Online appendix for the paper

Anytime Computation of Cautious Consequences in Answer Set Programming

published in Theory and Practice of Logic Programming

MARIO ALVIANO, CARMINE DODARO and FRANCESCO RICCA
Department of Mathematics and Computer Science, University of Calabria, 87036 Rende (CS), Italy
(*e-mail:* {alviano,dodaro,ricca}@mat.unical.it)

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

Appendix A Proof of Theorem 1

The proof is split into several lemmas using P_i, L_i, U_i, O_i, I_i to denote the content of variables P, L, U, O, I at step i of computation ($i \geq 0$). More in detail, in Lemma 1 we will first show that underestimates form an increasing sequence and, on the contrary, overestimates form a decreasing sequence. Then, in Lemma 2 we will prove properties of stable models of programs $P_i \cup L_i$ ($i \geq 0$). Correctness of estimates will be shown in Lemmas 3–4, and termination of the algorithms in Lemma 5. Finally, in Lemma 6 we will extend the proof to variants using ComputeStableModel*.

Lemma 1
$U_i \subseteq U_{i+1}$ and $O_{i+1} \subseteq O_i \subseteq Q$ for each $i \geq 0$.

Proof
Variable U is initially empty. EnumerationOfModels and OverestimateReduction reassign U only once. IterativeCoherenceTesting and UnderestimateReduction always enlarge the set stored in U by means of set union (line 4). Concerning variable O, it is initially equal to Q and restricted at each reassignment by means of set intersection (line 7 for OverestimateReduction; line 6 for the other procedures). □

Lemma 2
$SM(P_{i+1} \cup L_{i+1}) \subseteq SM(P_i \cup L_i)$ for each $i \geq 0$. For IterativeCoherenceTesting and IterativePartialCoherenceTesting we also have $SM(P_{i+1} \cup L_{i+1}) = SM(P_i \cup L_i)$ for each $i \geq 0$.

Proof
Variable P is reassigned only by EnumerationOfModels and OverestimateReduction, where constraints are added to the previous program. Constraints can only remove stable models (as a consequence of the Splitting Set Theorem by Lifschitz and Turner 1994). On the other hand, learned constraints stored in variable L are implicit in the program stored by variable P, and thus cannot change its semantics. □

Lemma 3
$O_i \supseteq Q \cap CC(P)$ for each $i \geq 0$.

Proof

The base case is true because $O_0 = Q$. Assume the claim is true for some $i \geq 0$ and consider $O_{i+1} = O_i \cap I_{i+1}$, where $I_{i+1} \in SM(P_i \cup L_i)$. By i applications of Lemma 2, we obtain $I_{i+1} \in SM(P_0 \cup L_0)$, i.e., $I_{i+1} \in SM(P)$. We can thus conclude $a \in O_i \setminus O_{i+1}$ implies $a \notin CC(P)$, and we are done. □

Lemma 4

$U_i \subseteq Q \cap CC(P)$ for each $i \geq 0$.

Proof

The base case is true because $U_0 = \emptyset$. Assume the claim is true for some $i \geq 0$ and consider U_{i+1}. If $U_{i+1} = U_i$ then the claim is true. Otherwise, we distinguish two cases.

For IterativeCoherenceTesting and IterativePartialCoherenceTesting, $U_{i+1} = U_i \cup \{a\}$ for some $a \in O_i \setminus U_i$. Moreover, there is no $M \in SM(P_i \cup L_i)$ such that $a \notin M$ because $I_{i+1} = \bot$. From Lemma 2, we can conclude that there is no $M \in SM(P)$ such that $a \notin M$, i.e., $a \in CC(P)$. Since $a \in O_i \setminus U_i$, we have $a \in O_i$ and thus $a \in Q$ by Lemma 1. Therefore, $a \in Q \cap CC(P)$ and we are done.

For EnumerationOfModels and OverestimateReduction, $U_{i+1} = O_i$ and the algorithm terminates. Exactly $i+1$ constraints were added to P, one for each stable model of P found, i.e., $I_1, \ldots, I_i$. Moreover, $I_{i+1} = \bot$ holds. Assume by contradiction that there is $a \in O_i \setminus CC(P)$. Hence, there is $M \in SM(P)$ such that $a \notin M$. Moreover, $a \in I_j$ $(j = 1, \ldots, i)$ and thus M is a model of all constraints added at line 1. Consequently, M is a stable model of $P_i \cup L_i$, which contradicts $I_{i+1} = \bot$. □

Lemma 5

Algorithm 1 terminates after finitely many steps.

Proof

When EnumerationOfModels is used, termination is guaranteed because P has a finite number of stable models. OverestimateReduction either sets U equal to O, or reduces O, which initially is equal to Q, a finite set. IterativeCoherenceTesting either increases U, or reduces O, and thus terminates because O is finite and $U_i \subseteq O_i$ holds for each $i \geq 0$ by Lemmas 3 and 4. Termination of IterativePartialCoherenceTesting is guaranteed if restarts are properly delayed during the computation, as it must be done already for guaranteeing termination of stable model search. □

Lemma 6

Underestimates produced by ComputeStableModel* are sound.

Proof

Follows by the fact that L contains constraints that are implicit in the program stored by variable P. □

References

LIFSCHITZ, V. AND TURNER, H. 1994. Splitting a logic program. In *Proceedings of the 11th International Conference on Logic Programming (ICLP'94)*, P. Van Hentenryck, Ed. MIT Press, Santa Margherita Ligure, Italy, 23–37.