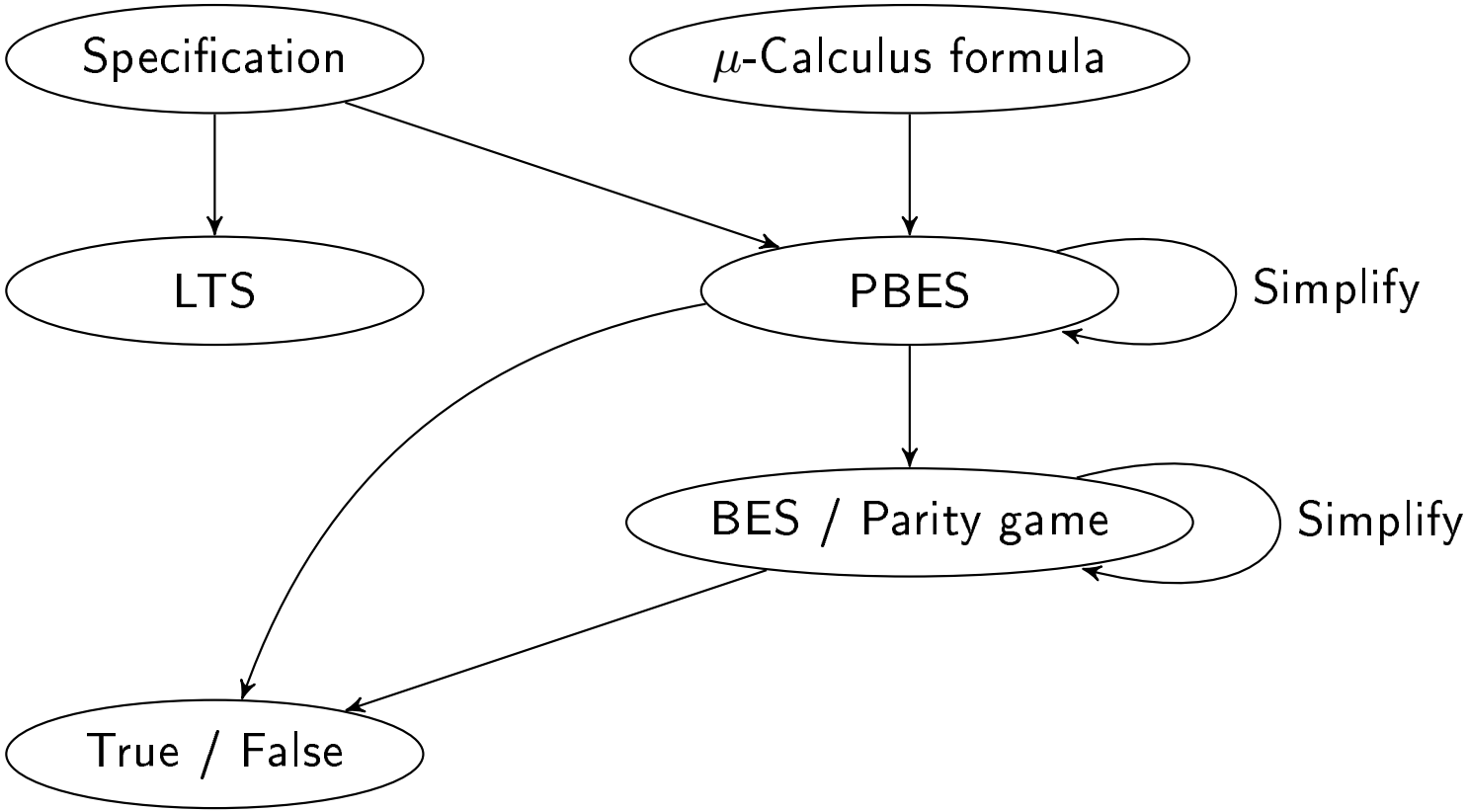
Model checking using Parameterised Boolean Equation Systems

Jeroen Keiren
(j.j.a.keiren@tue.nl)
http://www.win.tue.nl/∼jkeiren

Department of Mathematics and Computer Science

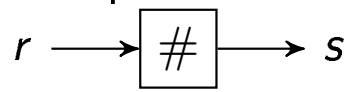
TU/e Technische Universiteit Eindhoven University of Technology

Methodology



Example: Specification

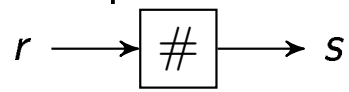
One place buffer



sort $Message$;
map $m_0 : Message$;
act $r, s : Message$;
proc $P(read : Bool, m : Message) =$
$\Sigma m' : Message.read \rightarrow r(m').P(\neg read, m')$
$\neg read \rightarrow s(m).P(\neg read, m_0)$;
init $P(true, m_0)$;

Example: Specification

One place buffer



sort $Message$;
map $m_0 : Message$;
act $r, s : Message$;
proc $P(read : Bool, m : Message) =$
$\Sigma m' : Message.read \rightarrow r(m').P(\neg read, m')$
$\neg read \rightarrow s(m).P(\neg read, m_0)$;
init $P(true, m_0)$;

"The buffer does not contain any deadlocks"

$$\nu X.[true]X \wedge \langle true \rangle true$$

Example: PBES

"The buffer does not contain any deadlocks"

$$\nu X.[true]X \wedge \langle true \rangle true$$

Example: PBES

"The buffer does not contain any deadlocks"

$$\nu X.[true]X \wedge \langle true \rangle true$$

sort $Message$;
map $m_0 : Message$;
pbes $\nu X(read : Bool, m : Message) =$
$(\forall m' : Message.read \implies X(\neg read, m'))$
$\wedge (\neg read \implies X(\neg read, m_0))$
init $X(true, m_0)$

How to continue?

- Instantiate all data to obtain BES/parity game
  - Reduce modulo equivalences
  - Solve BES/parity game ($NP \cap co - NP$)
- Simplify PBES on a symbolic level

How to continue?

- Instantiate all data to obtain BES/parity game
  - Reduce modulo equivalences
  - Solve BES/parity game ($NP \cap co - NP$)
- Simplify PBES on a symbolic level

Current PBES related research:

- Allow equivalence reductions on PBES level
- Improve static simplification techniques
- Simplify correctness proofs of simplification techniques

Thank you!

Want to know more? Come talk to me!