MoTraS - A Tool for Modal Transition Systems

Martin Stoll

Informatik VII
Chair for Foundations of Software Reliability and Theoretical Computer Science

Technische Universität München

27.09.2011
Outline

1 Theory

2 MoTraS

3 Implementation and Conclusion

4 Future Work
Outline

1. Theory
2. MoTraS
3. Implementation and Conclusion
4. Future Work
Modal Transition Systems (MTSs)

- Extension of Labeled Transition Systems (LTSs)
- A set of processes \( P \)
- The action alphabet \( \Sigma \)
- Two types of transitions:
  - May-transitions \( \xrightarrow{\text{may}} \) are allowed to be in every refinement of the given system
  - Must-transitions \( \xrightarrow{\text{must}} \) are obliged to be in every refinement of the given system
- SPECIAL CASE: Deterministic Modal Transition Systems (dMTSs)
Example
Disjunctive Modal Transition Systems (DMTSSs)

- Extension of MTSs
- **Disjunctive transitions**: A Set of must-transitions from which only one has to be realized in every refinement of the given system.
Implementations

Each may-transition of the given system is either omitted or turned into a must-transition.
Refinements

The refinement must contain every must-transition of the given system

AND

Every may-transition of the refinement system must be defined in the given (refined) system
Refinements II
Deterministic Hull

smallest (w.r.t. refinement) deterministic system refined by the original system
Conjunction

- greatest lower bound for a given set of processes w.r.t. the modal refinement
- not always expressible as MTS, but as DMTS
Composition

- synchronizing alphabet $\Gamma \subseteq \Sigma$
- sequential and synchronous scheduling of transitions
- Example with $\Gamma = \{a\}$:
Outline

1. Theory
2. MoTraS
3. Implementation and Conclusion
4. Future Work
Features

- single or double GUI
- drawing of MTSs and DMTSs
- import and export (various formats) of DMTSs
- random (d/D)MTS generator and MTS refinement generator
- runtime test for algorithms
- various sample MTSs
Algorithms

- Modal Refinement (two different algorithms)
- Smallest Common Implementation for dMTSs
- Greatest Common Implementation for DMTSs
- Deterministic Hull
- Composition
- Conjunction for DMTSs
- LTL Model Checking for DMTSs
MoTraS - a short demonstration ...
Outline

1. Theory
2. MoTraS
3. Implementation and Conclusion
4. Future Work
Two ways to check MR

### MRNaive

Fixed point computation over the whole Cartesian Product of the sets of processes

### MRImproved

BFS to find all relevant pairs of processes and fixed point computation on the resulting set
Implementation and Conclusion

Runtime evaluation

(a) transitions-processes-ratio (TPR) $1.5n$

(b) TPR $2n$

(c) TPR $0.25n^2$
Outline

1 Theory

2 MoTraS

3 Implementation and Conclusion

4 Future Work
Future Work

- Extension of algorithms for DMTSs
- Algorithms for checking Thorough Refinement
- Further optimizations of the algorithms
Thank you for your attention!