Nonmonotonic Reasoning with Quantified Boolean Constraints

Chris Pollett and Jeff Remmel

June 27, 1997
Overview

1. Motivation

2. Quantified Boolean formulas and the polynomial hierarchy

3. Our $QBF_k$ formalisms
   (a) $LP_k$, $CC_k$, $DL_k$

4. Compactness of knowledge representation
   (a) Succinctness results
Motivation

Constraint Logic Programming (JL)
- Allow more general constraints to body of LP clauses. For instance, CLP(ℝ) programs allow real inequalities and equalities as constraints.
- Constraints may be solved by special resources.
- Domain where constraints evaluated is fixed.

Constraint Programs (MNR)
- Is an extension of CLP paradigm.
- Domain is not fixed. An example use is in controlling a plant where set of applicable rules depends on plant’s state at discrete intervals.
- Constraints need to be satisfied before evaluate remainder of clause.
- Constraints true in model consisting of atoms computed in process.
- Can still use special hardware.

Generalized constraints are a new source of complexity to be studied. We will discuss this complexity for the propositional case of a variety of nonmonotonic formalisms.
The Polynomial Hierarchy (PH)

$P = \Delta^p_1$ = deterministic p-time
$NP = \Sigma^p_1$ = nondeterministic p-time
$\Delta^p_{i+1} = P^{\Sigma^p_i}$, $\Sigma^p_{i+1} = NP^{\Sigma^p_i}$, $\Pi^p_i = co - \Sigma^p_i$

$PH = \cup \Sigma^p_k$

Open: $PH \not\subseteq \Sigma^p_2$?

Quantified Boolean Formulas

$\Sigma_0^q = \Pi_0^q$ - propositional formulas
$\Sigma_k^q \supseteq \Pi_k^q$ closed under ($\exists x$) where intended meaning of ($\exists x)A(x, \bar{b})$ is $A(0, \bar{b}) \lor A(1, \bar{b})$.
$\Pi_k^q \supseteq \Sigma_k^q$ closed under ($\forall x$) where intended meaning of ($\forall x)A(x, \bar{b})$ is $A(0, \bar{b}) \land A(1, \bar{b})$.

$QBF_k$ - boolean combinations of $\Sigma_k^q$ and $\Pi_k^q$.

$QBF_k(A)$ - a $QBF_k$ formula where allow atoms $x_1, \ldots, x_n \in A$. View $x_1, \ldots, x_n$ as binary for a number and ask if in set $A$.

**FACT**: Validity of $\Sigma_k^q$-sentences is $\Sigma_k^p$-complete.
Our system for logic programming \( LP_k \)

An \( LP_k \) program \( P \) is a finite list of clauses:

\[
p \leftarrow a_1, \ldots, a_n : B_1(\vec{b}_1), \ldots, B_n(\vec{b}_m) \quad (\star)
\]

where \( p, a_1, \ldots, a_n \) are variables and \( B \in QBF_k \).

\( LP_{\infty} = \cup LP_k \). \( LP_k(A) \) - constraints from \( QBF_k(A) \).

**Stable Model Semantics** Let \( P \in LP_k \), \( M \) be a subset of \( P \)'s vars. Let \( \nu_M \) be truth assign.
induced by \( M \). Let \( P_M \) be obtained by deleting clauses whose constraints aren’t satisfied by \( \nu_M \) and by deleting the constraints from what’s left. Let \( N_M \) be least model of \( P_M \). \( M \) is a **stable model** of \( P \) if \( M = N_M \).

**Supported Model Semantics** A supported model of \( P \in LP_k \) is a truth assign. \( \nu \) to vars in \( P \) such that \( \nu(p) = 1 \) iff \( \exists \) a clause \((\star)\) in \( P \) and \( (\forall i, j) \)
\( \nu(a_i) = 1, \nu(B(b_j)) = 1 \). We write \( LP_k^{sup} \) if considering supported models. We write \( LP_k^{*} \) for programs with pairwise disjoint supported models.
Theorem
1. $LP_0$ is equivalent to logic programming with negation.
2. Whether an $LP_k$ program has a model is $\Sigma^p_{k+1}$-complete.
3. Whether an $LP_\infty$ program has a model is $PSPACE$-complete.
Our system for circumscription $CC_k$

Circumscribed models of a prop. formula are minimal models under inclusion. Could look at minimal models of $QBF_k$ formulas. This doesn't separate constraints from computational component. Instead, $CC_k$ program $P$ is a finite list of clauses:

$$B(\vec{a}) \iff C(\vec{b})$$

with $B \in QBF_0$ and $C \in QBF_k$. $CC_\infty = \cup CC_k$. $CC_k(A)$ - constraints from $QBF_k(A)$

**Semantics** Let $S$ be a subset of vars in $P$ and let $\nu_S$ be corresponding var. assignment. Define $\bigwedge P_S := \bigwedge_{B \in P_S} B$ where $P_S$ is

$$\{B \mid B \iff C \in P \land \nu_S(C) = 1\}.$$  

A model of $P$ is a model of the 2nd-order formula $\bigwedge P_M \land \neg \exists m[\bigwedge P_m \land m \subset M]$.  

**Theorem**

1. $CC_0$ is equivalent to prop. circumscription.
2. Whether a variable occurs in all models of a $CC_k$ program is $\Pi_{k+2}^p$-complete.
3. The problem for $CC_\infty$ is $PSPACE$-complete.
Our system for default logic $DL_k$

A $DL_k$ theory is a pair $\langle D, W \rangle$. Here $D$ is a finite collection of default rules:

$$\frac{\alpha : B_1(b_1), \ldots, B_m(b_m)}{\gamma}$$

where $\alpha, \gamma \in QBF_0$ and $B_i \in QBF_k$. $W$ is a finite set of prop. formulas. $DL_\infty = \cup DL_k$.

$DL_k(A)$ - constraints from $QBF_k(A)$

**Stable Model Semantics** A rule $d$ is **S-applicable** if $B_i \cup S$ is consistent for each constraint in $d$. Form $D_S$ by deleting non-S-applicable rule from $D$ and deleting constraints from rest. Let $Cn^X(W)$ be all formulas provable from $W$ using rules in $X$ and prop logic. An **extension** for $\langle D, W \rangle$ is a set of formulas $S$ such that $Cn^{DS}(W) = S$. A **stable model** for $\langle D, W \rangle \in DL_k$ is a truth assign. satisfying an extension of $\langle D, W \rangle$. 

7
Supported Model Semantics A rule \( d \) is strongly \( S \)-applicable if it is \( S \)-applicable and the prerequisite \( \alpha \) of \( d \) is in \( S \). Form \( D_{S,w} \) as \( D_s \) but use strong \( S \)-applicability. A weak extension for \( \langle D, W \rangle \) is a set of formulas \( S \) such that \( Cn^{D_{S,w}}(W) = S \). A supported model for \( \langle D, W \rangle \in DL_k \) is a truth assign. satisfying an weak extension of \( \langle D, W \rangle \). We write \( DL_k^{sup} \) if considering stable models.

Theorem
1. \( DL_0 \) is the same as usual default logic.
2. Whether \( \langle D, W \rangle \in DL_k \) has an extension is \( \Sigma_{k+2}^p \)-complete.
3. The problem for \( DL_\infty \) it is \( PSPACE \)-complete.
Compactness of knowledge representation

**Definition** (GKPS, CDS) Let $A$ and $B$ be reasoning formalisms. Then $A$ as succinct as $B$, written $B \leq_s A$ if: For each $\phi_B$ in $B$ there is a knowledge base $\phi_A$ in $A$ such that
(a) $\phi_B$ and $\phi_A$ use free variables and have the same models
(b) the size of $\phi_A$ is polynomial in the size of $\phi_B$.
We write $A \not\leq_s B$ if (a) and (b) fail to hold.

**Definition** We say $A$ is as weak succinct as $B$, written $B \leq_{ws} A$ if the conditions above hold but condition (a) is replaced with
(a') $\phi_A$ contains all of $\phi_B$'s variables and all models of $\phi_A$ are expansions of models of $\phi_B$.

- GKPS give a succinctness hierarchy among
  prop logic, Horn logic, circumscription and default logic which is strict provided PH ≠ L.
Hierarchies of Knowledge Formalisms

(a) $LP_k <_s DL_k$

(b) $LP^*_k <_s CC_k <_s DL_k$

(c) $LP^*_k \equiv_{ws} LP^*_k \equiv_{ws} LP_k <_{ws} CC_k \leq_{ws} CC_k \leq_{ws} DL^*_k \equiv_{ws} DL_k \equiv_{ws} LP_{k+1}$.

- Strictness in above under assumption $PH \not\subseteq \Sigma^p_{k+1}$.

(d) $\exists A, LP_k(A) <_{ws} CC_k(A) <_s DL_k(A)$

(e) $LP_\infty \equiv_{ws} CC_\infty \equiv_{ws} DL_\infty$

**Theorem** If $K$ is a reasoning formalism with $\Delta^p_{k+1}$-model checking then $CC_k \not\leq_s K$ unless $\Sigma^p_{k+1} \subseteq \Delta^p_{k+1/poly}$.

**Theorem** Suppose $K$ is a reasoning formalism for which $Model_{\phi_K}(\vec{x})$ can be expressed as a $QBF_k$ formula. Then $K \leq_{ws} LP_k$. 

10