Problem

- Decide satisfiability of a given DQBF
- DQBF = dependency quantified Boolean formula
  - Propositional logic formula extended with quantifiers with explicit dependencies between them
- NEXPTIME-complete problem
- Example:
  ∃x₁ ∀x₂ ∃y₁ (x₁) ∃y₂ (x₂). (x₁ ∧ x₂) ⇔ (y₁ ⇔ y₂)
  - y₁ depends only on x₁ (and y₂ on x₂), meaning that the value of y₁ cannot change based on the value of x₂
  - Formula is unsatisfiable as y₁ and y₂ cannot coordinate
  - Can be used for solving
  - controller synthesis problem (CSP)
  - partial equivalence checking (PEC) - Can a combinational circuit with black boxes (BB) be equivalent to a given specification?

Method

- Quantifier elimination is used as the basic solving technique
  - Quantifiers are iteratively eliminated until we end up with True or False
  - Algorithm improved by quantifier localisation
  - Quantifiers are pushed inside the formula resulting in a faster elimination
- Binary decision diagrams (BDDs) are used to represent propositional subformulas in DQBF
  - The BDD on the right represents
    (¬x₁ ∧ x₂ ∧ x₃) ∨ (x₁ ∧ ¬(x₂ ⇔ x₃))

Results

- Quantifier localisation improvements
  - Correction of existing results
  - Proved that it can be used in subformulas
  - Proved that universal quantifier elimination can be done locally
- Solver DQBDD
  - New algorithm solving DQBF satisfiability
  - Implemented in C++ using BDDs
  - Winner of the DQBF track of QBFEval'20 competition [1]
- Publications under preparation
  - Joint journal paper with the HQS team (University of Freiburg)
  - Publication about the new algorithm of DQBDD

Experiments

- Comparison of possible quantifier localisation and elimination strategies
- Comparison of DQBF solvers using different benchmarks
  - Results:
    - DQBDD is far better than other solvers for PEC
    - Figure below shows a cactus plot comparing runtimes of DQBF solvers for PEC instances

QBFEval'20 Competition

- Comparison of DQBF solvers on selected benchmarks
- Results
  1. DQBDD - 257 solved in 5396 s
  2. HQS - 195 solved in 2662 s
  3. iProver - 170 solved in 17399 s

References