# Algorithms for Model Checking (2IW55)

## Lecture 15 Retrospect + Outlook

Tim Willemse
(timw@win.tue.nl)
http://www.win.tue.nl/∼timw
HG 6.76

Department of Mathematics and Computer Science

TU/e Technische Universiteit Eindhoven University of Technology

# Retrospect



CTL$^*$ and fair CTL$^*$:

- Lecture 1: Syntax + Semantics of CTL$^*$
- Lecture 2: Labelling algorithm for CTL and fair CTL
- Lecture 3: Symbolic algorithm for CTL and fair CTL
- Lecture 4: Counterexamples and witnesses for fragments of CTL
- Lecture 5: Equivalence relations and preorders and their relation with CTL$^*$

# Retrospect



$\mu$-calculus and friends:

- Lecture 6: Emerson-Lei algorithm improving over the naive algorithm
- Lecture 8: Boolean equation systems
  - Encoding the model checking problem as a BES
  - and Gauß Elimination for solving these
- Lecture 9-10: Parameterised Boolean equation systems
  - 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
- Lecture 11-13: Parity Games
  - Transforming Parity Games to BESs and vice versa;
  - Algorithms for computing the winners in the game;

# Outlook: Selection of Research and Open Ends



- CTL$^*$/LTL:
  - Directly encode CTL$^*$ formulae as PBESs
    - A linear encoding to the first-order modal $\mu$-calculus was recently shown correct
- PBES technology:
  - Symbolic algorithms for reducing the complexity:
    - Symmetry detection and reduction
    - Confluence reduction
  - On-the-fly BES generation, minimising and solving
  - Counterexample generation and visualisation
  - Verification of real-time systems
- Parity Games:
  - Tighten bounds on the lower and worst-case complexity of Parity Game solving algorithms;
  - Heuristics for solving Parity Games (e.g., inspiration from Gauß Elimination);
- Model Checking:
  - Problem is in NP∩co-NP; what is its true complexity?
  - Bigstep algorithm (=Recursive+SPM) for Parity Games has best worst-case performance

# Outlook: Applications



CERN case study: Control software for the Large Hadron Collider

- Hierarchical system of >30 000 communicating FSMs
- Nearly fully semi-formally described
- Analysis of a subtree consisting of +/-6 FSMs:
  - 7 FSMs: $510^6$ states, $2410^6$ transitions; +/- 1 minute
  - 9 FSMs: $80010^6$ states; +/- 10 minutes
  - 11 FSMs: $12010^9$ states; +/- half a day
- Dedicated verification: SAT solving techniques