Tomáš Balyo³  Uwe Egly¹  Charles Jordan⁴  Lukasz Kaiser⁵
Florian Lonsing¹  Johannes Oetsch¹  Martina Seidl²

¹Knowledge-Based Systems Group, Vienna University of Technology, Austria
²Institute for Formal Models andVerification, JKU Linz, Austria
³Karlsruhe Institute of Technology (KIT) Karlsruhe, Germany
⁴Division of Computer Science, Hokkaido University, Japan
⁵LIAFA and CNRS, Université Paris Diderot, and Google Inc.

4th International Workshop on Quantified Boolean Formulas,
4 July 2016, Bordeaux, France (affiliated to SAT 2016)
**Variants of DepQBF (1/2)**

**QCDCL with Generalized QRES Axioms:** [LES16]

- Clause (cube) learning based on Q-resolution calculus (QRES).
- Traditional QCDCL: current assignment $A$ either falsifies a clause or satisfies all clauses of PCNF $\psi$.
- Learning: QRES guided by assignment $A$.
- Idea: incomplete satisfiability testing of $\psi[A]$ to learn stronger clauses.
- Implementation in DepQBF, submitted three variants to QBFEVAL CNF track (heuristics, amount of preprocessing, ...).

⇒ *talk on Tuesday, session 11:00-12:30.*
Variants of DepQBF (2/2)

**Incremental Solving Track:** [MMLB12, LE14, MMB15]

- Solve a sequence of PCNFs \( \langle \psi_1, \ldots, \psi_n \rangle \).
- PCNF \( \psi_i \) is syntactically related to \( \psi_{i+1} \).
- Reuse subset of clauses and cubes learned from \( \psi_i \) when solving \( \psi_{i+1} \).
- Submitted DepQBF 5.0 (latest public version).

**Certification Track:**

- Tool suite QBFCert: extracting Herbrand (Skolem) functions from clause (cube) resolution proofs [NPL+12].
- For SAT/UNSAT: DepQBF 5.0 (without dynamic QBCE [LBB+15]).
- For UNSAT only: DepQBF 5.0 with dynamic QBCE (redundant clauses ignored for proof generation).
Parallel Solving of Primal/Dual Encodings

Solver “pd-par-depqbf”:

- Idea: solve primal and dual encoding of non-CNF instance [VG13].
- Input: prenex non-CNF formula $\psi$.
- Encode $\psi$ as prenex CNF $\psi^+$ via Tseitin translation, apply Bloqger.
- Encode $\neg\psi$ as prenex CNF $\psi^-$ via Tseitin translation, apply Bloqger.
- Run two identical instances of DepQBF on $\psi^+$ and $\psi^-$ in parallel.
- No communication between solver instances.
- Simple shell script controls solver instances, returns appropriate exit code after termination.
Parallel QBF Solving Without Knowledge Sharing:

- MPI-based master-worker framework.
- Master splits search space into subproblems by assignments.
- Workers solve subproblems by solving input QBF under assumptions.
- Master combines results of subproblems.
- Workload balancing to avoid long idle times of workers.
- Integration of DepQBF in worker processes.
- Tool paper at SAT 2014 [JKLS14].
HordeQBF

Modular and Massively Parallel QBF Solving:

- Based on HordeSAT [BSS15].
- MPI-based parallel portfolio of arbitrary (Q)CDCL solvers.
- Parallel execution of identical (Q)CDCL solvers.
- Integration of DepQBF.
- No search-space partitioning.
- Solver instances are diversified by their parameters (heuristics, ...).
- Frequent clause/cube sharing.
- Tool paper at SAT 2016: promising experiments, up to 1024 cores.

⇒ talk by Tomáš Balyo on Friday, session 11:00–12:40.


Florian Lonsing and Uwe Egly.
Incremental QBF Solving.

Florian Lonsing, Uwe Egly, and Martina Seidl.
Q-Resolution with Generalized Axioms.

Christian Miller, Paolo Marin, and Bernd Becker.
Verification of partial designs using incremental QBF.
Paolo Marin, Christian Miller, Matthew D. T. Lewis, and Bernd Becker.
Verification of Partial Designs using Incremental QBF Solving.

Aina Niemetz, Mathias Preiner, Florian Lonsing, Martina Seidl, and Armin Biere.
Resolution-Based Certificate Extraction for QBF - (Tool Presentation).
Allen Van Gelder.
Primal and Dual Encoding from Applications into Quantified Boolean Formulas.