ▶ Common implementation (**CI** & **C**): EXPTIME-complete
- ▶ Thorough refinement (**TR**): PSPACE-complete



Acknowledgments

- ▶ **Harald Fecher** made us aware of the counterexample for incompleteness of refinement used in this paper. This then led to the rediscovery of a history of such counterexamples.
- ▶ **Nir Piterman** helped in improving the presentation of the proof for Theorem 8.
- ▶ We thank **Igor Walukiewicz**, **Wolfgang Thomas** and **Dietmar Berwanger** for independently confirming that validity of vectorized calculus formulae is in EXPTIME.

