Failed Literal Detection for QBF

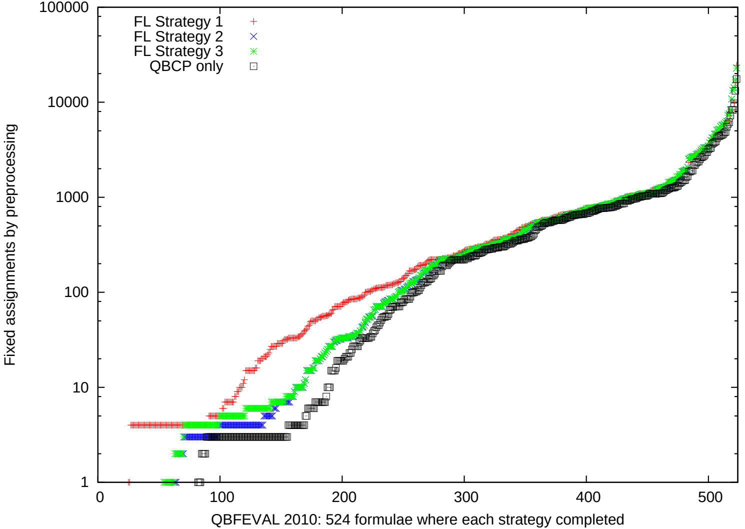
Florian Lonsing and Armin Biere

Institute for Formal Models and Verification (FMV)
Johannes Kepler University, Linz, Austria
http://fmv.jku.at

SAT'11
June 19 - 22, 2011
Ann Arbor, Michigan, USA

Failed Literal Detection (FL) for Preprocessing

- Established technique in SAT solving. What about QBF?
- This work: three (two novel) FL strategies for QBF.



Overview

Part 1: Preliminaries

- From propositional logic (SAT) to QBF.
- QBF semantics: assignment trees.

Part 2: Failed Literal Detection (FL)

- Necessary assignments and QBF models.
- Three FL approaches and related practical aspects.
- Incomparability results.
- Experiments.

Part 1: Preliminaries

Propositional Logic (SAT):

- Our focus: formulae in conjunctive normal form (CNF).
- Set of Boolean variables $V := \{x_1, \ldots, x_m\}$.
- Literals $l := v$ or $l := \neg v$ for $v \in V$.
- Clauses $C_i := (l_1 \vee \ldots \vee l_{k_i})$.
- CNF $\phi := \bigwedge C_i$.

Quantified Boolean Formulae (QBF):

- Prenex CNF: quantifier-free CNF over quantified Boolean variables.
- PCNF $Q_1 S_1 \ldots Q_n S_n. \phi$, where $Q_i \in \{\exists, \forall\}$, scopes S_i.
- Scope S_i: set of quantified variables.
- $Q_i S_i \leq Q_{i+1} S_{i+1}$: scopes are linearly ordered.

Example

Clauses (CNFs) are sets of literals (clauses).
A CNF: $\{x, \overline{y}\}, \{\overline{x}, y\}$ and a PCNF: $\forall x \exists y. \{x, \overline{y}\}, \{\overline{x}, y\}$.

Assignment Trees (AT):

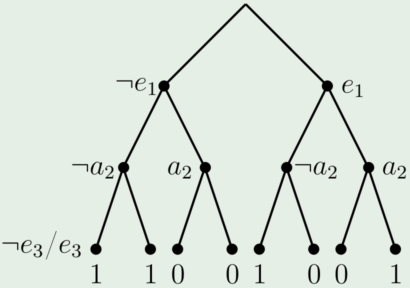
- Assignment $A : V \rightarrow \{true, false\}$ maps variables to truth values.
- Paths from root to a leaf in AT represent assignments.
- Nodes along path (except root) assign truth values to variables.

CNF-Model:

- A path in the assignment tree of a CNF ϕ which satisfies all clauses.
- CNF ϕ is satisfiable iff it has a CNF-model *m*: $m \models \phi$.

Example

$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$



SAT Semantics

Assignment Trees (AT):

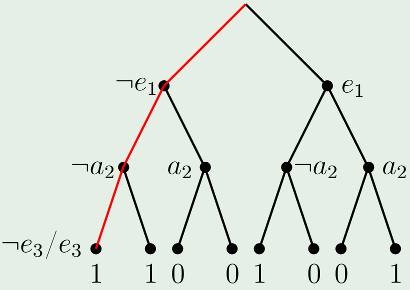
- Assignment $A : V \rightarrow \{true, false\}$ maps variables to truth values.
- Paths from root to a leaf in AT represent assignments.
- Nodes along path (except root) assign truth values to variables.

CNF-Model:

- A path in the assignment tree of a CNF ϕ which satisfies all clauses.
- CNF ϕ is satisfiable iff it has a CNF-model m: $m \models \phi$.

Example

$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$



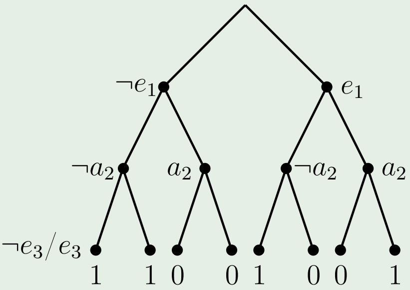
PCNF-Model: $\psi := Q_1 S_1 \dots Q_n S_n.\ \phi$

- An (incomplete) AT where *every* path is a CNF-model of CNF part ϕ.
- Restriction: nodes which assign $\forall$-variables have exactly one sibling.
- PCNF ψ is satisfiable iff it has a PCNF-model m: $m \models \psi$.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3.\ \phi$$

$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$

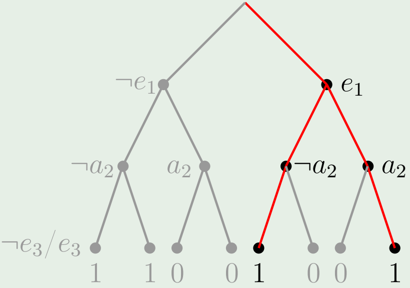


PCNF-Model: $\psi := Q_1 S_1 \dots Q_n S_n.\ \phi$

- An (incomplete) AT where *every* path is a CNF-model of CNF part ϕ.
- Restriction: nodes which assign $\forall$-variables have exactly one sibling.
- PCNF ψ is satisfiable iff it has a PCNF-model m: $m \models \psi$.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3.\ \phi$$
$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$



QBF Propagation Rules (1/5)

Definition (Assignments of Literals)

Given a PCNF ψ, the *assignment of a literal l* yields the formula $\psi[l]$ where clauses $Occs(l)$ and literals $\neg l$ in $Occs(\neg l)$ are deleted.

Example

$$\begin{aligned}
\psi &:= \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi \\
\phi &:= \{e_1, a_2, e_3, e_4\}, \\
&\quad \{e_1, a_2, \neg e_4\}, \\
&\quad \{\neg e_1, e_3, \neg e_4\}, \\
&\quad \{\neg a_2, \neg e_3\}
\end{aligned}$$

$\psi[e_4]$

QBF Propagation Rules (2/5)

Definition (Model-equivalence)

Two PCNFs ψ and ψ' are *model-equivalent*, written as $\psi \equiv_m \psi'$, iff for all assignment trees t: $t \models \psi$ iff $t \models \psi'$.

Definition (Universal Reduction)

Given a clause C, $UR(C) := C \setminus \{l_u \in L_\forall(C) \mid \nexists l_e \in L_\exists(C), l_u < l_e\}$, i.e. deleting universal literals which are “tailing” by quantifier ordering.
Given PCNF ψ and clause C. Then $\psi \wedge C \equiv_m \psi \wedge UR(C)$ [SDB06].

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi := \{e_1, a_2\}, \{\neg e_1, e_3\}, \{\neg a_2, \neg e_3\}$$

$UR(\{e_1, a_2\})$

QBF Propagation Rules (3/5)

Definition (Satisfiability-equivalence)

Two PCNFs ψ and ψ' are *satisfiability-equivalent*, written as $\psi \equiv_s \psi'$, iff: if ψ is satisfiable then ψ' is satisfiable and vice versa.

Definition (Pure Literal Rule)

Given a PCNF ψ, a literal l where $Occs(l) \neq \emptyset$ and $Occs(\neg l) = \emptyset$ is *pure*: if $q(l) = \exists$ then $\psi \equiv_s \psi[l]$, and if $q(l) = \forall$ then $\psi \equiv_s \psi[\neg l]$.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi := \{e_1\}, \{\neg e_1, e_3\}, \{\neg a_2, \neg e_3\}$$

Variable a_2 is pure: $\psi[a_2]$ (shortening clauses).

Definition (Unit Clause Rule)

Given a PCNF ψ. A clause $C \in \psi$ where $UR(C) = \{l\}$ is *unit* and $\psi \equiv_s \psi[l]$.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi := \{e_1\}, \{\neg e_1, e_3\}, \{\neg e_3\}$$

Clauses $\{e_1\}$ and $\{\neg e_3\}$ are unit: $\psi[e_1][\neg e_3]$.

QBF Propagation Rules (5/5)

Definition (Quantified Boolean Constraint Propagation)

Given a PCNF ψ and a literal x called *assumption*. Formula $QBCP(\psi, x)$ is obtained from $\psi[x]$ by applying UR, unit clause and pure literal rule.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi :=$$

$$\{\}$$

Empty clause derived from assumption e_4:

$\emptyset \in QBCP(\psi, e_4)$.

Part 2: Failed Literal Detection (FL)

Definition

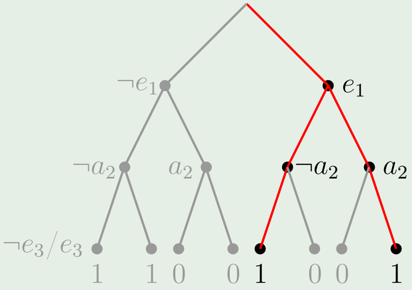
Given PCNF ψ and $x_i \in V$. Assignment $x_i \mapsto v$, where $v \in \{false, true\}$, is necessary for satisfiability of ψ iff $x_i \mapsto v$ is part of *every* path in *every* PCNF-model of ψ.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3 .\ \phi$$

$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$

- $e_1 \mapsto$ *true* is necessary for satisfiability of ψ.



Goal: Detection of (Subset of) Necessary Assignments in QBFs.

- Exponential reduction of search space.

Models and Necessary Assignments

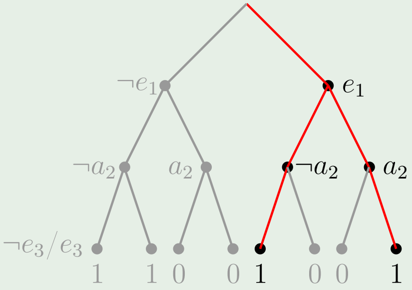
Definition

Given PCNF ψ and $x_i \in V$. Assignment $x_i \mapsto v$, where $v \in \{\mathit{false}, \mathit{true}\}$, is necessary for satisfiability of ψ iff $x_i \mapsto v$ is part of *every* path in *every* PCNF-model of ψ.

Example

$$\psi := \exists e_1 \forall a_2 \exists e_3.\ \phi$$
$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$

- $e_1 \mapsto \mathit{true}$ is necessary for satisfiability of ψ.



Goal: Detection of (Subset of) Necessary Assignments in QBFs.

- Exponential reduction of search space.

Failed Literal Detection (FL) for SAT:

- BCP-based approach to detect subset of necessary assignments.
- Def. failed literal x for CNF ϕ: if $\emptyset \in BCP(\phi, x)$ then $\phi \equiv_m \phi \wedge \{\neg x\}$.
- FL based on deriving empty clause from assumption and BCP.

FL for QBF:

- Def.: failed literal x for PCNF ψ: if $\psi \equiv_m \psi \wedge \{\neg x\}$.
- QBCP-based approach like for SAT is unsound due to $\exists/\forall$ prefix.

Example

$\psi := \forall x \exists y. \{x, \neg y\}, \{\neg x, y\}$. We have $\emptyset \in QBCP(\psi, y)$ but $\psi \not\equiv_s \psi \wedge \{\neg y\}$.

Our Work:

- Three (one known, two novel) FL approaches for QBF.
- Soundness established by SAT-testing, abstraction and Q-resolution.
- QBCP for efficiency.

Motivation

Failed Literal Detection (FL) for SAT:

- BCP-based approach to detect subset of necessary assignments.
- Def. failed literal x for CNF ϕ: if $\emptyset \in BCP(\phi, x)$ then $\phi \equiv_m \phi \wedge \{\neg x\}$.
- FL based on deriving empty clause from assumption and BCP.

FL for QBF:

- Def.: failed literal x for PCNF ψ: if $\psi \equiv_m \psi \wedge \{\neg x\}$.
- QBCP-based approach like for SAT is unsound due to $\exists/\forall$ prefix.

Example

$\psi := \forall x \exists y. \{x, \neg y\}, \{\neg x, y\}$. We have $\emptyset \in QBCP(\psi, y)$ but $\psi \not\equiv_s \psi \wedge \{\neg y\}$.

Our Work:

- Three (one known, two novel) FL approaches for QBF.
- Soundness established by SAT-testing, abstraction and Q-resolution.
- QBCP for efficiency.

Known Approach: SAT-Based FL for QBF

Theorem

1. *For CNF ϕ and literal x, $\phi \wedge \{\neg x\}$ is unsatisfiable iff $\phi \equiv_m \phi \wedge \{x\}$.*
2. *For CNFs ϕ and ϕ' with $\phi \equiv_m \phi'$, $Q_1 S_1 \ldots Q_n S_n.\ \phi \equiv_m Q_1 S_1 \ldots Q_n S_n.\ \phi'$.*

Full Propositional Satisfiability Testing: [PS10, SB05]

- PCNF $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$: find necessary assignments of CNF ϕ.
- Calling SAT solver: if $\phi \wedge \{\neg x\}$ is unsatisfiable, then $\psi \equiv_m \psi \wedge \{x\}$.
- Failed literals learnt (by SAT solver) on CNF can be added to PCNF.
- Drawback: exponential-time.

Example

$\psi := \exists e_1 \forall a_2 \exists e_3.\ \{e_1, a_2, e_3\}, \{e_1, a_2, \neg e_3\}, \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}$. A SAT solver will find out that the CNF part with assumption $\neg e_1$ is unsatisfiable, hence $\{e_1\}$ can be learnt.

Abstraction-Based FL (1/2)

Problems:

- SAT-testing is expensive: use QBCP instead.
- $QBCP(\psi, x)$ with assumption x for FL on PCNF ψ is unsound.

Definition (Quantifier Abstraction)

For $\psi := Q_1 S_1 \ldots Q_{i-1} S_{i-1} Q_i S_i \ldots\ldots Q_n S_n.\ \phi$, the quantifier abstraction of ψ with respect to S_i is $Abs(\psi, i) := \exists(S_1 \cup \ldots \cup S_{i-1}) Q_i S_i \ldots Q_n S_n.\ \phi$.

Idea: carry out QBCP on abstraction of ψ.

- If $x \in S_i$ then treat all variables smaller than x as existentially quantified.
- Example: $Abs(\exists x \forall y \exists z.\ \phi, 3) = \exists x \exists y \exists z.\ \phi$.
- Overapproximation: if $m \models \psi$ then $m \models Abs(\psi, i)$.

Theorem

Given PCNF $\psi := Q_1 S_1 \dots Q_n S_n.\ \phi$ *and literal* x *where* $v(x) \in S_i$. *If* $\emptyset \in QBCP(Abs(\psi, i), x)$ *then* $\psi \equiv_m \psi \wedge \{\neg x\}$.

Practical Application:

- FL using QBCP on abstraction is sound and runs in polynomial-time.

Example

$\psi := \forall a_1 \exists e_2, e_3 \forall a_4 \exists e_5.\ \{a_1, e_2\}, \{\neg a_1, e_3\}, \{e_3, \neg e_5\}, \{a_1, e_2, \neg e_3\}, \{\neg e_2, a_4, e_5\}$. We get $\emptyset \in QBCP(Abs(\psi, 2), \neg e_3)$: $\neg a_1$ (treated as existential), $\neg e_5$ and $\neg e_2$ (due to UR) are unit, $\{a_1, e_2\}$ is empty. Unit clause $\{e_3\}$ is learnt.

Drawback:

- $Abs(\psi, i)$ generally has fewer universal variables than ψ.
- QBCP on $Abs(\psi, i)$ is possibly “weaker” than on ψ (see slide 26).

QBCP-Guided Q-Resolution (1/2)

Definition (Q-resolution)

Let C_1, C_2 be clauses with $v \in C_1, \neg v \in C_2$ and $q(v) = \exists$ [BKF95].

1. $C_1 \otimes C_2 := (UR(C_1) \cup UR(C_2)) \setminus \{v, \neg v\}$.
2. If $\{x, \neg x\} \subseteq C_1 \otimes C_2$ (tautology) then no Q-resolvent exists.
3. Otherwise, Q-resolvent $C := UR(C_1 \otimes C_2)$ of C_1 and C_2 on v: $\{C_1, C_2\} \vdash^* C$.

Q-Resolution: combination of propositional resolution and UR.

- For PCNF ψ, clause C: if $\psi \vdash^* C$ then $\psi \equiv_m \psi \wedge C$ [BL99, SDB06].

Idea: heuristically validate $\emptyset \in QBCP(\psi, x)$ on *original* PCNF.

- Try to derive the negated assumption $\{\neg x\}$ by Q-resolution.
- Resolution candidates are selected from clauses “touched” by QBCP.
- Like conflict-driven clause learning (CDCL) in search-based solvers.

QBCP-Guided Q-Resolution (2/2)

Corollary

Given PCNF $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$ *and literal* x *where* $v(x) \in S_i$. *If* $\emptyset \in QBCP(\psi, x)$ *and* $\psi \vdash^* \{\neg x\}$ *then* $\psi \equiv_m \psi \wedge \{\neg x\}$.

Example

$\psi := \exists e_1, e_2 \forall a_3 \exists e_4, e_5.\ \{a_3, e_5\}, \{\neg e_2, e_4\}, \{\neg e_1, e_4\}, \{e_1, e_2, \neg e_5\}$.
We get $\emptyset \in QBCP(\psi, \neg e_4)$: $\neg e_1$, $\neg e_2$ and $\neg e_5$ are unit, $\{a_3, e_5\}$ is empty.
Unit clause $\{e_4\}$ is derived by resolving clauses in reverse-chronological order: $(\{a_3, e_5\}, \{e_1, e_2, \neg e_5\}) \vdash \{e_1, e_2\}$, $(\{e_1, e_2\}, \{\neg e_2, e_4\}) \vdash \{e_1, e_4\}$, $(\{e_1, e_4\}, \{\neg e_1, e_4\}) \vdash \{e_4\}$.

Practical Application:

- Advantage: original prefix allows full propagation power in QBCP.
- QBCP-based selection of resolution candidates is only a heuristic.

Comparing FL Approaches (1/2)

Proposition

SAT-based FL, abstraction-based FL and QBCP-guided Q-resolution are incomparable to each other with respect to detecting necessary assignments.

Consequences:

- There are PCNFs where one approach can detect a necessary assignment the other one cannot.
- No approach can detect all necessary assignments.
- Crucial: Q-resolution for CDCL is not optimal (see also [SB05]).
- (How) Can we apply quantifier abstraction for clause learning?

Comparing FL Approaches (2/2) – Example

Proposition

Abstraction-based FL and QBCP-guided Q-resolution are incomparable to each other with respect to detecting necessary assignments.

(Examples for other combinations may be found in our paper).

Example

$\psi := \forall a_1 \exists e_2, e_3 \forall a_4 \exists e_5.\ \{a_1, e_2\}, \{\neg a_1, e_3\}, \{e_3, \neg e_5\}, \{a_1, e_2, \neg e_3\}, \{\neg e_2, a_4, e_5\}$. We have $\emptyset \in QBCP(Abs(\psi, 2), \neg e_3)$ but $\psi \not\vdash^* \{e_3\}$: assignment $\{e_3\} \mapsto$ *true* is necessary but Q-resolution *cannot* derive $\{e_3\}$.

Example (same as on slide 24)

$\psi := \exists e_1, e_2 \forall a_3 \exists e_4, e_5.\ \{a_3, e_5\}, \{\neg e_2, e_4\}, \{\neg e_1, e_4\}, \{e_1, e_2, \neg e_5\}$.
We get $\emptyset \in QBCP(\psi, \neg e_4)$ and also $\psi \vdash^* \{e_4\}$ from touched clauses.
But $\emptyset \notin QBCP(Abs(\psi, 3), \neg e_4)$: due to abstraction, UR is not applicable to make $\{a_3, e_5\}$ empty.

Experiments: Solver Performance

Preprocessor "QxBF": FL-based preprocessor operating in rounds.

- Abstraction-based FL ("ABS"), QBCP-guided Q-resolution ("QRES"), and SAT-based FL ("SAT"). At most 80 seconds spent on FL.

Preprocessor "bloqqer":

- Joint work with Armin Biere and Martina Seidl:
  "Blocked Clause Elimination for QBF (QBCE)". In *Proc. CADE 2011*.
- BCE and related techniques [JBH10, HJB10] applied to QBF.

Preprocessor "sQueezeBF": part of QuBE solver [GMN10a, GMN10b].

DepQBF on QBFEVAL'10 (568 formulae, 900 sec. time limit)				
Preprocessing	*Solved*	*Time (Preprocessing)*	*SAT*	*UNSAT*
bloqqer+(ABS+SAT)	468	197.31 (16.47)	224	244
bloqqer	466	198.50 (7.69)	223	243
sQueezeBF	435	233.28 (36.94)	201	234
sQueezeBF+(ABS+SAT)	434	239.84 (42.79)	201	233
SAT	379	322.31 (7.17)	167	212
ABS+SAT	378	323.19 (7.21)	167	211
ABS	375	327.64 (3.33)	168	207
QRES	374	327.63 (1.83)	167	207
None	372	334.60 (—)	166	206

Experiments: FL and other Preprocessors

DepQBF on QBFEVAL'10 (568 formulae, 900 sec. time limit)		
Preprocessing	*Solved*	*Solved by Preprocessing (% of Solved)*
bloqqer+(ABS+SAT)	468	172 (36.7%)
bloqqer	466	148 (31.7%)
sQueezeBF	435	39 (8.9%)
sQueezeBF+(ABS+SAT)	434	64 (14.7%)
ABS+SAT	378	30 (7.9%)

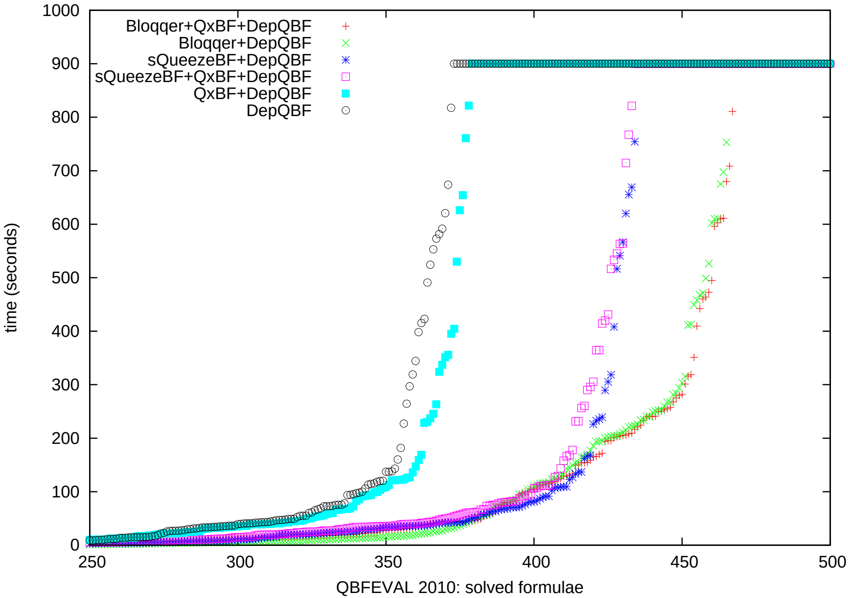
Notes:

- (ABS+SAT) only solved 2 instances which bloqqer did not solve, and 120 in other direction.
- (ABS+SAT) only solved 12 instances which sQueezeBF did not solve, and 21 in other direction.

On formulae not solved by preprocessing:

- (ABS+SAT) only fixed avg. 1348 assignments.
- After sQueezeBF, (ABS+SAT) still fixed avg. 186 assignments.
- After bloqqer, (ABS+SAT) still fixed avg. 16 assignments.

Times Plot: DepQBF with Different Preprocessors



Experiments: Incomparability

QBFEVAL'10: 524 formulae completed by all				
Preprocessing	None	ABS	QRES	SAT
Avg. Fixed	607.17	730.31	724.10	715.77
Med. Fixed	103.5	137.00	135.00	181.50
Avg. Time	0.02	3.19	0.76	10.80
Med. Time	0.00	0.16	0.02	0.20
Avg. Props/As	—	118.80	51.08	—
Med. Props/As	—	40.01	6.68	—

Table: Average and median run times, fixed assignments, and propagations per assumption for FL approaches. "None" is QBCP on original formula only.

QBFEVAL'10: formulae with different fixed assignments (FAs)						
	ABS vs. QRES		ABS vs. SAT		QRES vs. SAT	
Formulae with Diff. FAs	130		183		220	
Formulae wrt. Unique FAs	121	9	57	126	36	180
Total Unique FAs	3752	58	24268	16648	24237	19874
Avg. Unique FAs	28.86	0.44	132.61	90.97	110.16	90.33
Med. Unique FAs	1	0	0	13	0	5
Avg. Diff. in Unique FAs	28.41		41.63		19.83	
Med. Diff. in Unique FAs	1		-14		-4.5	

Table: Pairwise comparison of FL approaches (complete runs).

Conclusions

Failed Literal Detection (FL):

- Necessary assignments: exponential reduction of search-space.
- Soundness of FL for QBF: abstraction, Q-resolution, and SAT-testing.
- Future work: dynamic applications of FL.

Incomparability of FL Approaches:

- Not just theory, but also shows up in practice.
- CDCL by Q-resolution in QBF is not optimal, see also [SB05].

Experiments:

- Positive effects of FL on elimination- and search-based QBF solvers.
- FL is complementary to state-of-the-art preprocessing techniques.
- Up to 30% of QBFEVAL'10 formulae solved by preprocessing.
- See also our CADE 2011 paper, and bloqqer:
  `http://fmv.jku.at/bloqqer/`

QxBF is Open Source: `http://fmv.jku.at/qxbf/`

References

U. Bubeck and H. Kleine Büning.
Bounded Universal Expansion for Preprocessing QBF.
In J. Marques-Silva and K. A. Sakallah, editors, *SAT*, volume 4501 of *LNCS*, pages 244–257. Springer, 2007.

D. Le Berre.
Exploiting the Real Power of Unit Propagation Lookahead.
Electronic Notes in Discrete Mathematics, 9:59–80, 2001.

A. Biere.
Resolve and Expand.
In H. H. Hoos and D. G. Mitchell, editors, *SAT (Selected Papers)*, volume 3542 of *LNCS*, pages 59–70. Springer, 2004.

H. Kleine Büning, M. Karpinski, and A. Flögel.
Resolution for Quantified Boolean Formulas.
Inf. Comput., 117(1):12–18, 1995.

H. Kleine Büning and T. Lettmann.
Propositional Logic: Deduction and Algorithms.
Cambridge University Press, New York, NY, USA, 1999.

F. Bacchus and J. Winter.
Effective Preprocessing with Hyper-Resolution and Equality Reduction.

In E. Giunchiglia and A. Tacchella, editors, *SAT*, volume 2919 of *LNCS*, pages 341–355. Springer, 2003.

E. Giunchiglia, P. Marin, and M. Narizzano.
QuBE7.0 (System Description).
JSAT, 7:83–88, 2010.

E. Giunchiglia, P. Marin, and M. Narizzano.
sQueezeBF: An Effective Preprocessor for QBFs Based on Equivalence Reasoning.
In O. Strichman and S. Szeider, editors, *SAT*, volume 6175 of *LNCS*, pages 85 – 98. Springer, 2010.

M. Heule, M. Järvisalo, and A. Biere.
Clause Elimination Procedures for CNF Formulas.
In C. G. Fermüller and A. Voronkov, editors, *LPAR (Yogyakarta)*, volume 6397 of *Lecture Notes in Computer Science*, pages 357–371. Springer, 2010.

M. Järvisalo, A. Biere, and M. Heule.
Blocked Clause Elimination.
In J. Esparza and R. Majumdar, editors, *TACAS*, volume 6015 of *LNCS*, pages 129–144. Springer, 2010.

I. Lynce and J. P. Marques Silva.
Probing-Based Preprocessing Techniques for Propositional Satisfiability.

In *ICTAI*, pages 105–. IEEE Computer Society, 2003.

F. Pigorsch and C. Scholl.
An AIG-Based QBF-Solver Using SAT for Preprocessing.
In S. S. Sapatnekar, editor, *DAC*, pages 170–175. ACM, 2010.

J. Rintanen.
Improvements to the Evaluation of Quantified Boolean Formulae.
In T. Dean, editor, *IJCAI*, pages 1192–1197. Morgan Kaufmann, 1999.

H. Samulowitz and F. Bacchus.
Using SAT in QBF.
In P. van Beek, editor, *CP*, volume 3709 of *LNCS*, pages 578–592. Springer, 2005.

H. Samulowitz and F. Bacchus.
Binary Clause Reasoning in QBF.
In A. Biere and C. P. Gomes, editors, *SAT*, volume 4121 of *LNCS*, pages 353–367. Springer, 2006.

H. Samulowitz, J. Davies, and F. Bacchus.
Preprocessing QBF.
In F. Benhamou, editor, *CP*, volume 4204 of *LNCS*, pages 514–529. Springer, 2006.