

Algorithms for Model Checking (2IMF35)

Lecture 12 Retrospect + Outlook

Tim Willemse

(timw@win.tue.nl)

<http://www.win.tue.nl/~timw>

MF 6.073

Part I: basics

- ▶ Lecture 1: Syntax + Semantics of CTL* →PVR
- ▶ Lecture 2: Symbolic algorithm for CTL and fair CTL
- ▶ Lecture 3: Counterexamples and witnesses for fragments of CTL
- ▶ Lecture 4: Emerson-Lei algorithm improving over the naive algorithm →AR

part II: complexity

- ▶ Lecture 5: Boolean equation systems
- ▶ Lecture 6: Parity Games
- ▶ Lecture 7: Recursive algorithm
- ▶ Lecture 8: Small Progress Measures

part III: data

- ▶ Lecture 9-11: Parameterised Boolean equation systems →SV
 - Symbolic encoding the model checking problem as a PBES
 - Redundant parameter detection and elimination
 - Instantiating to a BES and solving the BES
 - Symbolic approximation + Gauß Elimination

Model Checking:

- ▶ Problem is in $NP \cap co-NP$; what is its true complexity?
- ▶ Bigstep algorithm (=Recursive+SPM) for Parity Games has best worst-case performance when $d \ll n$, viz. roughly $\mathcal{O}(n^{d/3})$, and the Subexponential algorithm has asymptotically best worst-case complexity, viz. roughly $\mathcal{O}(n^{\sqrt{n}})$.

Model Checking:

- ▶ Problem is in $NP \cap co-NP$; what is its true complexity?
- ▶ Bigstep algorithm (=Recursive+SPM) for Parity Games has best worst-case performance when $d \ll n$, viz. roughly $\mathcal{O}(n^{d/3})$, and the Subexponential algorithm has asymptotically best worst-case complexity, viz. roughly $\mathcal{O}(n^{\sqrt{n}})$.

Parity Games:

- ▶ Solve Parity Games/BESs using *Integer Equation Systems*;
- ▶ Investigate runtime complexity of parity game algorithms on special games;
- ▶ Lower bounds for algorithms?

Model Checking:

- ▶ Problem is in $NP \cap co-NP$; what is its true complexity?
- ▶ Bigstep algorithm (=Recursive+SPM) for Parity Games has best worst-case performance when $d \ll n$, viz. roughly $\mathcal{O}(n^{d/3})$, and the Subexponential algorithm has asymptotically best worst-case complexity, viz. roughly $\mathcal{O}(n^{\sqrt{n}})$.

Parity Games:

- ▶ Solve Parity Games/BESs using *Integer Equation Systems*;
- ▶ Investigate runtime complexity of parity game algorithms on special games;
- ▶ Lower bounds for algorithms?

PBES technology:

- ▶ Symmetry and confluence detection and reduction for reducing the complexity
- ▶ Abstract Interpretation and predicate abstraction

Model Checking:

- ▶ Problem is in $NP \cap co-NP$; what is its true complexity?
- ▶ Bigstep algorithm (=Recursive+SPM) for Parity Games has best worst-case performance when $d \ll n$, viz. roughly $\mathcal{O}(n^{d/3})$, and the Subexponential algorithm has asymptotically best worst-case complexity, viz. roughly $\mathcal{O}(n^{\sqrt{n}})$.

Parity Games:

- ▶ Solve Parity Games/BESs using *Integer Equation Systems*;
- ▶ Investigate runtime complexity of parity game algorithms on special games;
- ▶ Lower bounds for algorithms?

PBES technology:

- ▶ Symmetry and confluence detection and reduction for reducing the complexity
- ▶ Abstract Interpretation and predicate abstraction

Verification:

- ▶ Analyse DSL programs using dedicated transformations to parity games/PBES;
- ▶ Probabilistic and quantitative verification using *Rational Equation Systems*

Control software for the Large Hadron Collider

- ▶ Hierarchical system of $>25\,000$ communicating FSMs
- ▶ Nearly fully semi-formally described
- ▶ BDD-based analysis of a subtree consisting of:
 - 7 FSMs: $5 \cdot 10^6$ states, $24 \cdot 10^6$ transitions; ± 1 minute
 - 9 FSMs: $800 \cdot 10^6$ states; ± 10 minutes
 - 11 FSMs: $120 \cdot 10^9$ states; \pm half a day
- ▶ Dedicated verification: SAT solving techniques
- ▶ Results:
 - Approx. 5% of all FSMs suffer from livelocks (20% of the FSMs that *can* be affected)
 - Approx. 4% of all FSMs suffer from reachability issues

Verification for Dezyne

- ▶ Dezyne: DSL and toolsuite for modelling and generating software
- ▶ Verification of Dezyne models using FDR and, more recently, mCRL2
- ▶ Dezyne used at:
 - ASML
 - Fei
 - NSpyre
 - ...

The DIRAC grid solution used at the Large Hadron Collider beauty experiment

- ▶ cooperating distributed services
- ▶ light-weight agents delivering the workload to the Grid resources
- ▶ agents run concurrently
- ▶ State spaces of $160 \cdot 10^6$ states are no exception
- ▶ Results:
 - Livelocks
 - Race conditions
 - Dead jobs reviving (zombies)
 - ...

Internship/final projects (possibly at CERN?)/research for fun? Contact me!