QuBE 7.0

Solver Description

Paolo Marin
j.w.w. E. Giunchiglia, M. Narizzano

Laboratory of Systems and Technologies for Automated Reasoning (STAR-Lab)
DIST - Univ. Genova - Italy

Pragmatics of SAT, Edinburgh, 2010 July 10th

Quantified Boolean Formulas

- Generalization of propositional logic
- Adds the *quantifiers* to the propositional variables
- Prototypical P–Space Complete problem

$$\varphi = \forall x(\exists y((x \vee y) \wedge \exists z(\neg x \vee y \vee z)))$$

closed QBF in Prenex Conjunctive Normal Form

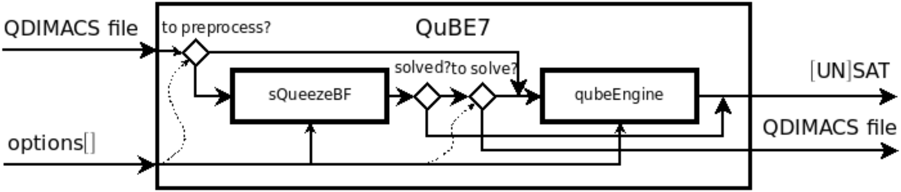
$$\overbrace{Q_1 z_1 \cdots Q_n z_n}^{\text{prefix}} \; \overbrace{\phi(z_1, \ldots, z_n)}^{\text{matrix}} \qquad n \geq 0$$

- Every Q_i ($1 \leq i \leq n$) is a quantifier, either existential $\exists$ or universal $\forall$
- Every z_i is a Boolean variable
- The level of a variable z_i with $j \geq i$ and $Q_j \neq Q_{j+1}$ is the number of alternating quantifier blocks from left to right (starting with 1)
- ϕ is a Boolean formula over the set of variables $\{z_1, \ldots z_n\}$ using standard Boolean connectives and the constants $\perp$ and $\top$

Outline

1. The Tool
2. sQueezeBF
   - Equivalence Reasoning
   - Variable Elimination by Q-Resolution
3. qubeEngine
   - Data Structure
   - Main Algorithms
   - Conflict and solution analysis

QuBE7.0 Architecture



Outline

1. The Tool
2. sQueezeBF
   - Equivalence Reasoning
   - Variable Elimination by Q-Resolution
3. qubeEngine
   - Data Structure
   - Main Algorithms
   - Conflict and solution analysis

sQueezeBF

Why Preprocessing?

- QBF is a powerful extension/generalization of SAT
- for SAT formulae has been proven to be very effective, reducing size, extracting structural info, decreasing the total solving time
- it works very well for QBF too!
- to extend from SAT to QBF is not a trivial task!

GOALS

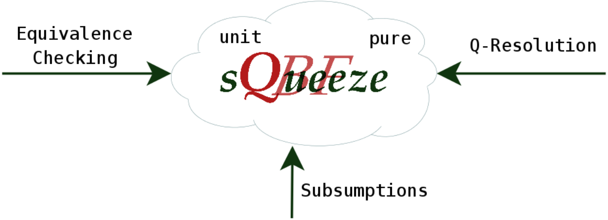
- Decrease as much as possible the size of the formula
- Explicitate hidden information

GOALS

- Decrease as much as possible the size of the formula
- Explicitate hidden information

GOALS

- Decrease as much as possible the size of the formula
- Explicitate hidden information



Preprocessing Loop

function *sQueezeBF*(φ)
  do
    φ' = φ
    φ = *Simplify*(φ)
    φ = *EquivalenceSubstitution*(φ)
    φ = *EquivalenceRewriting*(φ)
    φ = *Q-resolution*(φ)
    if $\varphi \equiv$ TRUE **return** φ
    if $\varphi \equiv$ FALSE **return** φ
  while $\varphi' \neq \varphi$
  return φ

Equivalence Reasoning

Real World Problems contain lots of equivalences:

$$l \equiv l_1 \vee l_2 \quad \Rightarrow \quad (\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2})$$

$$l \equiv l_1 \wedge l_2 \quad \Rightarrow \quad (l \vee \overline{l_1} \vee \overline{l_2}) \wedge (\bar{l} \vee l_1) \wedge (\bar{l} \vee l_2)$$

$$l \equiv \overline{l_1} \quad \Rightarrow \quad (l \vee l_1) \wedge (\bar{l} \vee \overline{l_1})$$

$$l \equiv l_1 \equiv l_2 \quad \Rightarrow \quad (l \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1} \vee \overline{l_2}) \wedge (\bar{l} \vee l_1 \vee \overline{l_2}) \wedge (\bar{l} \vee \overline{l_1} \vee l_2)$$

$$l \equiv \gamma \quad \Rightarrow \quad (l \rightarrow \gamma) \wedge (\gamma \rightarrow l)$$

Equivalence Reasoning (cont'd)

The algorithm works in 2 steps:

1. identification of definitions
2. variable substitution

The size of the formula should not increase in terms of literals

$$dim(\varphi(l)) + dim(l \equiv l_1 \vee l_2) \leq \varphi(l_1 \vee l_2/l) + K$$

Equivalence Reasoning (cont'd)

The algorithm works in 2 steps:

1. identification of definitions
2. variable substitution

The size of the formula should not increase in terms of literals

$$dim(\varphi(l)) + dim(l \equiv l_1 \vee l_2) \leq \varphi(l_1 \vee l_2 / l) + K$$

Equivalence Reasoning (cont'd)

The algorithm works in 2 steps:

1. identification of definitions
2. variable substitution according to an order

If the size of the formula increases?

$$dim(\varphi(l)) + dim(l \equiv l_1 \vee l_2) > \varphi(l_1 \vee l_2 / l) + K$$

$\Rightarrow$ Equivalence Rewriting

Equivalence Rewriting

Transforms the given QBF

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi$$

into the equivalent

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge (\bar{l} \vee \overline{l'}) \wedge \phi$$

where:

$$level(l') = level(l)$$

Q-Resolution

- Given two clauses
  $C_1 = x \vee y_1 \vee ... \vee y_n$; $C_2 = \overline{x} \vee z_1 \vee ... \vee z_m$
  then the q-resolution between C_1 and C_2 over the variable $|x|$ is
  $$C = C_1 \otimes C_2 = y_1 \vee ... \vee y_n \vee z_1 \vee ... \vee z_m$$
- we can think a formula as two distinct sets :
  $\phi = \langle P(X_1, ..., X_n), M(X_1, ..., X_n) \rangle$ where :
  - $P(X_1, ..., X_n)$ is an ordered set of atoms
  - $M(X_1, ..., X_n)$ is a set of clauses over the variables $X_1 ... X_n$
- given a clause C then we define dim(C) = $|C|$
- given a set of clauses S, we define dim(S) = $\sum_{i=1}^{|S|} dim(C_i)$

Variable Elimination by Q-Resolution

Given

- $\phi = \langle P(X_1, ...X_n), M(X_1, ..., X_n)\rangle$
- $S_x \subseteq M : \nexists S'_x \subseteq M$ such that $S'_x \not\subseteq S_x$
- $S_{\overline{x}} \subseteq M : \nexists S'_{\overline{x}} \subseteq M$ such that $S'_{\overline{x}} \not\subseteq S_{\overline{x}}$
- $S = S_x \otimes S_{\overline{x}} = \{C_x \otimes C_{\overline{x}} | C_x \in S_x and C_{\overline{x}} \in S_{\overline{x}}\}$

$$\Downarrow$$

$$\phi = \langle P(X_1, ...X_n) \backslash \{x\}, (M(X_1, ..., X_n) \cup S) \backslash (S_x \cup S_{\overline{x}})\rangle$$

The size of the formula should not increase in terms of literals

$$dim(S) \leq dim(S_x) + dim(S_{\overline{x}}) + \mathrm{K}$$

Clause Elimination by Subsumption

- Given a QBF which matrix is

$$\phi = (x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3)$$

$(x_1 \vee x_2)$ subsumes $(x_1 \vee x_2 \vee x_3)$

del: $(x_1 \vee x_2 \vee x_3)$

- Given a QBF which matrix is

$$\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3)$$

$$(x_1 \vee x_2) \otimes_{x_1} (\bar{x}_1 \vee x_2 \vee x_3) = (x_2 \vee x_3)$$

Q-Resolving the two clauses on x_1 results in $c = (x_2 \vee x_3)$ which subsumes $(\bar{x}_1 \vee x_2 \vee x_3)$, then $(x_1 \vee x_2)$ self-subsumes $(\bar{x}_1 \vee x_2 \vee x_3)$

add: $(x_2 \vee x_3)$
del: $(\bar{x}_1 \vee x_2 \vee x_3)$

Clause Elimination by Subsumption

- Given a QBF which matrix is

$$\phi = (x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3)$$

  $(x_1 \vee x_2)$ subsumes $(x_1 \vee x_2 \vee x_3)$

del: $(x_1 \vee x_2 \vee x_3)$

- Given a QBF which matrix is

$$\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3)$$

$$(x_1 \vee x_2) \otimes_{x_1} (\bar{x}_1 \vee x_2 \vee x_3) = (x_2 \vee x_3)$$

  Q-Resolving the two clauses on x_1 results in $c = (x_2 \vee x_3)$ which subsumes $(\bar{x}_1 \vee x_2 \vee x_3)$, then $(x_1 \vee x_2)$ self-subsumes $(\bar{x}_1 \vee x_2 \vee x_3)$

add: $(x_2 \vee x_3)$
del: $(\bar{x}_1 \vee x_2 \vee x_3)$

Advertisement

Presentation of:

Giunchiglia E., Marin. P., and Narizzano M.

sQueezeBF:

An Effective Preprocessor for QBFs Based on Equivalence Reasoning

SAT'10, Monday, h16:30 @ Appleton Tower, Room LT2

Outline

1. The Tool
2. sQueezeBF
   - Equivalence Reasoning
   - Variable Elimination by Q-Resolution
3. qubeEngine
   - Data Structure
   - Main Algorithms
   - Conflict and solution analysis

The new qubeEngine

Search-based p-cnf QBF solver

- unit, pure & don't care literal propagation
- conflict & solution non-chronological backtracking
- recursive resolution for eliminating tautological-clauses

Data Structure

$\mathbf{W}_1$	$\mathbf{W}_2$	l_0	l_1	$\cdots$	l_n	**0**

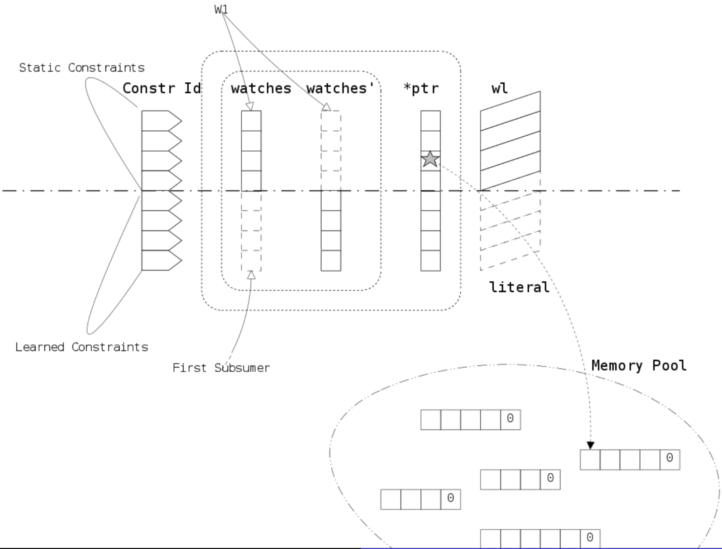
Figure: Constraint Encoding

Q	**C**	b_{28}	b_{27}	$\cdots$	b_0	**S**

Figure: Literal Encoding

- Binary and n-ary clauses are stored separately
- Main search loop designed to be lazy
- New learning algorithm

Data Structure (cont'd)



Search Loop

function qubeEngine(φ)
  $\mu = \emptyset$
  while (TRUE)
    Propagate(μ,φ)
    if (empty $\notin \varphi$)
      μ.*push*(Heuristic(φ))
    if (empty $\in \varphi$)
      Backtrack(μ,φ)
    else if (μ.*top*(*B*) $==$ 0)
      BuildPrimeImplicant(μ,φ)
      Backtrack(μ,φ)
    if (emptyClause $\in \varphi$)
      return FALSE
    if (emptyTerm $\in \varphi$)
      return TRUE

Propagation Loop

function Propagate(μ,φ)
start:
while (($l = \mu.next(B)) \neq 0$)
ResolveBinaries(l,μ,φ)
if (empty $\in \varphi$) **return**
while (($l = \mu.next(S)) \neq 0$)
Subsume(l,φ)
while (($l = \mu.next(N)) \neq 0$)
ResolveNaries(l,μ,φ)
if (empty $\in \varphi$) **return**
if ($\mu.top(B) \neq 0$) **goto** start
while (($l = \mu.next(P)) \neq 0$)
Search4pure(l,μ,φ)
if ($\mu.top(B) \neq 0$) **goto** start

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{y_1 ; x_1 ; y_4 ; y_2 ; y_5 ; x_4 ; y_3 ; x_3\}$$
$$b ; u ; p ; b ; u ; dc ; b ; u$$
$$@ \; 1 ; 1 ; 1 ; 2 ; 2 ; 2 ; 3 ; 3$$

$$C = x_1 \vee y_5 \vee x_3$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{lcccccccc} \mu = \{ & y_1 ; & x_1 ; & y_4 ; & y_2 ; & y_5 ; & x_4 ; & y_3 ; & x_3 \} \\ & b ; & u ; & p ; & b ; & u ; & dc ; & b ; & u \\ @ & 1 ; & 1 ; & 1 ; & 2 ; & 2 ; & 2 ; & 3 ; & 3 \end{array}$$

$$C = x_1 \vee y_5 \vee x_3$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{ccccccccccccccccc} \mu = \{ y_1 & ; & x_1 & ; & y_4 & ; & y_2 & ; & y_5 & ; & x_4 & ; & y_3 & ; & x_3 \} \\ b & ; & u & ; & p & ; & b & ; & u & ; & dc & ; & b & ; & u \\ @ \; 1 & ; & 1 & ; & 1 & ; & 2 & ; & 2 & ; & 2 & ; & 3 & ; & 3 \end{array}$$

$$C = x_1 \vee y_5 \vee x_3$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{rl} \mu = & \{y_1 ; x_1 ; y_4 ; y_2 ; y_5 ; x_4 ; y_3 ; x_3\} \\ & \;\; b ; u ; p ; b ; u ; dc ; b ; u \\ @ & \;\; 1 ; 1 ; 1 ; 2 ; 2 ; 2 ; 3 ; 3 \end{array}$$

$$C = x_1 \vee y_5 \vee x_3 \quad @\text{dl } 2$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{lccccccccccccccc} \mu = \{ & y_1 & ; & x_1 & ; & y_4 & ; & y_2 & ; & y_5 & ; & x_4 & ; & y_3 & ; & x_3 \} \\ & b & ; & u & ; & p & ; & b & ; & u & ; & dc & ; & b & ; & u \\ @ & 1 & ; & 1 & ; & 1 & ; & 2 & ; & 2 & ; & 2 & ; & 3 & ; & 3 \end{array}$$

$$C = x_1 \vee y_5 \vee \neg y_5 \vee x_3$$

Conflicting Literals!

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{lcccccccc} \mu = \{ & y_1 ; & x_1 ; & y_4 ; & y_2 ; & y_5 ; & x_4 ; & y_3 ; & x_3 \} \\ & b ; & u ; & p ; & b ; & u ; & dc ; & b ; & u \\ @ & 1 ; & 1 ; & 1 ; & 2 ; & 2 ; & 2 ; & 3 ; & 3 \end{array}$$

$$C = x_1 \lor y_5 \lor \neg y_5 \lor x_3$$

Conflicting Literals!

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{y_1 ; x_3 ; y_4 ; y_2 ; y_5 ; x_4 ; y_3 ; x_1\}$$

$$b ; u ; p ; b ; u ; dc ; b ; u$$

$$@ \; 1 ; 1 ; 1 ; 2 ; 2 ; 2 ; 3 ; 3$$

$$C = x_1 \vee y_5 \vee x_3$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{ y_1 ; x_3 ; y_4 ; y_2 ; y_5 ; x_4 ; y_3 ; x_1 \}$$

$$b ; u ; p ; b ; u ; dc ; b ; u$$

$$@ \; 1 ; 1 ; 1 ; 2 ; 2 ; 2 ; 3 ; 3$$

$$C = x_1 \vee y_5 \vee x_3$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{rcccccccc} \mu = \{ & y_1 ; & x_3 ; & y_4 ; & y_2 ; & y_5 ; & x_4 ; & y_3 ; & x_1 \} \\ & b ; & u ; & p ; & b ; & u ; & dc ; & b ; & u \\ @ & 1 ; & 1 ; & 1 ; & 2 ; & 2 ; & 2 ; & 3 ; & 3 \end{array}$$

$$C = x_1 \vee y_5 \vee x_3$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{ y_1 \; ; \; x_3 \; ; \; y_4 \; ; \; y_2 \; ; \; y_5 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1 \}$$

$$b \; ; \; u \; ; \; p \; ; \; b \; ; \; u \; ; \; dc \; ; \; b \; ; \; u$$

$$@ \; 1 \; ; \; 1 \; ; \; 1 \; ; \; 2 \; ; \; 2 \; ; \; 2 \; ; \; 3 \; ; \; 3$$

$$C = x_1 \vee y_5 \vee x_3 \quad @\text{dl } 2$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{llllllllll} \mu = \{ & y_1 & ; x_3 & ; y_4 & ; y_2 & ; y_5 & ; x_4 & ; y_3 & ; x_1 \} \\ & b & ; u & ; p & ; b & ; u & ; dc & ; b & ; u \\ @ & 1 & ; 1 & ; 1 & ; 2 & ; 2 & ; 2 & ; 3 & ; 3 \end{array}$$

$$C = x_1 \vee y_5 \vee x_3 \quad \text{@dl } 1$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\begin{array}{lcccccccc} \mu = \{ & y_1 ; & x_3 ; & y_4 ; & y_2 ; & y_5 ; & x_4 ; & y_3 ; & x_1 \} \\ & b ; & u ; & p ; & b ; & u ; & dc ; & b ; & u \\ @ & 1 ; & 1 ; & 1 ; & 2 ; & 2 ; & 2 ; & 3 ; & 3 \end{array}$$

$$C = x_1 \vee y_5 \vee x_3 \quad @\text{dl } 1$$

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{y_1 \; ; \; x_3 \; ; \; y_4 \; ; \; y_2 \; ; \; y_5 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1\}$$

$$b \; ; \; u \; ; \; p \; ; \; b \; ; \; u \; ; \; dc \; ; \; b \; ; \; u$$

$$@ \quad 1 \; ; \; 1 \; ; \; 1 \; ; \; 2 \; ; \; 2 \; ; \; 2 \; ; \; 3 \; ; \; 3$$

$$C = x_1 \vee y_5 \vee \neg y_5 \vee x_3$$

Conflicting Literals!

Conflict and solution analysis

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{y_1 \; ; \; x_3 \; ; \; y_4 \; ; \; y_2 \; ; \; y_5 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1\}$$

$$b \; ; \; u \; ; \; p \; ; \; b \; ; \; u \; ; \; dc \; ; \; b \; ; \; u$$

$$@ \quad 1 \; ; \; 1 \; ; \; 1 \; ; \; 2 \; ; \; 2 \; ; \; 2 \; ; \; 3 \; ; \; 3$$

$$C = x_1 \vee y_5 \vee \neg y_5 \vee x_3$$

Conflicting Literals!

out of order resolution

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{y_1\ ;\ x_2\ ;\ y_4\ ;\ y_5\ ;\ y_2\ ;\ x_4\ ;\ y_3\ ;\ x_1\ ;\ x_3\}$$

$$b\ ;\ u\ ;\ p\ ;\ u\ ;\ b\ ;\ dc\ ;\ b\ ;\ u\ ;\ p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

$$\mu = \{y_1 \ ; \ x_2 \ ; \ y_4 \ ; \ y_5 \ ; \ y_2 \ ; \ x_4 \ ; \ y_3 \ ; \ x_1 \ ; \ x_3\}$$

$$b \ ; \ u \ ; \ p \ ; \ u \ ; \ b \ ; \ dc \ ; \ b \ ; \ u \ ; \ p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$$\mu = \{y_1 \; ; \; x_2 \; ; \; y_4 \; ; \; y_5 \; ; \; y_2 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1 \; ; \; x_3\}$$

b ; u ; p ; u ; b ; dc ; b ; u ; p

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$$\mu = \{y_1 ; x_2 ; y_4 ; y_5 ; y_2 ; x_4 ; y_3 ; x_1 ; x_3\}$$

$$b ; u ; p ; u ; b ; dc ; b ; u ; p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$$\mu = \{ y_1 \; ; \; x_2 \; ; \; y_4 \; ; \; y_5 \; ; \; y_2 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1 \; ; \; x_3 \}$$

b ; u ; p ; u ; b ; dc ; b ; u ; p

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$$\mu = \{y_1 \; ; \; x_2 \; ; \; y_4 \; ; \; y_5 \; ; \; y_2 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1 \; ; \; x_3\}$$

b ; u ; p ; u ; b ; dc ; b ; u ; p

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$$\mu = \{y_1 \; ; \; x_2 \; ; \; y_4 \; ; \; y_5 \; ; \; y_2 \; ; \; x_4 \; ; \; y_3 \; ; \; x_1 \; ; \; x_3\}$$

$$b \; ; \; u \; ; \; p \; ; \; u \; ; \; b \; ; \; dc \; ; \; b \; ; \; u \; ; \; p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$$\mu = \{y_1 ; x_2 ; y_4 ; y_5 ; y_2 ; x_4 ; y_3 ; x_1 ; x_3\}$$

b ; u ; p ; u ; b ; dc ; b ; u ; p

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$\exists \Longrightarrow$ 6 7 8 9

$$\mu = \{y_1 \;;\; x_2 \;;\; y_4 \;;\; y_5 \;;\; y_2 \;;\; x_4 \;;\; y_3 \;;\; x_1 \;;\; x_3\}$$

$$b \;;\; u \;;\; p \;;\; u \;;\; b \;;\; dc \;;\; b \;;\; u \;;\; p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$\exists \Longrightarrow$ 6 7 8 9

$$\mu = \{y_1 \,;\, x_2 \,;\, y_4 \,;\, y_5 \,;\, y_2 \,;\, x_4 \,;\, y_3 \,;\, x_1 \,;\, x_3\}$$

$$b \,;\, u \,;\, p \,;\, u \,;\, b \,;\, dc \,;\, b \,;\, u \,;\, p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$\exists \Longrightarrow$ 6 7 8 9

$$\mu = \{y_1 \ ; x_2 \ ; y_4 \ ; y_5 \ ; y_2 \ ; x_4 \ ; y_3 \ ; x_1 \ ; x_3\}$$

$$b \ ; u \ ; p \ ; u \ ; b \ ; dc \ ; b \ ; u \ ; p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Prime implicant

$$\varphi = \forall y_1 y_2 y_3 \exists x_1 x_2 \forall y_4 y_5 \exists x_3 x_4 \phi$$

5 4 3 2 1 $\Longleftarrow \forall$

$\exists \Longrightarrow$ 6 7 8 9

$$\mu = \{y_1 \;;\; x_2 \;;\; y_4 \;;\; y_5 \;;\; y_2 \;;\; x_4 \;;\; y_3 \;;\; x_1 \;;\; x_3\}$$

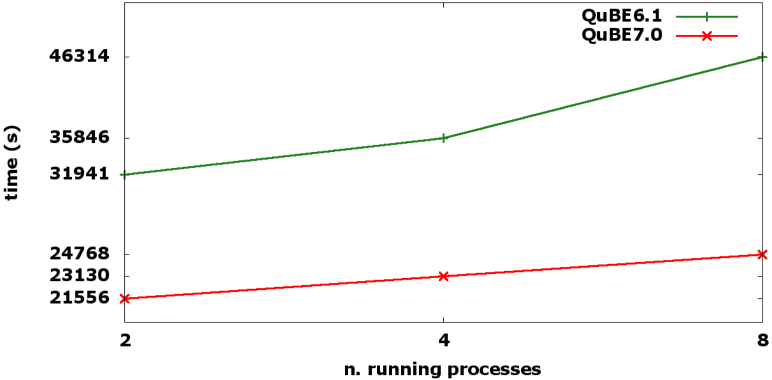
$$b \;;\; u \;;\; p \;;\; u \;;\; b \;;\; dc \;;\; b \;;\; u \;;\; p$$

$$T = y_1 \wedge y_2 \wedge y_3 \wedge x_1 \wedge x_2 \wedge y_4 \wedge y_5 \wedge x_3 \wedge \neg x_4$$

Results

instances	568 fixed-structure, selected for QBFEval'10
hardware I	farm of 9 PCs, intel Core2 Duo 2.13 GHz, 4 GB RAM, GNU Linux Debian 2.6.18.5
hardware II	cluster of 4 IBM HS21 blades, 2x intel Quad Core Xeon 2.5 GHz, 16 GB RAM, GNU Linux CentOS 5
time limit	set to 1200 s

Efficiency on new computer architectures



Results on QBFEval'10 testset

Solver	Total		Sat		Unsat	
	#	Time	#	Time	#	Time
aqme-10	434	32091.1	184	15825.6	250	16265.5
QuBE7.0	**403**	44204.3	180	26342.4	223	17861.9
depqbf	370	21515.3	164	13771.8	206	7743.5
qmaiga	361	43058.1	180	20696.6	181	22361.4
depqbf-pre	356	18995.9	172	12453.8	184	6542.1
AIGSolve	329	22786.6	171	12091.5	158	10695.1
struqs-10	240	32839.7	109	13805.5	131	19034.2
nenofex-qbfeval10	225	13786.9	109	8241.9	116	5545.1
quantor-3.1	205	6711.4	100	4130.6	105	2580.7

Comparison against QuBE6.x on QBFEval'08 testset

Solver	Total	Sat	Unsat
AQME-1NN	2434	977	1457
QuBE7.0+ps	**2367**	901	1466
QuBE7.0	**2277**	852	1425
QuBE6.1	**2144**	828	1316
Nenofex	985	459	526
quantor3.0	972	485	487
ssolve-A	965	450	515
ssolveB	960	450	510
ssolveC	939	444	495

QuBE7.0 ran on a slightly different CPU

Future Work

*QuBE*7.0 is a state-of-the-art QBF Solver

... future improvements:

sQueezeBF

- equivalence reasoning on further types of gates
- dependency schemas
- criteria for stopping

qubeEngine

- progress saving
- learning mechanism/schema
- dependency schemas
- add unlearning strategies
- parallel search

Future Work

*QuBE*7.0 is a state-of-the-art QBF Solver

... future improvements:

sQueezeBF

- equivalence reasoning on further types of gates
- dependency schemas
- criteria for stopping

qubeEngine

- progress saving
- learning mechanism/schema
- dependency schemas
- add unlearning strategies
- parallel search

Future Work

*QuBE*7.0 is a state-of-the-art QBF Solver

... future improvements:

sQueezeBF

- equivalence reasoning on further types of gates
- dependency schemas
- criteria for stopping

qubeEngine

- progress saving
- learning mechanism/schema
- dependency schemas
- add unlearning strategies
- parallel search

Questions?

Thank you

Command line

Available options:

`-ss`	:	enables Selfsubsumption resolution
`-qr`	:	enables Variable Elimination by Q-Resolution
`-es`	:	enables equivalence reasoning on AND/OR gates
`-3e`	:	enables equivalence reasoning on XOR gates
`-er`	:	enables equivalence rewriting
`-all`	:	enables all the above preprocessing techniques
`-noprepro`	:	disables sQueezeBF, and gives the input formula
	:	to qubeEngine
`-solve`	:	solves by using qubeEngine

Pure and Unit Literal Detection

Unit literal

A literal l is unit in φ iff it is the only existential in some clause $c \in \phi$ and all the universal literals $l' \in c$ are s.t. $depth(l') > depth(l)$

Pure literal

An existential (resp. universal) literal l is pure in φ iff $\bar{l} \notin c$ (resp. $l \notin c$) for all clauses $c \in \phi$

Equivalence Substitution

Satisfiability-preserving transformation

$$\bar{l} \leftarrow \overline{l_1} \wedge \overline{l_2}$$
$$l \leftarrow l_1 \vee l_2$$
$$l \equiv l_1 \vee l_2$$

$$\varphi \quad (\bar{l} \vee \overline{l_1} \vee y) \wedge (l \vee \overline{l_2} \vee \overline{y}) \wedge (\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2})$$

$$(\overline{l_1} \vee \overline{l_1} \vee y) \quad (l_1 \vee l_2 \vee \overline{l_2} \vee \overline{y})$$
$$(\overline{l_1} \vee \overline{l_2} \vee y)$$

$$\Downarrow$$
$$\varphi' = \varphi(\gamma / l)$$

$$\Downarrow$$
$$(\overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee y)$$

Equivalence Substitution

Satisfiability-preserving transformation

$$\bar{l} \leftarrow \overline{l_1} \wedge \overline{l_2}$$
$$l \leftarrow l_1 \vee l_2$$
$$l \equiv l_1 \vee l_2$$

$$\varphi \qquad (\bar{l} \vee \overline{l_1} \vee y) \wedge (l \vee \overline{l_2} \vee \bar{y}) \wedge (\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2})$$

$$(\overline{l_1} \vee \overline{l_1} \vee y) \quad (l_1 \vee l_2 \vee \overline{l_2} \vee \bar{y})$$
$$(\overline{l_1} \vee \overline{l_2} \vee y)$$

$$\Downarrow \qquad \Downarrow$$

$$\varphi' = \varphi(\gamma/l) \qquad (\overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee y)$$

Equivalence Substitution

Satisfiability-preserving transformation

$$\bar{l} \leftarrow \bar{l_1} \wedge \bar{l_2}$$
$$l \leftarrow l_1 \vee l_2$$

$$l \equiv l_1 \vee l_2$$

$$\varphi \quad (\bar{l} \vee \bar{l_1} \vee y) \wedge (l \vee \bar{l_2} \vee \bar{y}) \wedge (\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \bar{l_1}) \wedge (l \vee \bar{l_2})$$

$$(\bar{l_1} \vee \bar{l_1} \vee y) \qquad (l_1 \vee l_2 \vee \bar{l_2} \vee \bar{y})$$
$$(\bar{l_1} \vee \bar{l_2} \vee y)$$

$$\Downarrow$$
$$\varphi' = \varphi(\gamma / l)$$

$$\Downarrow$$
$$(\bar{l_1} \vee y) \wedge (\bar{l_1} \vee \bar{l_2} \vee y)$$

Equivalence Substitution

Satisfiability-preserving transformation

$$\bar{l} \leftarrow \bar{l_1} \wedge \bar{l_2}$$
$$l \leftarrow l_1 \vee l_2$$
$$\overbrace{l \equiv l_1 \vee l_2}$$

$$\varphi \qquad (\bar{l} \vee \bar{l_1} \vee y) \wedge (l \vee \bar{l_2} \vee \bar{y}) \wedge \overbrace{(\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \bar{l_1}) \wedge (l \vee \bar{l_2})}$$

$$\underbrace{(\bar{l} \vee \bar{l_1} \vee y)}_{(\bar{l_1} \vee \bar{l_1} \vee y)\ (\bar{l_1} \vee \bar{l_2} \vee y)} \qquad \underbrace{(l \vee \bar{l_2} \vee \bar{y})}_{(l_1 \vee l_2 \vee \bar{l_2} \vee \bar{y})}$$

$$\Downarrow \qquad \Downarrow$$

$$\varphi' = \varphi(\gamma/l) \qquad (\bar{l_1} \vee y) \wedge (\bar{l_1} \vee \bar{l_2} \vee y)$$

Dependency Graph

Given the QBF:

$$\varphi = \phi \wedge (l_1 \equiv l_2 \vee l_4) \wedge (l_2 \equiv l_3 \vee l_4) \wedge (l_3 \equiv l_1 \vee l_5)$$

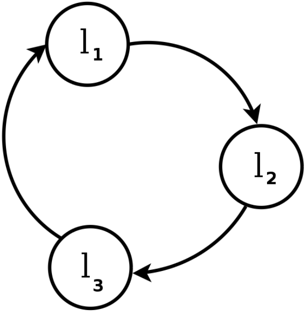
if l_1 is eliminated first, then when l_3 is substituted in the formula, l_1 is reintroduced

Dependency Graph

Given the QBF:

$$\varphi = \phi \wedge (l_1 \equiv l_2 \vee l_4) \wedge (l_2 \equiv l_3 \vee l_4) \wedge (l_3 \equiv l_1 \vee l_5)$$

if l_1 is eliminated first, then when l_3 is substituted in the formula, l_1 is reintroduced

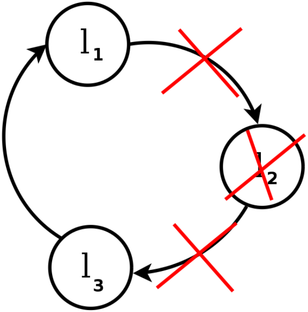


Dependency Graph

Given the QBF:

$$\varphi = \phi \wedge (l_1 \equiv l_2 \vee l_4) \wedge (l_2 \equiv l_3 \vee l_4) \wedge (l_3 \equiv l_1 \vee l_5)$$

if l_1 is eliminated first, then when l_3 is substituted in the formula, l_1 is reintroduced



Variable Elimination by Equivalence Checking Example

$$\overline{x} \leftarrow \overline{l_1} \vee \overline{l_2}$$
$$x \leftarrow l_1 \wedge l_2$$
$$x \equiv l_1 \wedge l_2$$

1: $(x \vee \overline{l_1} \vee y) \wedge (\overline{x} \vee \overline{l_2} \vee \overline{y}) \wedge (\overline{x} \vee l_1 \vee l_2) \wedge (x \vee \overline{l_1}) \wedge (x \vee \overline{l_1})$

$(l_1 \vee \overline{l_1} \vee y)$ $\quad (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$
$(l_2 \vee \overline{l_1} \vee y)$

$\Downarrow$

2: $(l_2 \vee \overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$

Variable Elimination by Equivalence Checking Example

$\overline{x} \leftarrow \overline{l_1} \vee \overline{l_2}$
$x \leftarrow l_1 \wedge l_2$

$x \equiv l_1 \wedge l_2$

1: $(x \vee \overline{l_1} \vee y) \wedge (\overline{x} \vee \overline{l_2} \vee \overline{y}) \wedge (\overline{x} \vee l_1 \vee l_2) \wedge (x \vee \overline{l_1}) \wedge (x \vee \overline{l_1})$

$(l_1 \vee \overline{l_1} \vee y)$ $(\overline{l_1} \vee \overline{l_2} \vee \overline{y})$
$(l_2 \vee \overline{l_1} \vee y)$

$\Downarrow$

2: $(l_2 \vee \overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$

Variable Elimination by Equivalence Checking Example

$$\overline{x} \leftarrow \overline{l_1} \vee \overline{l_2}$$
$$x \leftarrow l_1 \wedge l_2$$

$$x \equiv l_1 \wedge l_2$$

$$1: (x \vee \overline{l_1} \vee y) \wedge (\overline{x} \vee \overline{l_2} \vee \overline{y}) \wedge (\overline{x} \vee l_1 \vee l_2) \wedge (x \vee \overline{l_1}) \wedge (x \vee \overline{l_1})$$

$$(l_1 \vee \overline{l_1} \vee y) \quad (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$$
$$(l_2 \vee \overline{l_1} \vee y)$$

$$\Downarrow$$

$$2: (l_2 \vee \overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$$

Variable Elimination by Equivalence Checking
Example

$$\overline{x} \leftarrow \overline{l_1} \vee \overline{l_2}$$
$$x \leftarrow l_1 \wedge l_2$$

$$x \equiv l_1 \wedge l_2$$

$$1: (x \vee \overline{l_1} \vee y) \wedge (\overline{x} \vee \overline{l_2} \vee \overline{y}) \wedge (\overline{x} \vee l_1 \vee l_2) \wedge (x \vee \overline{l_1}) \wedge (x \vee \overline{l_1})$$

$$(l_1 \vee \overline{l_1} \vee y) \quad (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$$
$$(l_2 \vee \overline{l_1} \vee y)$$

$$\Downarrow$$

$$2: (l_2 \vee \overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$$

Variable Elimination by Equivalence Checking
Example

$$\overline{x} \leftarrow \overline{l_1} \vee \overline{l_2}$$
$$x \leftarrow l_1 \wedge l_2$$
$$x \equiv l_1 \wedge l_2$$

1: $(x \vee \overline{l_1} \vee y) \wedge (\overline{x} \vee \overline{l_2} \vee \overline{y}) \wedge (\overline{x} \vee l_1 \vee l_2) \wedge (x \vee \overline{l_1}) \wedge (x \vee \overline{l_1})$

$(l_1 \vee \overline{l_1} \vee y)$ $(\overline{l_1} \vee \overline{l_2} \vee \overline{y})$
$(l_2 \vee \overline{l_1} \vee y)$

$\Downarrow$

2: $(l_2 \vee \overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$

Variable Elimination by Equivalence Checking
Example

$$\overline{x} \leftarrow \overline{l_1} \vee \overline{l_2}$$
$$x \leftarrow l_1 \wedge l_2$$

$$x \equiv l_1 \wedge l_2$$

1: $(x \vee \overline{l_1} \vee y) \wedge (\overline{x} \vee \overline{l_2} \vee \overline{y}) \wedge (\overline{x} \vee l_1 \vee l_2) \wedge (x \vee \overline{l_1}) \wedge (x \vee \overline{l_1})$

$(l_1 \vee \overline{l_1} \vee y)$ $\quad (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$

$(l_2 \vee \overline{l_1} \vee y)$

$$\Downarrow$$

2: $(l_2 \vee \overline{l_1} \vee y) \wedge (\overline{l_1} \vee \overline{l_2} \vee \overline{y})$

Equivalence Rewriting (cont'd)

$$l \rightarrow l_1 \wedge l_2$$
$$\overline{l'} \rightarrow \overline{l_1} \vee \overline{l_2}$$
$$l \vee \overline{l'}$$
$$l \equiv l_1 \wedge l_2$$

$$(l \vee y \vee z) \wedge (\bar{l} \vee y \vee \bar{z}) \wedge (\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2})$$

$$(l' \vee y \vee z)$$

$$\Downarrow$$

$$(l' \vee y \vee z) \wedge (\bar{l} \vee y \vee \bar{z}) \wedge (\overline{l'} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2}) \wedge (l \vee \overline{l'})$$

Equivalence Rewriting (cont'd)

$$l \rightarrow l_1 \wedge l_2$$
$$\overline{l'} \rightarrow \overline{l_1} \vee \overline{l_2}$$
$$l \vee \overline{l'}$$
$$l \equiv l_1 \wedge l_2$$

$$(l \vee y \vee z) \wedge (\bar{l} \vee y \vee \bar{z}) \wedge \overbrace{(\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2})}$$

$$\underbrace{}_{(l' \vee y \vee z)}$$

$$\Downarrow$$

$$(l' \vee y \vee z) \wedge (\bar{l} \vee y \vee \bar{z}) \wedge (\overline{l'} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2}) \wedge (l \vee \overline{l'})$$

Equivalence Rewriting (cont'd)

$$l \rightarrow l_1 \wedge l_2$$
$$\overline{l'} \rightarrow \overline{l_1} \vee \overline{l_2}$$
$$l \vee \overline{l'}$$
$$l \equiv l_1 \wedge l_2$$

$$(l \vee y \vee z) \wedge (\bar{l} \vee y \vee \bar{z}) \wedge \overbrace{(\bar{l} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2})}$$

$$\underbrace{\quad}_{(l' \vee y \vee z)}$$

$$\Downarrow$$

$$(l' \vee y \vee z) \wedge (\bar{l} \vee y \vee \bar{z}) \wedge (\overline{l'} \vee l_1 \vee l_2) \wedge (l \vee \overline{l_1}) \wedge (l \vee \overline{l_2}) \wedge (l \vee \overline{l'})$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi$$

$$\alpha = \top$$

$$\varphi' = (l' \vee \beta) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \text{ (l is pure)}$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = (l \vee \alpha) \wedge (l \equiv \gamma) \wedge \phi$$

$\beta = \top$

$$\varphi' = (l \vee \alpha) \wedge (l \rightarrow \gamma) \wedge \phi \text{ } (\overline{l'} \text{ is pure})$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = (l \equiv \gamma) \wedge \phi$$

$\alpha = \top$ and $\beta = \top$

$$\varphi' = \phi \ (l \text{ and } \overline{l'} \text{ are pure})$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \lor \alpha) \land (\bar{l} \lor \beta) \land (l \equiv \gamma) \land \phi \text{ *original*}$$

$$\Downarrow$$

$$\varphi = \beta \land \gamma \land \phi$$

$\alpha = \bot$ (*l* is unit)

$$\varphi' = (l' \lor \beta) \land \gamma \land (\gamma \rightarrow \overline{l'}) \land \phi$$

$$\Uparrow$$

$$\varphi' = (l \lor \alpha) \land (l' \lor \beta) \land (l \rightarrow \gamma) \land (\gamma \rightarrow \overline{l'}) \land \phi \text{ *rewritten*}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \text{ original}$$

$$\Downarrow$$

$$\varphi = \beta \wedge \gamma \wedge \phi$$

$\alpha = \bot$ (*l* is unit)

$$\varphi' = (l' \vee \beta) \wedge \gamma \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \text{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = \beta \wedge \gamma \wedge \phi$$

$\alpha = \bot$ (*l* is unit)

$$\varphi' = (l' \vee \beta) \wedge \gamma \wedge (\gamma \rightarrow \overline{l'}) \wedge (\bar{l} \vee \overline{l'}) \wedge \phi$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = \alpha \wedge (\neg\gamma) \wedge \phi$$

$$\beta = \bot$$

$$\varphi' = (l \vee \alpha) \wedge (l \rightarrow \gamma) \wedge (\neg\gamma) \wedge \phi \text{ (l' is unit)}$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = \alpha \wedge (\neg\gamma) \wedge \phi$$

$\beta = \bot$

$$\varphi' = (l \vee \alpha) \wedge (l \rightarrow \gamma) \wedge (\neg\gamma) \wedge \phi \text{ (l' is unit)}$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Influence of Equivalence Rewriting

$$\varphi = (l \vee \alpha) \wedge (\bar{l} \vee \beta) \wedge (l \equiv \gamma) \wedge \phi \textit{ original}$$

$$\Downarrow$$

$$\varphi = \alpha \wedge (\neg\gamma) \wedge \phi$$

$\beta = \perp$

$$\varphi' = (l \vee \alpha) \wedge (l \rightarrow \gamma) \wedge (\neg\gamma) \wedge (\bar{l} \vee \overline{l'}) \wedge \phi \ (l' \text{ is unit})$$

$$\Uparrow$$

$$\varphi' = (l \vee \alpha) \wedge (l' \vee \beta) \wedge (l \rightarrow \gamma) \wedge (\gamma \rightarrow \overline{l'}) \wedge \phi \textit{ rewritten}$$

Example:

Given

$$\exists x_1 \forall y \exists x_2\, x_3 (\bar{x}_1 \vee y \vee \bar{x}_2) \wedge (x_1 \vee y \vee x_3) \wedge (y \vee \bar{x}_3) \wedge (x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$$

let's remove x_3 by Q-Resolution:

$$\begin{array}{ll} (x_1 \vee y \vee x_3) \otimes (y \vee \bar{x}_3) & \Rightarrow x_1 \\ (x_1 \vee y \vee x_3) \otimes (\bar{x}_2 \vee \bar{x}_3) & \Rightarrow x_1 \vee y \vee \bar{x}_2 \\ (x_2 \vee x_3) \otimes (y \vee \bar{x}_3) & \Rightarrow y \vee x_2 \\ (x_2 \vee x_3) \otimes (\bar{x}_2 \vee \bar{x}_3) & \Rightarrow x_2 \vee \bar{x}_2 \Rightarrow \top \end{array}$$

The given QBF becomes:

$$\exists x_1 \forall y \exists x_2 (\bar{x}_1 \vee y \vee \bar{x}_2) \wedge (x_1) \wedge (x_1 \vee y \vee \bar{x}_2) \wedge (y \vee x_2)$$

Variable Elimination by Q-Resolution Example:

Given

$$\exists x_1 \forall y \exists x_2\, x_3 (\bar{x}_1 \vee y \vee \bar{x}_2) \wedge (x_1 \vee y \vee x_3) \wedge (y \vee \bar{x}_3) \wedge (x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$$

let's remove x_3 by Q-Resolution:

$$\begin{array}{ll} (x_1 \vee y \vee x_3) \otimes (y \vee \bar{x}_3) & \Rightarrow x_1 \\ (x_1 \vee y \vee x_3) \otimes (\bar{x}_2 \vee \bar{x}_3) & \Rightarrow x_1 \vee y \vee \bar{x}_2 \\ (x_2 \vee x_3) \otimes (y \vee \bar{x}_3) & \Rightarrow y \vee x_2 \\ (x_2 \vee x_3) \otimes (\bar{x}_2 \vee \bar{x}_3) & \Rightarrow x_2 \vee \bar{x}_2 \Rightarrow \top \end{array}$$

The given QBF becomes:

$$\exists x_1 \forall y \exists x_2 (\bar{x}_1 \vee y \vee \bar{x}_2) \wedge (x_1) \wedge (x_1 \vee y \vee \bar{x}_2) \wedge (y \vee x_2)$$

Backjumping

Problem

Time spent visiting parts of the search space in vain because some choices may not be responsible for the result of the search

Solution [Giunchiglia, Narizzano, Tacchella 2001]

1. for each node of the search tree, compute a subset (called "reason") of the assigned variables which are responsible for the current result; and
2. while backtracking, skip nodes which do not belong to the reason for the discovered conflicts/solutions:

   CBJ Conflict Backjumping
   SBJ Solution Backjumping

Backjumping

Problem

Time spent visiting parts of the search space in vain because some choices may not be responsible for the result of the search

Solution [Giunchiglia, Narizzano, Tacchella 2001]

1. for each node of the search tree, compute a subset (called "reason") of the assigned variables which are responsible for the current result; and
2. while backtracking, skip nodes which do not belong to the reason for the discovered conflicts/solutions:

   CBJ Conflict Backjumping
   SBJ Solution Backjumping

Example

Original Formula:

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_4 x_5$$
$$(x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_2} \vee x_4) \wedge (\overline{x_3} \vee x_4) \wedge (x_1 \vee y_1 \vee x_3 \vee x_5) \wedge$$
$$(\overline{x_3} \vee x_4 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_4) \wedge (x_1 \vee y_2 \vee \overline{x_4}) \wedge$$
$$(\overline{y_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

$(\overline{x_3} \vee x_4 \vee x_5)$ is subsumed;

$x_4 \equiv x_3 \vee x_2$: remove x_4 by Equivalence Checking:

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_5$$
$$(x_1 \vee y_1 \vee x_3 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_3}) \wedge (x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Example

Original Formula:

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_4 x_5$$
$$(x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_2} \vee x_4) \wedge (\overline{x_3} \vee x_4) \wedge (x_1 \vee y_1 \vee x_3 \vee x_5) \wedge$$
$$(\overline{x_3} \vee x_4 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_4) \wedge (x_1 \vee y_2 \vee \overline{x_4}) \wedge$$
$$(\overline{y_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

$(\overline{x_3} \vee x_4 \vee x_5)$ is subsumed;

$x_4 \equiv x_3 \vee x_2$: remove x_4 by Equivalence Checking:

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_5$$
$$(x_1 \vee y_1 \vee x_3 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_3}) \wedge (x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Example

Original Formula:

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_4 x_5$$
$$(x_2 \lor x_3 \lor \overline{x_4}) \land (\overline{x_2} \lor x_4) \land (\overline{x_3} \lor x_4) \land (x_1 \lor y_1 \lor x_3 \lor x_5) \land$$
$$(\overline{x_3} \lor x_4 \lor x_5) \land (x_1 \lor y_1 \lor x_2 \lor \overline{x_5}) \land (x_1 \lor \overline{y_2} \lor x_4) \land (x_1 \lor y_2 \lor \overline{x_4}) \land$$
$$(\overline{y_1} \lor \overline{x_3}) \land (\overline{x_1} \lor x_2 \lor \overline{x_3})$$

$(\overline{x_3} \lor x_4 \lor x_5)$ is subsumed;

$x_4 \equiv x_3 \lor x_2$: remove x_4 by Equivalence Checking:

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_5$$
$$(x_1 \lor y_1 \lor x_3 \lor x_5) \land (x_1 \lor y_1 \lor x_2 \lor \overline{x_5}) \land (x_1 \lor \overline{y_2} \lor x_2 \lor x_3) \land$$
$$(x_1 \lor y_2 \lor \overline{x_3}) \land (x_1 \lor y_2 \lor \overline{x_2}) \land (\overline{x_1} \lor x_2 \lor \overline{x_3})$$

Example (cont.)

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_5$$
$$(x_1 \vee y_1 \vee x_3 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_3}) \wedge (x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Try to remove x_2 by Q-Resolution:

$$(x_1 \vee \overline{y_2} \vee x_2 \vee x_3\} \otimes_{x_2} \{x_1 \vee y_2 \vee \overline{x_2}\} = \{x_1 \vee y_2 \vee \overline{y_2} \vee x_3)$$

y_2 locks (... same x_3 ...)

Try to remove x_5 by Q-Resolution:

$$(x_1 \vee y_1 \vee x_3 \vee x_5) \otimes_{x_5} (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) = (x_1 \vee y_1 \vee x_2 \vee x_3)$$

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3$$
$$(x_1 \vee y_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Example (cont.)

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_5$$
$$(x_1 \vee y_1 \vee x_3 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_3}) \wedge (x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Try to remove x_2 by Q-Resolution:

$$(x_1 \vee \overline{y_2} \vee x_2 \vee x_3\} \otimes_{x_2} \{x_1 \vee y_2 \vee \overline{x_2}\} = \{x_1 \vee y_2 \vee \overline{y_2} \vee x_3)$$

y_2 locks (... same x_3 ...)

Try to remove x_5 by Q-Resolution:

$$(x_1 \vee y_1 \vee x_3 \vee x_5) \otimes_{x_5} (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) = (x_1 \vee y_1 \vee x_2 \vee x_3)$$

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3$$
$$(x_1 \vee y_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Example (cont.)

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3 x_5$$
$$(x_1 \vee y_1 \vee x_3 \vee x_5) \wedge (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_3}) \wedge (x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Try to remove x_2 by Q-Resolution:

$$(x_1 \vee \overline{y_2} \vee x_2 \vee x_3\} \otimes_{x_2} \{x_1 \vee y_2 \vee \overline{x_2}\} = \{x_1 \vee y_2 \vee \overline{y_2} \vee x_3)$$

y_2 locks (... same x_3 ...)

Try to remove x_5 by Q-Resolution:

$$(x_1 \vee y_1 \vee x_3 \vee x_5) \otimes_{x_5} (x_1 \vee y_1 \vee x_2 \vee \overline{x_5}) = (x_1 \vee y_1 \vee x_2 \vee x_3)$$

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3$$
$$(x_1 \vee y_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Example (cont.)

$\exists x_1 \forall y_1 y_2 \exists x_2 x_3$
$(x_1 \vee y_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$
$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$

y_1 is pure:

$\exists x_1 \forall y_2 \exists x_2 x_3$
$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$
$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$

Example (cont.)

$$\exists x_1 \forall y_1 y_2 \exists x_2 x_3$$
$$(x_1 \vee y_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

y_1 is pure:

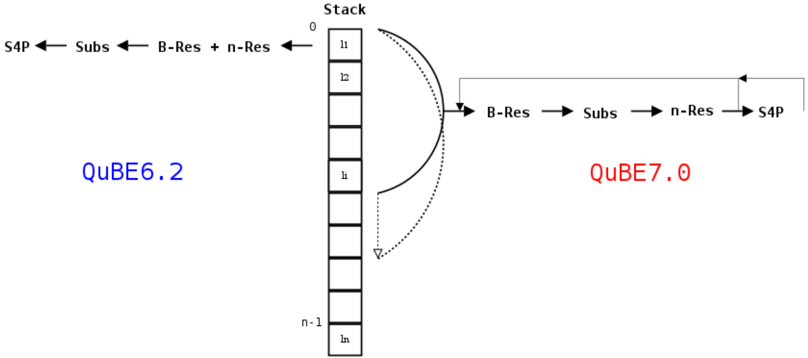
$$\exists x_1 \forall y_2 \exists x_2 x_3$$
$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{y_2} \vee x_2 \vee x_3) \wedge (x_1 \vee y_2 \vee \overline{x_3}) \wedge$$
$$(x_1 \vee y_2 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Main Search Loop

$\forall l \in$ AssignmentStack we apply 4 different operations:

1. B–Res: Resolution (a.k.a. *Unit Propagation*) on Binary Constraints
2. Subs: Subsumption of Binary and *n*-ary Constraints
3. *n*-Res: Resolution on *n*-ary Constraints
4. S4P: Search for Pure Literals

Comparison of QuBE6.x a and QuBE7 main loops



Results on QBFEval'10 testset

Solver	Total		Sat		Unsat	
	#	Time	#	Time	#	Time
aqme-10	434	32091.1	184	15825.6	250	16265.5
QuBE7	**410**	52142.1	189	35489.7	221	1402.62
QuBE7.0	**403**	44204.3	180	26342.4	223	17861.9
depqbf	370	21515.3	164	13771.8	206	7743.5
qmaiga	361	43058.1	180	20696.6	181	22361.4
depqbf-pre	356	18995.9	172	12453.8	184	6542.1
AIGSolve	329	22786.6	171	12091.5	158	10695.1
struqs-10	240	32839.7	109	13805.5	131	19034.2
nenofex-qbfeval10	225	13786.9	109	8241.9	116	5545.1
quantor-3.1	205	6711.4	100	4130.6	105	2580.7