QBF function problems: preliminary results

Sylvie Coste-Marquis [*], ***Hélène Fargier*** [†], ***Jérôme Lang*** [†],

Daniel Le Berre [*] ***and Pierre Marquis*** [*]

CRIL, Université d'Artois, Lens [*] **and IRIT, UPS, Toulouse** [†]

Friday, November 21th, 2003

Layout of the talk

- Motivations
- Total policies and the first function problem FQBF
- Partial policies and the second function problem SFQBF
- Policy representation
- Two approaches for a case study: $\text{SFQBF}_{2,\forall}$
- Conclusion and perspectives

Motivations

- Existing work on QBF focus on
  - Solving QBFs (Cadoli et al., 1998; Rintanen 1999, 2001, Giunchiglia et al., 2001; Letz, 2001; Zhang and Malik, 2002)
  - Taking advantage of QBF solvers in AI (Egly et al., 2000, 2002; Rintanen, 1999)
- But some problems require more than a **boolean output**!

Example: sequential games against nature

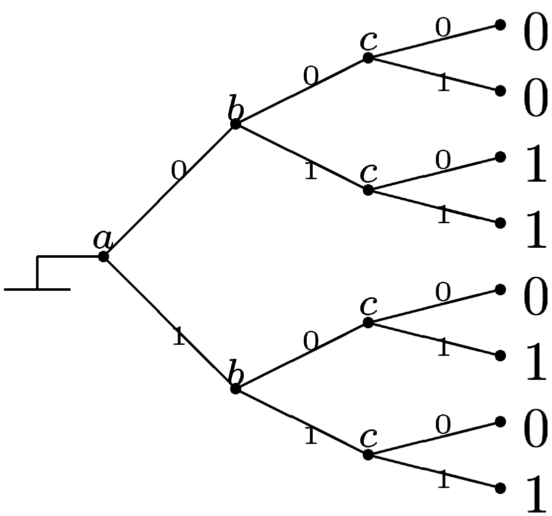
- The game is given by a QBF
- Two players: the $\exists$ one, and the $\forall$ one (nature)
- Players play alternatively by assigning a truth value to a propositional symbol
- The $\exists$ player has a **winning strategy** for the game iff the QBF is **valid**

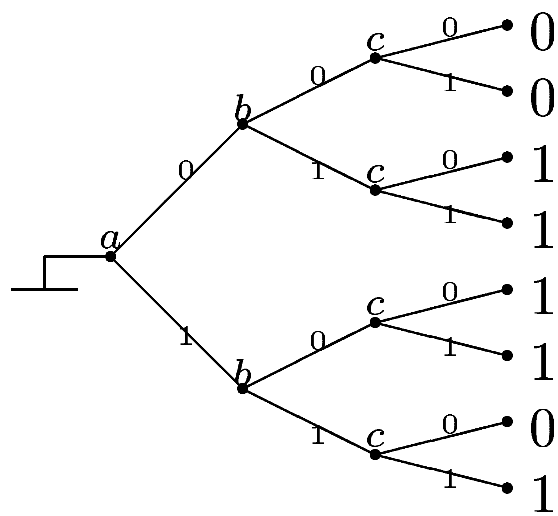
Example (cont'd)

- Two games
  - $P = \forall a\ \exists b\ \forall c\ (\neg a \vee c) \wedge (a \vee b)$
  - $P\prime = \forall a\ \exists b\ \forall c\ (\neg a \vee \neg b \vee c) \wedge (a \vee b)$
- There is no winning strategy for P, but a winning strategy for $P\prime$
- Knowing that a winning strategy exists is not sufficient: given a move of the $\forall$ player, the $\exists$ player must know **a move she has to play** in order to win
- Such an output is called a **(decision) policy**

Example (cont'd)

- $P = \forall a \; \exists b \; \forall c \; (\neg a \vee c) \wedge (a \vee b)$
- $P' = \forall a \; \exists b \; \forall c \; (\neg a \vee \neg b \vee c) \wedge (a \vee b)$





QBF

- Let k be a positive integer and $q \in \{\forall, \exists\}$. A **Quantified Boolean Formula (QBF)** (in prenex normal form) is a $(k+3)$-uple $P = \langle k, q, X_k, ..., X_1, \Phi \rangle$ where $\{X_1, ..., X_k\}$ is a partition of the set $Var(\Phi)$ of propositional symbols occurring in $\Phi \in PROP_{PS}$
- QBF$_{k,q}$ is the set of all QBFs of rank k and first quantifier q
- $P = \forall a\ \exists b\ \forall c\ (\neg a \vee c) \wedge (a \vee b)$ is represented as $\langle 3, \forall, \{a\}, \{b\}, \{c\}, (\neg a \vee c) \wedge (a \vee b) \rangle$ and belongs to QBF$_{3,\forall}$

QBF (cont'd)

- $P = \langle k, q, X_k, ..., X_1, \Phi\rangle$ is a **positive** instance of $\text{QBF}_{k,q}$ iff one of the following conditions is true:
  - $k = 0$ and $\Phi = \top$;
  - $k \geq 1$ and $q = \exists$ and there exists an X_k-instantiation $\vec{x_k} \in 2^{X_k}$ such that $\langle k-1, \forall, X_{k-1}, ..., X_1, \Phi_{\vec{x_k}}\rangle$ is a positive instance of $\text{QBF}_{k-1,\forall}$;
  - $k \geq 1$ and $q = \forall$ and for each X_k-instantiation $\vec{x_k} \in 2^{X_k}$, $\langle k-1, \exists, X_{k-1}, \ldots, X_1, \Phi_{\vec{x_k}}\rangle$ is a positive instance of $\text{QBF}_{k-1,\exists}$
- $P = \langle 3, \forall, \{a\}, \{b\}, \{c\}, (\neg a \vee c) \wedge (a \vee b)\rangle$ is not a positive instance of $\text{QBF}_{3,\forall}$

Total policies

- The set $TP(k, q, X_k, ..., X_1)$ of **total policies** for QBFs from $\text{QBF}_{k,q}$ is defined inductively by (λ means "success"):
  - $TP(0, q) = \{\lambda\}$;
  - $TP(k, \exists, X_k, ..., X_1) = \{\vec{x_k}\ ; \pi_{k-1} \mid \pi_{k-1} \in TP(k-1, \forall, X_{k-1}, ..., X_1)\}$;
  - $TP(k, \forall, X_k, ..., X_1) = 2^{X_k} \rightarrow TP(k-1, \exists, X_{k-1}, ..., X_1)$
- $\pi = \begin{bmatrix} (a), & \mapsto (\neg b); (\lambda_c) \\ (\neg a) & \mapsto (b); (\lambda_c) \end{bmatrix}$ is a total policy for $P\prime = \langle 3, \forall, \{a\}, \{b\}, \{c\}, (\neg a \vee \neg b \vee c) \wedge (a \vee b) \rangle$

Total policies (cont'd)

- A policy π of $TP(k, q, X_k, ..., X_1)$ **satisfies** $P = \langle k, q, X_k, ..., X_1, \Phi\rangle$ of $\text{QBF}_{k,q}$ (denoted $\pi \models P$) iff one of these conditions is verified:
  - $k = 0$ and $\pi = \lambda$, and $\Phi \equiv \top$;
  - $k \geq 1$ and $q = \exists$ and $\pi = (\vec{x_k}; \pi')$ with $\pi' \models \langle k-1, \forall, X_{k-1}, ..., X_1, \Phi_{\vec{x_k}}\rangle$;
  - $k \geq 1$ and $q = \forall$ and for all $\vec{x_k} \in 2^{X_k}$ we have $\pi(\vec{x_k}) \models \langle k-1, \exists, X_{k-1}, ..., X_1, \Phi_{\vec{x_k}}\rangle$
- $\pi = \begin{bmatrix} (a) & \mapsto (\neg b); \lambda_c \\ (\neg a) & \mapsto (b); \lambda_c \end{bmatrix}$ satisfies $P\prime = \langle 3, \forall, \{a\}, \{b\}, \{c\}, (\neg a \vee \neg b \vee c) \wedge (a \vee b)\rangle$

FQBF

- Let $P = \langle k, q, X_k, ..., X_1, \Phi \rangle$ be a QBF. Solving the function problem FQBF$_{k,q}$ for P consists in **generating a total policy** π such that $\pi \models P$, if there exists any
- Policies are the **expected outputs** for the function problem associated to QBF: $P = \langle k, q, X_k, ..., X_1, \Phi \rangle$ is a **positive instance** of QBF$_{k,q}$ iff **there exists a solution policy**, i.e., a total policy $\pi \in TP(k, q, X_k, ..., X_1)$ such that $\pi \models P$

Partial policies

- A solution policy is often **too much demanding**
- $P = \forall a \; \exists b \; \forall c \; (\neg a \vee c) \wedge (a \vee b)$ has no solution policy
- Nevertheless, if the $\forall$ player assigns 0 to a, then the $\exists$ player **can win** (just assign 1 to b)
- This calls for a notion of partial policy: The set $PP(k, q, X_k, ..., X_1)$ of **partial policies** for the QBF $P = \langle k, q, X_k, \ldots, X_1 \rangle$ is defined inductively as follows ($\times$ means "failure"):
  - $PP(1, \exists, X_1) = 2^{X_1} \cup \{\times\}$;
  - $PP(1, \forall, X_1) = 2^{X_1} \rightarrow \{\lambda, \times\}$;
  - $PP(k, \exists, X_k, ..., X_1) =$
    $\{\vec{x_k}; \pi_{k-1} \,|\pi_{k-1} \in PP(k-1, \forall, X_{k-1}, .., X_1)\} \cup \{\times\}$;
  - $PP(k, \forall, X_k, ..., X_1) = 2^{X_k} \rightarrow PP(k-1, \exists, X_{k-1}, ..., X_1)$

Example (cont'd)

- $P = \forall a \ \exists b \ \forall c \ (\neg a \vee c) \wedge (a \vee b)$
- $\pi_1 = \left[\begin{array}{ll} (a) & \mapsto b; \left[\begin{array}{ll} (c) & \mapsto \lambda \\ (\neg c) & \mapsto \times \end{array} \right] \\ (\neg a) & \mapsto (b); \lambda_c \end{array} \right]$
- $\pi_2 = \left[\begin{array}{ll} (a) & \mapsto \times \\ (\neg a) & \mapsto (b); \lambda_c \end{array} \right]$
- $\pi_3 = \left[\begin{array}{ll} (a) & \mapsto \times \\ (\neg a) & \mapsto (\neg b); \lambda_c \end{array} \right]$

Sound policies

- A partial policy $\pi \in PP(k, q, X_k, ..., X_1)$ is **sound** for $P = \langle k, q, X_k, ..., X_1, \Phi\rangle$ iff one of these conditions is satisfied:
  - $q = \exists$ and $\pi = \times$;
  - $(k, q) = (1, \exists)$, $\pi = \vec{x_1}$ and $\vec{x_1} \models \Phi$;
  - $(k, q) = (1, \forall)$ and for any $\vec{x_1} \in 2^{X_1}$ we have either $\pi(\vec{x_1}) = \times$, or ($\pi(\vec{x_1}) = \lambda$ and $\vec{x_1} \models \Phi$);
  - $k > 1$, $q = \exists$, $\pi = \vec{x_k}; \pi_{k-1}$ and π_{k-1} is sound for $\langle k-1, \forall, X_{k-1}, ..., X_1, \Phi_{\vec{x_k}}\rangle$;
  - $k > 1$, $q = \forall$, and for any $\vec{x_k} \in 2^{X_k}$, $\pi(\vec{x_k})$ is sound for $\langle k-1, \exists, X_{k-1}, ..., X_1, \Phi_{\vec{x_k}}\rangle$
- π_1 and π_2 are sound for P, but π_3 is not

Maximal sound policies

- Let π and π' two partial policies of $PP(q, k, X_k, ..., X_1)$. π is **at least as covering as** π', denoted by $\pi \sqsupseteq \pi'$, iff one of the following conditions is satisfied:
  - $q = \exists$ and $\pi' = \times$;
  - $q = \forall$, $k = 1$ and for all $\vec{x_1} \in 2^{X_1}$, we have either $\pi'(\vec{x_1}) = \times$ or $\pi(\vec{x_1}) = \lambda$;
  - $q = \exists$, $\pi = [\vec{x_k}; \pi_{k-1}]$, $\pi' = [\vec{x'_k}; \pi'_{k-1}]$, and $\pi_{k-1} \sqsupseteq \pi'_{k-1}$;
  - $q = \forall$, $k > 1$ and for all $\vec{x_k} \in 2^{X_k}$, we have $\pi(\vec{x_k}) \sqsupseteq \pi'(\vec{x_k})$
- π is a **maximal sound** policy for a QBF P iff π is sound for P and there is no sound policy π' for P such that $\pi' \sqsupseteq \pi$ and $\pi \not\sqsupseteq \pi'$
- We have $\pi_1 \sqsupseteq \pi_2$ and π_1 is a maximal sound policy for P

SFQBF

- **Every QBF P has a maximal sound policy**
- **If a solution policy for P exists, then solution policies and maximal sound policies coincide**
- Solving the second function problem $\text{SFQBF}_{k,q}$ for P consists in finding a maximal sound policy π for P

Policy representation

- Policy $\pi \neq$ representation σ of π
- A **representation scheme** $\mathcal{S}$ for policies is a set of data structures representing policies. Associated with any representation scheme $\mathcal{S}$ is an **interpretation function** $I_{\mathcal{S}}$ such that for any $\sigma \in \mathcal{S}$, $\pi = I_{\mathcal{S}}(\sigma)$ is the policy represented by σ
- The simplest representation scheme is the **explicit** one: the representation of a policy is the policy itself
- Explicit representations of total policies are **certificates** for QBFs: there is a **polytime algorithm** whose input is the explicit representation of a policy $\pi \in TP(k, q, X_k, ..., X_1)$ and a QBF $P = \langle k, q, X_k, ..., X_1, \Phi \rangle$ and which returns 1 if π is a solution policy for P and 0 otherwise

Example

- The explicit representation of $\pi \in TP(1, \exists, X_1)$ is a world $\vec{x_1} \in 2^{X_1}$
- The explicit representation of $\pi \in TP(1, \forall, X_1)$ is the set of pairs $\{(\vec{x_1}, \lambda) \mid \vec{x_1} \in 2^{X_1}\}$
- It can be represented in the exponentially more succinct way as λ_{X_1} but this representation is not a certificate for $\text{QBF}_{1,\forall}$ unless P = NP
- The existence of a certificate of polynomial size for $\text{QBF}_{1,\forall}$ would imply NP = coNP

$\text{SFQBF}_{2,\forall}$

- The simplest problem for which the size of a representation of a policy is a significant problem
- In the case of $\text{SFQBF}_{2,\forall}$, a partial policy for $P = \langle 2, \forall, X, Y, \Phi \rangle$ is a function π from 2^X to $2^Y \cup \{\times\}$
- A policy representation scheme $\mathcal{S}$ for maximal sound policies for $\text{QBF}_{2,\forall}$ is said to be **polynomially compact** iff there is a polysize function $R_{\mathcal{S}}$ that associates each $P = \langle 2, \forall, X, Y, \Phi \rangle \in \text{QBF}_{2,\forall}$ to a representation $\sigma \in \mathcal{S}$ of a maximal sound policy π for P
- A policy representation scheme $\mathcal{S}$ for maximal sound policies for $\text{QBF}_{2,\forall}$ is said to be **tractable** iff there exists a polytime algorithm $D_{\mathcal{S}}$ such that for any $\sigma \in \mathcal{S}$, $D_{\mathcal{S}}$ computes $\pi(\vec{x}) = D_{\mathcal{S}}(\sigma, \vec{x})$ for any $\vec{x} \in 2^X$, where $\pi = I_{\mathcal{S}}(\sigma)$

Representation schemes

- The explicit representation scheme is not polynomially compact:

$$\forall\{x_1,\ldots,x_n\}\exists\{y_1,\ldots,y_n\}\bigwedge_{i=1}^{n}(x_i \Leftrightarrow y_i)$$

- **If a polynomially compact and tractable representation scheme $\mathcal{S}$ for maximal sound policies for $\mathrm{QBF}_{2,\forall}$ exists, then the polynomial hierarchy collapses at the second level**

- Relaxing the polynomial compacity requirement, we look for tractable representations of policy: a representation σ of a policy π for $P = \langle 2, \forall, X, Y, \Phi\rangle \in \mathrm{QBF}_{2,\forall}$ is said to be **tractable** if and only if there exists an algorithm D_σ such that for any $\vec{x} \in 2^X$, D_σ computes $\pi(\vec{x}) = D_\sigma(\vec{x})$ in time polynomial in $|\sigma| + |\vec{x}|$

The decomposition approach

Two ideas

- It is often needless looking for a specific Y-instantiation for each X-instantiation: some Y-instantiations may **cover** large sets of X-instantiations, which can be described **in a compact way**, for instance by a propositional formula.
- It may be the case that some sets of variables from Y are more or less **independent** given X w.r.t. Φ and therefore that their assigned values can be computed separately

Partial subpolicies

- A **partial subpolicy** is a function from $2^X \to 3^Y$ associating consistent Y-terms (**subdecisions**) to some X-worlds
- The **merging** of subdecisions is the commutative and associative internal operator on $3^Y \cup \{\times\}$ defined by:
  - $\gamma_Y.\lambda = \lambda.\gamma_Y = \gamma_Y$;
  - $\gamma_Y.\times = \times.\gamma_Y = \times$;
  - if γ_Y, γ'_Y are two Y-terms, then

$$\gamma_Y.\gamma'_Y = \left\{ \begin{array}{ll} \gamma_Y \wedge \gamma'_Y & \text{if } \gamma_Y \wedge \gamma'_Y \textit{ is consistent} \\ \times & \text{otherwise} \end{array} \right\}$$

- The **merging** of two subpolicies π_1, π_2 is defined by: $\forall \vec{x} \in 2^X$, $(\pi_1 \odot \pi_2)(\vec{x}) = \pi_1(\vec{x}).\pi_2(\vec{x})$

Example

- $\sigma_1 = \texttt{if } x_1 \Leftrightarrow x_2 \texttt{ then } y_1 \texttt{ else } \neg y_1$
- $\sigma_2 = \texttt{if } x_1 \texttt{ then } \neg y_2$
- $\sigma = \sigma_1 \odot \sigma_2$
- The corresponding policies are given by

	$I_{PD}(\sigma_1)$	$I_{PD}(\sigma_2)$	$I_{PD}(\sigma)$
(x_1, x_2)	y_1	$\neg y_2$	$(y_1, \neg y_2)$
$(x_1, \neg x_2)$	$\neg y_1$	$\neg y_2$	$(\neg y_1, \neg y_2)$
$(\neg x_1, x_2)$	$\neg y_1$	$\times$	$\times$
$(\neg x_1, \neg x_2)$	y_1	$\times$	$\times$

The PD scheme

- The **policy description scheme** PD is a representation scheme for maximal sound policies for $\text{QBF}_{2,\forall}$, defined inductively as follows:
  - λ and $\times$ are in PD;
  - any consistent Y-term γ_Y is in PD;
  - if φ_X is a propositional formula built on X and σ_1, σ_2 are in PD, then `if` φ_X `then` σ_1 `else` σ_2 is in PD
- PD **is a tractable representation scheme for maximal sound policies for** $\mathbf{QBF}_{2,\forall}$

Example (cont'd)

A tractable polysize representation in PD of the solution policy for

$$\forall\{x_1,\ldots,x_n\}\exists\{y_1,\ldots,y_n\}\bigwedge_{i=1}^{n}(x_i \Leftrightarrow y_i)$$

is

$$\sigma = \odot_{i=1}^{n}((\texttt{if}\ x_i\ \texttt{then}\ y_i) \odot (\texttt{if}\ \neg x_i\ \texttt{then}\ \neg y_i))$$

Decomposition

- Let $P = \langle 2, \forall, X, Y, \Phi \rangle$ and let $\{\varphi_1^X, \varphi_1^Y, \ldots, \varphi_p^X, \varphi_p^Y\}$ be $2p$ formulas such that
$$\Phi \equiv (\varphi_1^X \wedge \varphi_1^Y) \vee \ldots \vee (\varphi_p^X \wedge \varphi_p^Y)$$
Let $J = \{j \mid \varphi_j^Y \text{ is consistent}\} = \{j_1, \ldots, j_q\}$ and for every $j \in J$, let $\vec{y}_j \models \varphi_j^Y$. **Then the policy π represented by the description**
$$\sigma = \texttt{Case } \varphi_{j_1}^X\text{: } \vec{y}_{j_1}\text{; \ldots; } \varphi_{j_q}^X\text{: } \vec{y}_{j_q} \texttt{ End}$$
is a maximal sound policy for P

- When Φ has the required form, problem for P comes down to solving p SAT instances

- **A decomposition of Φ always exist**: $\Phi \equiv \bigvee_{\vec{x} \in 2^X} (\vec{x} \wedge \Phi_{\vec{x}})$

- When Φ is in DNF, the second function problem **can be solved in polynomial time** (while the decision problem is coNP-complete)

Decomposition (cont'd)

- Let $\{Y_1, Y_2\}$ be a partition of Y such that Y_1 and Y_2 are conditionally independent given X with respect to Φ, which means that there exist two formulas φ_{X,Y_1} and φ_{X,Y_2} of respectively $PROP_{X \cup Y_1}$ and $PROP_{X \cup Y_2}$ such that $\Phi \equiv \varphi_{X,Y_1} \wedge \varphi_{X,Y_2}$
- π **is a maximal sound policy for** P **iff there exist two subpolicies** π_1**,** π_2**, which are maximal and sound for** $\forall X \exists Y \varphi_{X,Y_1}$ **and for** $\forall X \exists Y \varphi_{X,Y_2}$ **respectively, such that** $\pi = \pi_1 \odot \pi_2$
- Find a $\{Y_1, Y_2\}$ which can be turned into such a partition through case analysis on the common variables (use a decomposition tree)

The compilation approach

- Let $P = \forall X\ \exists Y\ \Phi$ be a QBF and let σ be a propositional formula equivalent to Φ and which belongs to a propositional fragment $\mathcal{F}$ enabling polytime conditioning and polytime model finding. σ **is a tractable representation of a maximal sound policy for** P
- σ alone does not represent any policy for P but a specific maximal sound policy for P is fully characterized by the way a model of $\sigma_{\vec{x}}$ is computed for each $\vec{x}$
- Many $\mathcal{F}$ are candidates: Krom, Horn CNF, renamable Horn CNF, DNF, OBDD, DNNF
- No guarantee that σ remains "small" in the worst case (but may work quite well in practice)

Conclusion and perspectives

- Function problems for QBF
- Theoretical limitations on representation schemes
- Two approaches for solving $\text{SFQBF}_{2,\forall}$
- Perspectives
  - Experimentations
  - Extension to other forms of maximality for policies (gradual satisfaction)