Beyond NP: Quantifying over Answer Sets Supplementary Material

GIOVANNI AMENDOLA[1]

FRANCESCO RICCA[1]

MIREK TRUSZCZYNSKI[2]

[1] *University of Calabria, Rende, Italy*
(*e-mail:* {amendola,ricca}@mat.unical.it)
[2] *University of Kentucky, KY, USA*
(*e-mail:* mirek@cs.uky.edu)

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Appendix A Proofs of Some Theorems

In the following we report the main theorems presented in the paper with their proofs.

Theorem 1
Let P be an ASP program, and let Π be the ASP(Q) program of the form (2), where $n = 1$, $\Box_1 = \exists^{st}$, $P_1 = P$, and $C = \emptyset$. Then, $AS(P) = QAS(\Pi)$.

Proof
By definition, M is a quantified answer set of Π if and only if M is an answer set of P and $\emptyset \cup fix_P(M) = fix_P(M)$ is coherent. The latter condition is trivially true as M is an answer set of $fix_P(M)$. $\Box$

Theorem 2
The COHERENCE problem is PSPACE-complete, even under the restriction to normal ASP(Q) programs.

Proof
(Membership) It is well known that answer sets of a disjunctive logic program can be enumerated in polynomial space in the size of the program. Let us assume that p is a polynomial providing that bound. We prove that the coherence of an $ASP(Q)$ program Π of the form (2) can be decided in space $O(n \times p(s(\Pi)))$, where $s(\Pi)$ is the size of Π, and n is the number of quantifiers in Π.

To this end, we consider the following recursive algorithm. It consists of enumerating all answer sets of P_1. If $n = 1$, we have $\Pi = \Box P_1 : C$. To decide coherence, for each enumerated answer set M of P_1, we decide whether $C \cup fix_{P_1}(M)$ is coherent. Depending on whether $\Box = \exists^{st}$ or $\forall^{st}$, if for some (every) answer set M of P_1, $C \cup fix_{P_1}(M)$ is coherent, we return that Π is coherent. Otherwise, we return that Π is not coherent. For $n \geq 2$, for each enumerated answer set M of P_1, we recursively check whether $\Pi' = (\Box_2 P_2 \ldots \Box_n P_n : C)_{P_1,M}$ is coherent, and decide about coherence of Π similarly as in the case $n = 1$, depending on the outermost quantifier.

By the comment above, we can enumerate all answer sets M of P_1 in space $O(p(s(\Pi)))$ (indeed, $s(P_1) = O(s(\Pi))$). Moreover, if $n = 1$, testing coherence of $C \cup fix_{P_1}(M)$ can be accomplished in time and so, also in space $O(s(C \cup fix(M))) = O(s(\Pi))$. Thus, if $n = 1$, the algorithm requires $O(p(s(\Pi)))$ space, establishing the base case of the induction. If $n \geq 2$, we need $O(p(s(\Pi)))$ space for enumerating answer sets and, using the induction hypothesis, $O((n-1) \times p(s(\Pi)))$ space for each recursive call. Thus, the total space requirement is $O(n \times p(s(\Pi)))$, completing the inductive step.

We now observe that $n = O(s(\Pi))$, which shows that the algorithm we described runs in space $O(s(\Pi) \times p(s(\Pi)))$. This implies the assertion.

(Hardness) We give a reduction from the problem of deciding the validity of a QBF formula $\Phi = Q_1 x_1 \ldots Q_n x_n \varphi$, where for every $i = 1, \ldots, n$, $Q_i \in \{\exists, \forall\}$ and x_i is a propositional variable, and where φ is a propositional formula over $\{x_1, \ldots, x_n\}$. The problem is PSPACE-complete even when φ is in 3-CNF. Thus, let us assume that $\varphi = C_1 \wedge \ldots \wedge C_m$, where $C_j = l_j^1 \vee l_j^2 \vee l_j^3$ and $l_j^1, l_j^2, l_j^3 \in \{x_i, \neg x_i \mid i = 1, \ldots, n\}$, for each $j = 1, \ldots, m$. We construct an ASP(Q) program Π as follows. For each $i = 1, \ldots, n$, we define $P_i = \{x_i \leftarrow not\ nx_i;\ nx_i \leftarrow not\ x_i\}$ and $\Box_i = Q_i^{st}$. We also define $C = \{ok_j \leftarrow \sigma(l_j^h) \mid j = 1, \ldots, m \text{ and } h = 1,2,3\} \cup \{\leftarrow not\ ok_i \mid i = 1, \ldots, m\}$, where $\sigma(l) = x_i$ if $l = x_i$, and $\sigma(l) = nx_i$ if $l = \neg x_i$. It is easy to see that Π is coherent iff Φ is valid. Moreover, as each program P_i is normal, Π is a normal ASP(Q) program. $\Box$

Theorem 3

The COHERENCE problem is (*i*) Σ_n^P-complete for normal existential ASP(Q) programs with n quantifiers in the prefix; and (*ii*) Π_n^P-complete for normal universal ASP(Q) programs with n quantifiers in the prefix.

Proof

(Membership) We proceed by induction on n. We start with $n = 1$. If $\Pi = \exists^{st} P_1 : C$ then deciding coherence amounts to checking whether there is an answer set I of P_1 such that $fix_{P_1}(I) \cup C$ is coherent. This problem is in NP ($= \Sigma_1^P$) because one can check coherence of a normal stratified program with constraints in polynomial time (Dantsin et al. 2001). If $\Pi = \forall^{st} P_1 : C$ then deciding coherence amounts to checking whether there is no answer set I of P_1 such that $fix_{P_1}(I) \cup C$ is not coherent. This problem is in co-NP ($= \Pi_1^P$) because its complement, the problem to decide whether there is an answer set I of P_1 such that $fix_{P_1}(I) \cup C$ is not coherent, is in NP (indeed, one can check coherence of a normal stratified program with constraints in polynomial time).

Next, let us assume that $n \geq 2$. Further, let Π be a normal ASP(Q) program of the form (2). If $\Box_1 = \exists^{st}$, then to decide coherence of Π we have to decide whether there is an interpretation I such that I is an answer set of P_1 and $\Pi_{P_1,I}$ is coherent. Checking that I is an answer set of P is a polynomial-time task (we recall that P_1 is normal). Checking that $\Pi_{P_1,I}$ is coherent can be accomplished with a call to an oracle for a problem in Σ_{n-1}^P or in Π_{n-1}^P depending on whether $\Box_2$ in Π is $\exists^{st}$ or $\forall^{st}$. Indeed, by the induction hypothesis, the problem of deciding coherence for normal $ASP(Q)$ programs with $n-1$ quantifiers and with the outermost quantifier fixed to $\exists^{st}$ ($\forall^{st}$, respectively) is in Σ_{n-1}^P (Π_{n-1}^P, respectively).

If $\Box_1 = \forall^{st}$, to decide coherence of Π we have to decide that for every answer set of P_1, $\Pi_{P_1,I}$ is coherent. The complement to this problem consists of deciding whether there is an an answer set I of P_1 such that $\Pi_{P_1,I}$ is not coherent. By a similar argument as above, this problem is in Σ_n^P (observe that an oracle deciding whether an $ASP(Q)$ program is coherent, can be used to decide whether an $ASP(Q)$ program is not coherent). It follows that deciding coherence for programs with n quantifiers in the prefix and with $\forall^{st}$ as the outermost quantifier is in Π^P_n.

(Hardness) Let us consider a QBF $\Phi = Q_1X_1 \dots Q_nX_n\varphi$, where $X_1, \dots, X_n$ are disjoint sets of propositional variables, each $Q_i = \exists$ or $\forall$, the quantifiers alternate, and φ is a 3-CNF or 3-DNF formula over the variables in $X_1 \cup \dots \cup X_n$. We encode Φ as an $ASP(Q)$ program Π_Φ of the form (2) as follows. For every $i = 1, \dots, n$, we set $\Box_i = Q^{st}_i$ and $P_i = \{x \leftarrow not\ nx \mid x \in X_i\} \cup \{nx \leftarrow not\ x \mid x \in X\}$ (similarly as in the previous proof). If φ is a 3-CNF formula, we define a normal stratified program with constraints C as in the previous proof. So, assume φ is a 3-DNF formula, say $\varphi = D_1 \vee \dots \vee D_m$, where $D_j = l^1_j \wedge l^2_j \wedge l^3_j$ and $l^1_j, l^2_j, l^3_j \in X_1 \cup \dots \cup X_n$, for each $j = 1, \dots, m$. In this case, we set $C = \{ok_j \leftarrow \sigma(l^1_j), \sigma(l^2_j), \sigma(l^3_j) \mid j = 1, \dots, m\} \cup \{\leftarrow not\ ok_1, \dots, not\ ok_m\}$, where $\sigma(l) = x$ if $l = x$, and $\sigma(l) = nx$ if $l = \neg x$.

It is easy to see that in both cases Φ is valid iff Π_Φ is coherent. Moreover, both encodings can be obtained by a polynomial-time procedure. Now, according to well-known complexity results (Stockmeyer 1976) the problem to decide validity for QBFs such that (1) $Q_1 = \exists$, φ is in 3-DNF, and n is even; (2) $Q_1 = \exists$, φ is in 3-CNF, and n is odd; (3) $Q_1 = \forall$, φ is in 3-CNF, and n is even; (4) $Q_1 = \forall$, φ is in 3-DNF and n is odd is Σ^P_n-complete for the cases (1) and (2), and Π^P_n-complete for the cases (3) and (4). Thus, the hardness follows. □

Theorem 7

(i) There is a polynomial-time reduction that assigns to every propositional nested combined program Π of depth n, a normal existential ASP(Q) program Π_q with $n \geq 2$ quantifiers such that answer sets of Π and Π_q, correspond to each other.

(ii) There is a polynomial-time reduction that assigns to every propositional normal existential ASP(Q) program Π with $n \geq 2$ quantifiers in the prefix, a propositional nested combined program Π_c of depth n such that answer sets of Π and Π_c correspond to each other.

Proof

Follows from the fact that checking the existence of stable unstable model of combined programs of depth n is Σ^P_n-complete as well as checking coherence of a normal existential ASP(Q) program with n quantifiers. □

Proposition 1

Unless the polynomial hierarchy collapses, there exists no polynomial reduction that encodes formulas $\Psi = \forall X \exists Y \psi$, where ψ is a 3-CNF formula, as a combined program $P = (P_1, P_2)$, where P_1 and P_2 are normal logic programs, such that Ψ is valid iff P admits stable unstable models.

Proof

The thesis follows from the observation that checking the validity of Ψ is a Π^P_2-complete problem, whereas the existence of stable unstable model of the combined program P is Σ^P_2-complete. □