SYMMETRIC BOOLEAN QUANTIFIERS

Quantified Boolean formulae (QBF)

A QBF has the form $Q_1x_1...Q_nx_n:\phi(x_1,x_2,...,x_n)$ where $Q_1,...,Q_n$ are Boolean quantifiers $\exists$ or $\forall$, and $\phi(x_1,x_2,...,x_n)$ is a Boolean (or propositional) logic formula in the variables $x_1,x_2,...,x_n$. (The variables $x_1,x_2,...,x_n$ may only take the values T, F.)

The Boolean quantifiers $\exists$, $\forall$ have the following meanings.

$\exists x_1$: $\phi(x_1,x_2,...,x_n)$ means $\phi(T,x_2,...,x_n) \vee \phi(F,x_2,...,x_n)$.

$\forall x_1$: $\phi(x_1,x_2,...,x_n)$ means $\phi(T,x_2,...,x_n) \wedge \phi(F,x_2,...,x_n)$.

Example of a QBF: $\exists x_1\forall x_2$: $(x_1\vee{\sim}x_2)$. It is easily seen that this formula is true.

Boolean formulae quantified by symmetric Boolean quantifiers (SQBF)

An SQBF has the form $Q_1x_1...Q_nx_n:\phi(x_1,x_2,...,x_n)$ where $Q_1,...,Q_n$ are symmetric Boolean quantifers $\exists!$ or $\forall!$, and $\phi(x_1,x_2,...,x_n)$ is a Boolean logic formula in the variables $x_1,x_2,...,x_n$.

The symmetric Boolean quantifiers $\exists!$, $\forall!$ have the following meanings.

$\exists!x_1$: $\phi(x_1,x_2,...,x_n)$ means $\phi(T,x_2,...,x_n) \oplus \phi(F,x_2,...,x_n)$ where $\oplus$ denotes "exclusive or".

$\forall!x_1$: $\phi(x_1,x_2,...,x_n)$ means $\phi(T,x_2,...,x_n) \Leftrightarrow \phi(F,x_2,...,x_n)$ where $\Leftrightarrow$ denotes "if and only if".

Example of an SQBF: $\exists!x_1\forall!x_2$: $(x_1\vee{\sim}x_2)$. This formula is also true but it is harder to see why.

Comment

Boolean quantifiers are based on the connectives $\vee$, $\wedge$ whereas symmetric Boolean quantifiers are based on the connectives $\oplus$, $\Leftrightarrow$.

Complexity

Determining the truth value of a quantified Boolean formulae (QBF) is known to be a complete problem for P-space. What can be said about SQBF?

We shall show that any SQBF $Q_1x_1...Q_nx_n:\phi(x_1,x_2,...,x_n)$ is equivalent $Q_1x_1...Q_1x_n:\phi(x_1,x_2,...,x_n)$ i.e. all symmetric Boolean quantifiers can be made equal to Q_1, the first one. From this it will follow that the complexity of determining the truth value of an SQBF is the same as the complexity of determining the truth value of its negation.

Properties of $\oplus$, $\Leftrightarrow$, $\exists!$, $\forall!$

1) $\oplus$, $\Leftrightarrow$ are commutative and associative, are true in two cases false in two cases, and are negations of each other.

2) $\oplus$, $\Leftrightarrow$ are the only Boolean connectives (x f y) which are true in two cases and false in two cases and different from x, y, ~x, ~y.

3) $x \oplus T \equiv T \oplus x \equiv {\sim}x$. $x \oplus F \equiv F \oplus x \equiv x$. $x \oplus x \equiv F$.

4) $x_1 \Leftrightarrow x_2 \Leftrightarrow x_3 \equiv x_1 \oplus x_2 \oplus x_3$. To see this, rewrite each $\Leftrightarrow$ as $(T \oplus (...\oplus...))$, rearrange and simplify. However $x_1 \Leftrightarrow x_2 \equiv {\sim}(x_1 \oplus x_2)$.

More generally $x_1 \Leftrightarrow x_2 ... \Leftrightarrow x_n \equiv x_1 \oplus x_2 ... \oplus x_n$ when n is odd.

However, $x_1 \Leftrightarrow x_2 ... \Leftrightarrow x_n \equiv {\sim}(x_1 \oplus x_2 ... \oplus x_n)$ when n is even.

5) $x_1 \oplus x_2 ... \oplus x_n$ is true if and only if an odd number of the variables x_i are true.

6) When n is even, $x_1 \Leftrightarrow x_2 ... \Leftrightarrow x_n$ is true if and only if an even number of the variables x_i are true. However when n is odd, $x_1 \Leftrightarrow x_2 ... \Leftrightarrow x_n$ is true if and only if an odd number of the variables x_i are true.

7) ${\sim}\exists!x_1$: $\phi(x_1,x_2,...,x_n) \equiv \forall!x_2$: $\phi(x_1,x_2,...,x_n)$.
${\sim}\forall!x_1$: $\phi(x_1,x_2,...,x_n) \equiv \exists!x_2$: $\phi(x_1,x_2,...,x_n)$.

This follows directly from the definitions of $\exists!$, $\forall!$ since $\oplus$, $\Leftrightarrow$ are negations of each other.

8) $\exists!x_1$: ${\sim}\phi(x_1,x_2,...,x_n) \equiv \exists!x_2$: $\phi(x_1,x_2,...,x_n)$.
$\forall!x_1$: ${\sim}\phi(x_1,x_2,...,x_n) \equiv \forall!x_2$: $\phi(x_1,x_2,...,x_n)$.

This follows from the definitions of $\exists!$, $\forall!$ by rewriting each ~ as $(T \oplus ...)$, each $\Leftrightarrow$ as $(T \oplus (...\oplus...))$, then rearrange and simplify.

Making the symmetric Boolean quantifiers the same as the first one

a) $\exists!x_1\forall!x_2$: $\phi(x_1,x_2,...,x_n) \equiv \exists!x_1\exists!x_2$: $\phi(x_1,x_2,...,x_n)$.

b) $\forall!x_1\exists!x_2$: $\phi(x_1,x_2,...,x_n) \equiv \forall!x_1\forall!x_2$: $\phi(x_1,x_2,...,x_n)$.

c) More generally any SQBF $Q_1x_1...Q_nx_n:\phi(x_1,x_2,...,x_n) \equiv Q_1x_1...Q_1x_n:\phi(x_1,x_2,...,x_n)$ i.e. all symmetric Boolean quantifiers can be made equal to Q_1, the first one.

Proofs:

a) $\exists!x_1\forall!x_2\text{: }\phi(x_1,x_2,...,x_n)$
$\equiv (\phi(T,T,...,x_n) \Leftrightarrow \phi(T,F,...,x_n)) \oplus (\phi(F,T,...,x_n) \Leftrightarrow \phi(F,F,...,x_n))$
$\equiv \sim(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \oplus \sim(\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv T+(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \oplus T+(\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv T+T+(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \oplus (\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv F+(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \oplus (\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv (\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \oplus (\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv \exists!x_1\exists!x_2\text{: }\phi(x_1,x_2,...,x_n)$

b) $\forall!x_1\exists!x_2\text{: }\phi(x_1,x_2,...,x_n)$
$\equiv (\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \Leftrightarrow (\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv F+(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \Leftrightarrow (\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv T+T+(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \Leftrightarrow (\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv T+(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \Leftrightarrow T+(\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv \sim(\phi(T,T,...,x_n) \oplus \phi(T,F,...,x_n)) \Leftrightarrow \sim(\phi(F,T,...,x_n) \oplus \phi(F,F,...,x_n))$
$\equiv (\phi(T,T,...,x_n) \Leftrightarrow \phi(T,F,...,x_n)) \Leftrightarrow (\phi(F,T,...,x_n) \Leftrightarrow \phi(F,F,...,x_n))$
$\equiv \forall!x_1\forall!x_2\text{: }\phi(x_1,x_2,...,x_n)$

c) Repeated use of (a) when Q_1 is $\exists!$. Repeated use of (b) when Q_1 is $\forall!$.

The complexity of determining the truth value of an SQBF is the same as the complexity of determining the truth value of its negation i.e. SQBF ≡ co-SQBF

In view of the results of the previous section, we may assume that all the symmetrical Boolean quantifiers are the same. So we may assume that the SQBF we are dealing with has one of the forms

$\exists!x_1...\exists!x_n{:}\phi(x_1,x_2,...,x_n)$ or $\forall!x_1...\forall!x_n{:}\phi(x_1,x_2,...,x_n)$.

By repeated use of properties 7,8 it follows that:

$\sim\exists!x_1...\exists!x_n{:}\phi(x_1,x_2,...,x_n) \equiv \forall!x_1...\forall!x_n{:}\phi(x_1,x_2,...,x_n)$.
$\sim\forall!x_1...\forall!x_n{:}\phi(x_1,x_2,...,x_n) \equiv \exists!x_1...\exists!x_n{:}\phi(x_1,x_2,...,x_n)$.

However $\exists!x_1...\exists!x_n{:}\phi(x_1,x_2,...,x_n)$ is determined by "summing" using $\oplus$ all possible 2^n values of $\phi(x_1,x_2,...,x_n)$ in its truth table and so by property 5 will be true when $\phi(x_1,x_2,...,x_n)$ has the value T in an odd number of rows in its truth table. Similarly $\forall!x_1...\forall!x_n{:}\phi(x_1,x_2,...,x_n)$ is determined by "summing" using $\Leftrightarrow$ all possible 2^n values of $\phi(x_1,x_2,...,x_n)$ in its truth table and so by property 6 will be true when $\phi(x_1,x_2,...,x_n)$ has the value T in an even number of rows in its truth table, since 2^n is even.

Let us now define $\psi(x_1,x_2,...,x_n)$ to be like $\phi(x_1,x_2,...,x_n)$ except for one row in the true table. For example if we make them differ only in the row where every x_i is true, then $\psi(x_1,x_2,...,x_n)$ can be the formula $((x_1 \wedge x_2,... \wedge x_n) \Rightarrow \sim\phi(x_1,x_2,...,x_n)) \wedge (\sim(x_1 \wedge x_2,... \wedge x_n) \Rightarrow \phi(x_1,x_2,...,x_n))$.

Clearly $\phi(x_1,x_2,...,x_n)$ is true in an odd number of cases if and only if $\psi(x_1,x_2,...,x_n)$ is true in an even number of cases and vice versa. As odd, even are negations of each other we can write this as the following equivalences between SQBF.

$\sim\exists!x_1...\exists!x_n{:}\phi(x_1,x_2,...,x_n) \equiv \forall!x_1...\forall!x_n{:}\ \psi(x_1,x_2,...,x_n)$.
$\sim\forall!x_1...\forall!x_n{:}\phi(x_1,x_2,...,x_n) \equiv \exists!x_1...\exists!x_n{:}\ \psi(x_1,x_2,...,x_n)$.

Clearly this transformation of ϕ to ψ can be done in polynomial time and the size of ψ is linear in the size of ϕ. So SQBF ≡ co-SQBF.

(From the above proof we also note that $\exists!x_1...\exists!x_n{:}\phi(x_1,x_2,...,x_n)$ means that $\phi(x_1,x_2,...,x_n)$ is true in an odd number of cases and $\forall!x_1...\forall!x_n{:}\phi(x_1,x_2,...,x_n)$ means that $\phi(x_1,x_2,...,x_n)$) is true in an even number of cases.)

The complexity of determining the truth value of an SQBF is in LIN-space

As before we may assume that all the symmetrical Boolean quantifiers can be made the same as the first one. The truth value of an SQBF $Q_1x_1...Q_nx_n{:}\phi(x_1,x_2,...,x_n)$ can now be determined as follows.

```
IF $Q_1$ is $\exists!$
THEN
    result := F
    FOR $x_1$:=T,F, $x_2$:=T,F, ...,$x_n$:=T,F
    LOOP result := result $\oplus$ $\phi(x_1,x_2,...,x_n)$; ENDLOOP;
ENDIF;
```

```
IF $Q_1$ is $\forall!$
THEN
    result := T
    FOR $x_1$:=T,F, $x_2$:=T,F, ...,$x_n$:=T,F
    LOOP result := result $\Leftrightarrow \phi(x_1,x_2,...,x_n)$; ENDLOOP;
ENDIF;
```

In view of this and the previous result, the complexity of solving this problem seems to be in P or in (NP $\cap$ co-NP) or in ($\Sigma_i \cap \Pi_i$) for some i (which may or may not depend on the SQBF).

The complexity of determining the truth value of a restricted SQBF is in P

When the only connectives allowed in $\phi(x_1,x_2,...,x_n)$ are $\oplus$, $\Leftrightarrow$, determining the truth value of an SQBF $Q_1x_1...Q_nx_n:\phi(x_1,x_2,...,x_n)$ is in P. (This implicitly includes ~ since $\sim x \equiv (x \Leftrightarrow x) \oplus x$.)
Let us rewrite each $\Leftrightarrow$ as (T $\oplus$ (...$\oplus$...)), then rearrange and simplify in the obvious way to obtain a formula in which the only connective is $\oplus$. If the same variable or value occurs twice in the formula then eliminate both occurrences since $x \oplus x \equiv$ F. In this way we obtain an equivalent simplified formula $\psi(x_1,x_2,...,x_n)$ which is either T or F or a "sum" of some or all of the variables $x_1,x_2,...,x_n$ using $\oplus$, or T $\oplus$ a "sum" of some or all of the variables $x_1,x_2,...,x_n$ using $\oplus$. Note that in all these cases, $\psi(x_1,x_2,...,x_n)$ is true in an even number of cases except when n=1 and $\psi(x_1)$ is x_1 or $\psi(x_1)$ is T $\oplus$ x_1 in which case $\psi(x_1)$ is true in an odd number of cases.
Clearly this simplification can be done in polynomial time.
As before we may assume that all the all the symmetrical Boolean quantifiers are the same.
So the SQBF we are dealing with is equivalent to $\exists!x_1...\exists!x_n:\psi(x_1,x_2,...,x_n)$ or $\forall!x_1...\forall!x_n:\psi(x_1,x_2,...,x_n)$. (Note that the equivalent simplified formula $\psi(x_1,x_2,...,x_n)$ is used here.)
Since $\psi(x_1,x_2,...,x_n)$ is true in an even number of cases, the SQBF $\exists!x_1...\exists!x_n:\psi(x_1,x_2,...,x_n)$ is false and the SQBF is $\forall!x_1...\forall!x_n:\psi(x_1,x_2,...,x_n)$ is true except when n=1 and $\psi(x_1)$ is x_1 or $\psi(x_1)$ is T $\oplus$ x_1 in which case $\exists!x_1:\psi(x_1)$ is true and $\exists!x_1:\psi(x_1)$ is false.
Clearly even with these additional steps, the truth value of such a restricted SQBF can be determined in polynomial time.

Questions

1) SQBF is in LIN-space which is a subset of P-space and is therefore reducible in polynomial time to QBF. (QBF is P-space complete.) What form does this reduction take? How can an SQBF be expressed as a QBF?
2) Is SQBF complete for P-space?
3) Is SQBF solvable in polynomial time?
4) Is NP reducible in polynomial time to SQBF?
   Is SQBF reducible in polynomial time to NP?
5) Is co-NP reducible in polynomial time to SQBF?
   Is SQBF reducible in polynomial time to co-NP?
6) Is SQBF a complete problem for NP and for co-NP?
7) What is the complexity of determining if the function defined by a Boolean formula is symmetric? This problem seems to be in LIN-space but can we say more?
   A symmetric Boolean function is unaltered when the values of any two variables are interchanged. It is not clear if this question is related to the topic of this article.
   Furthermore, what can be said of the complexity of a QBF and of an SQBF whose Boolean formula is symmetric?

APPENDIX

Clausal form based on exclusive or $\oplus$ (CXNF)

Clausal form, also known as conjunctive normal form (CNF) has the form of a conjunction of disjunctions of literals, where a literal is a Boolean variable or its negation, and in the disjunctions inclusive or $\vee$ is used.
CXNF is like clausal form or CNF, but the disjunctions use exclusive or $\oplus$.
Since $x \vee y \equiv x \oplus y \oplus (x \wedge y)$, it follows that ~, $\oplus$, $\wedge$, is a complete set of connectives.
However, not every Boolean formula can be converted to CXNF, e.g. $x \vee y$.

An incomplete resolution principle based on ⊕

The following resolution principle holds for CXNF.
Let A_i, B_j be literals and let R be a Boolean varialble.
Given that the following two CXNF clauses are true

$A_1 \oplus A_2 \oplus \dots \oplus A_m \oplus R$

$B_1 \oplus B_2 \oplus \dots \oplus B_n \oplus \sim R$

then the following CXNF clause is also true.

$A_1 \oplus A_2 \oplus \dots \oplus A_m \oplus B_1 \oplus B_2 \oplus \dots \oplus B_n$.

This follows from property 5 when considering the two cases R is T and R is F.
If R is T then an even number of A_i's and an odd number of B_j's are true and so an odd number of A_i's and B_j's are true.
If R is F then an odd number of A_i's and an even number of B_j's are true and so an odd number of A_i's and B_j's are true.
So irrespective of the truth value of R, an odd number of A_i's and B_j's are true, i.e. $A_1 \oplus A_2 \oplus \dots \oplus A_m \oplus B_1 \oplus B_2 \oplus \dots \oplus B_n$ is true.
However, this resolution principle is not complete, since the following two CXNF clauses are contradictory but the empty clause is not derivable from them by this resolution principle.
(i) $x \oplus y$ (ii) $x \oplus \sim y$
(Can inference rules be added to make such a deduction system complete?)

Satisfiability of CXNF can be determined in polynomial time using Gaussian elimination

A formula in CXNF is a conjunction of disjunctions of literals, with exclusive or ⊕ in the disjunctions. Let us rewrite each literal of the form ~x by ($T \oplus x$), and simplify as before. The disjunctions now have the form of a sum using ⊕ of Boolean variables, or T ⊕ a sum using ⊕ of Boolean variables, which is just a negation of a sum using ⊕ of Boolean variables.
So these disjunctions can be viewed as simultaneous linear equations having the following two forms: (i) sum using ⊕ of Boolean variables = T.
(ii) sum using ⊕ of Boolean variables = F.
By replacing T by 1, F by 0, ⊕ by "+ modulo 2" we obtain a set of linear equations to solve over the finite field of arithmetic modulo 2, as ⊕ and "+ modulo 2" are isomorphic based on these replacements. We can determine if a solution exists and find values of the Boolean variables satisfying these equations by Gaussian elimination.
Clearly all the above can be done in polynomial time.
(It seems to us that for such linear equations, Gaussian elimination can be generalized for use on any group for which $x+x=0$; there is no need for a ring or field. Variables have no coefficients in these equations; they either appear or are omitted.)

Disjunctive normal form based on exclusive or ⊕ (DXNF)

Similarly, a formula in DXNF is a disjunction of conjunctions of literals, with exclusive or ⊕ in the disjunction. Further work is needed to investigate matters of complexity of DXNF.

Other combinations of quantifiers

Perhaps more results can be found for other combinations of quantifiers such as ∃! with ∀ or ∃ with ∀!.

R.B. Yehezkael (formerly Haskell)
Revised April 2005 - ניסן תשס"ה (Minor changes made November 2008 – חשון תשס"ט)
Jerusalem College of Technology - Machon Lev,
Hawaad Haleumi 21, Jerusalem 91160, ISRAEL.
Tel: 02-6751111
e-mail : rafi@mail.jct.ac.il home page: http://cc.jct.ac.il/~rafi