Resolve and Expand

May 13, 2004

Armin Biere
Computer Systems Institute
Department of Computer Science
ETH Zürich, Switzerland

SAT'04

Seventh International Conference on

Theory and Applications of Satisfiability Testing

May 10–13, 2004, Vancouver, BC, Canada

T boolean formula encoding a (finite transition) relation

$$[[T]] \subseteq \{0,1\}^n \times \{0,1\}^n$$

Transitive Closure

$$T^* \equiv T^{2^n}$$

Standard Linear Unfolding

$$T^{i+1}(s,t) \equiv \exists\, m.\, T^i(s,m) \wedge T(m,t)$$

Iterative Squaring via Copying

$$T^{2\cdot i}(s,t) \equiv \exists\, m.\, T^i(s,m) \wedge T^i(m,t)$$

Non Copying Iterative Squaring

$$T^{2\cdot i}(s,t) \equiv \exists\, m.\, \forall c.\, \exists\, l,r.\; (c \longrightarrow (l,r) = (s,m)) \wedge (\overline{c} \longrightarrow (l,r) = (m,t)) \wedge T^i(l,r)$$

Why is QBF harder than SAT?

$$\models \quad \forall x \,.\, \exists y \,.\, (x \leftrightarrow y)$$

$$\not\models \quad \exists y \,.\, \forall x \,.\, (x \leftrightarrow y)$$

Decision Order Matters!

solve-sat(assignment) [DavisLogemannLoveland62]

boolean-constraint-propagation()

if contains-empty-clause() **then return** *false*

if no-clause-left() **then return** *true*

v := next-unassigned-variable()

return solve-sat(S $\cup \{v \mapsto$ *false*$\}$) $\vee$ solve-sat(S $\cup \{v \mapsto$ *true*$\}$)

solve-qbf(assignment) [CadoliGiovanardiSchaerf98]

boolean-constraint-propagation()

if contains-empty-clause() **then return** *false*

if no-clause-left() **then return** *true*

v := next-outermost-unassigned-variable()

@ := is-existential(v) ? $\vee$: $\wedge$

return solve-sat(S $\cup \{v \mapsto$ *false*$\}$) @ solve-sat(S $\cup \{v \mapsto$ *true*$\}$)

- almost all implementations are QBF-enhanced DPLL: [Cadoli...98] [Rintanen01]
  - recently **learning** was added [Giunchiglia...01] [Letz01] [ZhangMalik02]
  - all deterministic solvers (except one) in QBF-Evaluation'03 were DPLL based
  - top-down: split on variables from the outside to the inside
- multiple quantifier elimination procedures:
  - **enumeration** [PlaistedBiereZhu03] [McMillan02]
  - **expansion** [Aziz-Abdulla...00] [WilliamsBiere...00] [AyariBasin02]
  - bottom-up: eliminate variables from the inside to the outside
- **q-resolution** [Kleine-Büning...95]

Forall Reduction and Q-Resolution

- **collect** variables in scopes, **order** variables and scopes according to nesting depth:

$$\underbrace{\exists\, a,b,c,d.}_{\text{scope 0}} \quad \underbrace{\forall x,y,z.}_{\text{scope 1}} \quad \underbrace{\exists\, r,s,t.}_{\text{scope 2}} \quad (c \vee d)(a \vee \overline{c} \vee \overline{x} \vee y)(\overline{a} \vee x \vee s)(t \vee \ldots)\ \cdots$$

attach clauses to the scope of its innermost variables

- **remove** innermost universal literals in clauses attached to universal scopes:

$$(a \vee \overline{c} \vee \overline{x} \vee y) \quad \text{simplifies to} \quad (a \vee \overline{c})$$

- q-resolution = resolution + forall reduction

Invariant

- all clauses are forall reduced

  $\Rightarrow$ innermost scope is always existential

  $\Rightarrow$ no clauses attached to universal scopes

- normalized structure of quantified CNF:

$$\Omega(S_1)\, S_1 \,.\quad \Omega(S_2)\, S_2 \,.\quad \ldots\quad \forall S_{m-1}\,.\quad \exists\, S_m\,.\quad f \wedge g \qquad m \geq 2$$

$f \equiv$ clauses of scope S_m

$g \equiv$ clauses of outer scopes $S_i,\quad i < m-1$

$S_\exists \equiv S_m$

$S_\forall \equiv S_{m-1}$

resolve-and-expand()

forever

simplify()

if contains-empty-clause() **then return** *false*

if no-clause-left() **then return** *true*

if is-propositional() **then return** sat-solve($\emptyset$)

v := schedule-cheapest-to-eliminate-variable()

if is-existential(v) **then** resolve(v)

if is-universal(v) **then** expand(v)

original clauses in which v or $\neg v$ occurs:

$\neg r \vee v$

$s \vee v$

$x \vee y \vee v$

$\boxed{\times}$

$\neg v \vee r$

$\neg v \vee \neg x \vee \neg y \vee r$

add forall reduced non-trivial resolvents:

$(s \vee r)$, $(x \vee y \vee r)$, and $(s \vee \neg x \vee \neg y \vee r)$

remove original clauses

one-to-one mapping of variables: $u \in S_\exists$ mapped to $u' \in S'_\exists$

before expansion:

$$\Omega(S_1)\, S_1 \,.\quad \Omega(S_2)\, S_2 \,.\quad \ldots\quad \forall S_\forall \qquad .\quad \exists\, S_\exists \qquad . \qquad f \wedge g$$

after expansion:

$$\Omega(S_1)\, S_1 \,.\quad \Omega(S_2)\, S_2 \,.\quad \ldots\quad \forall (S_\forall - \{v\}).\quad \exists (S_\exists \cup S'_\exists).\quad f\{v/0\} \,\wedge\, f'\{v/1\} \wedge g$$

- elimination cost: number of expected added literals

$$
\begin{aligned}
o(l) &\equiv \text{number of clauses with literal } l \\
s(l) &\equiv \text{sum of lengths of clauses with literal } l \\
s(S) &\equiv \text{sum lengths of clauses with scope } S
\end{aligned}
$$

- expansion cost: $\mathrm{s}(\mathrm{S}_{\exists}) \quad - \quad \Big(\mathrm{s}(\mathrm{v}) + \mathrm{s}(\neg\mathrm{v}) + \mathrm{o}(\mathrm{v}) + \mathrm{o}(\neg\mathrm{v})\Big)$

- resolution cost: $\mathrm{o}(\neg\mathrm{v}) \cdot \Big(\mathrm{s}(\mathrm{v}) - \mathrm{o}(\mathrm{v})\Big) + \mathrm{o}(\mathrm{v}) \cdot \Big(\mathrm{s}(\neg\mathrm{v}) - \mathrm{o}(\neg\mathrm{v})\Big) - \Big(\mathrm{s}(\mathrm{v}) + \mathrm{s}(\neg\mathrm{v})\Big)$

Benchmarking Structured Instances of SAT'03 QBF Evaluation

benchmark family		#inst	**decide**	**qube**	**semprop**	*expand*	**quantor**
1	adder*	16	2	2	2	1	**3**
2	Adder2*	14	2	2	2	2	**3**
3	C[0-9]*	27	2	3	2	3	**4**
4	CHAIN*	11	10	7	**11**	4	**11**
5	comp*	5	4	4	**5**	**5**	**5**
6	flip*	7	6	**7**	**7**	**7**	**7**
7	impl*	16	12	**16**	**16**	**16**	**16**
8	k*	171	77	91	97	60	**108**
9	mutex*	2	1	**2**	**2**	**2**	**2**
10	robots*	48	0	**36**	**36**	15	24
11	term1*	4	2	**3**	**3**	1	**3**
12	toilet*	260	187	**260**	**260**	259	259
13	TOILET*	8	**8**	6	**8**	**8**	**8**
14	tree*	12	10	**12**	**12**	8	**12**
#(among best in family)			**1**	**7**	**10**	**5**	**12**
#(single best in family)			**0**	**0**	**0**	**0**	**4**

(families with no difference and two actually random families removed)

Simplify

- resolve quadratic in number of occurrences, expand may double the size

  $\Rightarrow$ simplify CNF as much as possible before elimination

- standard simplification: **unit propagation**, **pure literal rule**, **forall reduction**

- **equivalence reasoning**: extract bi-implications and substitute variables

$$\forall x\,.\exists y\,.(x \vee y)(x \rightarrow y)(y \rightarrow x) \quad \equiv \quad \forall x\,.\exists y\,.(x \vee y)(x = y) \quad \equiv \quad \forall x\,.\exists y\,.(x \vee x) \quad \equiv \quad 0$$

- **subsumption**: remove subsumed clauses
  - backward subsumption is checked on-the-fly whenever a clause is added
  - forward subsumption is expensive and only checked before expensive operations

Solved Hard Instances of SAT'03 QBF Evaluation: QUANTOR

	hard instance	time	space	∀	∃	units	pure	subsu.	subst.	∀red.
1	Adder2-6-s	29.6	19.7	90	13732	126	13282	174081	0	37268
2	adder-4-sat	0.2	2.8	42	1618	0	884	6487	0	960
3	adder-6-sat	36.6	22.7	90	13926	0	7290	197091	0	54174
4	C49*1.*_0_0*	27.9	13.3	1	579	0	0	48	84	0
5	C5*1.*_0_0*	56.2	16.0	2	2288	10	0	4552	2494	0
6	k_path_n-15	0.1	0.8	32	977	66	82	2369	2	547
7	k_path_n-16	0.1	0.8	34	1042	69	85	2567	2	597
8	k_path_n-17	0.1	0.9	36	1087	72	100	3020	2	639
9	k_path_n-18	0.1	0.9	36	1146	76	106	3242	2	725
10	k_path_n-20	0.1	0.9	38	1240	84	149	3967	2	855
11	k_path_n-21	0.1	1.0	40	1318	84	130	4470	2	909
12	k_t4p_n-7	15.5	105.8	43	88145	138	58674	760844	8	215
13	k_t4p_p-8	5.8	178.6	29	12798	206	5012	85911	4	138
14	k_t4p_p-9	0.3	4.5	32	4179	137	1389	23344	10	142
15	k_t4p_p-10	27.9	152.9	35	130136	193	63876	938973	4	137
16	k_t4p_p-11	86.0	471.5	38	196785	204	79547	1499430	4	140
17	k_t4p_p-15	84.6	354.7	50	240892	169	181676	1336774	9	226
18	k_t4p_p-20	3.6	16.1	65	27388	182	21306	197273	11	325

time in seconds, space in MB

Solved Hard Instances of SAT'03 QBF Evaluation: *EXPAND ONLY*

	hard instance	time	space	∀
1	Adder2-6-s	(12.2)	m.o.	–
2	adder-4-sat	(12.1)	m.o.	–
3	adder-6-sat	(13.0)	m.o.	–
4	C49*1.*_0_0*	98.3	40.8	1
5	C5*1.*_0_0*	357.0	45.6	2
6	k_path_n-15	(16.5)	m.o.	–
7	k_path_n-16	(16.6)	m.o.	–
8	k_path_n-17	(16.2)	m.o.	–
9	k_path_n-18	(16.8)	m.o.	–
10	k_path_n-20	(21.4)	m.o.	–
11	k_path_n-21	(21.0)	m.o.	–
12	k_t4p_n-7	(16.8)	m.o.	–
13	k_t4p_p-8	(21.4)	m.o.	–
14	k_t4p_p-9	(21.2)	m.o.	–
15	k_t4p_p-10	(17.3)	m.o.	–
16	k_t4p_p-11	(17.3)	m.o.	–
17	k_t4p_p-15	(21.3)	m.o.	–
18	k_t4p_p-20	(20.9)	m.o.	–

time in seconds, space in MB, m.o. = memory out ($>$ 1 GB)

Conclusion

- novel **resolution** and **expansion** based QBF decision procedure:
  - simplifications: fast subsumption algorithm, ...
  - resource-driven scheduler
  - efficient implementation: QUANTOR

- future directions:
  - combine bottom-up with top-down
  - QBF more complex than SAT:

    $\Rightarrow$ many and more expensive optimizations possible
  - compact data structures: BDDs or ZBDDs