Guarded resolution for Answer Set Programming

V.W. Marek
J.B. Remmel
Department of Computer Science, University of Kentucky,
Lexington, KY 40506 *
J.B. Remmel
Departments of Mathematics and Computer Science,
University of California at San Diego,
La Jolla, CA 92903 †

April 28, 2009

Abstract

We investigate a proof system based on a *guarded resolution rule* and show its adequacy for stable semantics of normal logic programs. As a consequence, we show that Gelfond-Lifschitz operator can be viewed as a proof-theoretic concept. As an application, we find a propositional theory E_P whose models are precisely stable models of programs.

1 Introduction

In this note, we introduce a rule of proof, called *guarded unit resolution*. Guarded unit resolution is a generalization of a special case of resolution rule, namely, *positive unit resolution*. In positive unit resolution , one of the inputs is a atom clause. Positive unit resolution is complete for Horn clauses, specifically, given a consistent Horn theory T and an atom p, the atom p belongs to the least model of T, $lm(T)$ if and only if there is a positive unit resolution derivation of p using T.

The modification we introduce in this note concerns *guarded atoms* and *guarded Horn clauses*. Guarded atoms are strings of the form: $p : \{r_1, \ldots, r_m\}$ where $p, r_1, \ldots, r_m$ are propositional atoms. Guarded Horn clauses are strings of the form $p \leftarrow q_1, \ldots, q_n : \{r_1, \ldots, r_m\}$ again with $p, q_1, \ldots, q_n, r_1, \ldots, r_m$ propositional atoms.

These guarded atoms and guarded rules will be used to obtain a characterization of stable models of normal logic programs. There are many characterizations of stable models. In fact, Lifschitz, in his [Li08] presents already

*email: marek@cs.uky.edu
†email: jremmel@ucsd.edu

twelve characterizations. The characterization of stable models that we present in this paper has a distinctly proof-theoretic flavor and makes easy to prove some basic results on Answer Set Programming such as Fages' Theorem [Fa94], the Erdem-Lifschitz Theorem [EL03], and Dung's Theorem [DK89].

The outline of this paper is as follows. First, we introduce the definition of the guarded resolution rule of proof and then derive its connection to the Gelfond-Lifschitz operator [GL88]. Once we do this, we will get the desired lifting of the Kowalski-Kuehner [KK71] result on the completeness of unit resolution for Horn theories to the context of stable semantics of logic programs.

2 Guarded resolution and Stable Semantics

By a *logic program clause* we mean a string of the form

$$C = p \leftarrow q_1, \ldots, , q_n, not\ r_1, \ldots, not\ r_m$$

We will interpret such clause as a guarded Horn clause:

$$g(C) = p \leftarrow q_1, \ldots, , q_n : \{r_1, \ldots, r_m\}.$$

We define $g(P) = \{g(C) : C \in P\}$. Observe that when we interpret a logic program clause as a guarded Horn clause, the polarity of atoms appearing negatively in the body of the programming clause changes sign. That is, they occurred *negatively* in the body of clause and they now appear *positively* in the guard. By convention, we think of a propositional atom is a guarded atom with an empty guard.

We now introduce our guarded resolution rule as follows. It has two arguments: the first is a guarded Horn clause and the second is a guarded atom $q : \{r_1, \ldots, r_n\}$. The guarded atom q must occur in the body of the guarded Horn clause. The result of the application of the rule is a guarded Horn clause whose body is the body of the original guarded Horn clause minus the atom q. The guard of the resulting guarded Horn clause is the union of the guards of the guarded atom and the guard of the original guarded Horn clause. Formally, our guarded resolution rule has the following form:

$$\frac{p \leftarrow q_1, \ldots, , q_n : \{r_1, \ldots, r_m\} \quad q_j : \{s_1, \ldots, s_h\}}{p \leftarrow q_1, \ldots, q_{j-1}, q_{j+1}, \ldots, q_n, \{r_1, \ldots, r_m, s_1, \ldots, s_h\}}.$$

Next, we discuss the Gelfond-Lifschitz operator associated with a normal propositional program. Given a set of atoms M and a normal logic program P, we first define the Gelfond-Lifschitz reduct P_M of P. P_M is constructed according to the following two step process. First, if

$$C = p \leftarrow q_1, \ldots, , q_n, not\ r_1, \ldots, not\ r_m$$

is a clause in P and $r_j \in M$ for some $1 \leq j \leq m$, then we eliminate C. Second, for each clause C as above which remains, we replace C by $p \leftarrow$

$q_1, \ldots, , q_n$,.Clearly, P^M is a Horn program. Thus P_M possesses the least model N_M. The Gelfond-Lifschitz operator assigns to M the set of atoms N_M.

Our guarded unit resolution rule, naturally leads to the notion of a guarded resolution proof $\mathcal{P}$ of a guarded atom $p : S$ from the program P. Here S is a, possibly empty, set of atoms. That is, a guarded resolution proof of $p : S$ is a labeled tree such that every node that is not a leaf has two parents, one labeled with a guarded Horn clause, the other labeled with a guarded atom, and the label of the node is the result of executing guarded resolution on the labels of the parents. Each leaf is either a guarded Horn clause $p \leftarrow q_1, \ldots, , q_n : \{r_1, \ldots, r_m\}$ such that $p \leftarrow q_1, \ldots, , q_n, \mathit{not}\ r_1, \ldots, \mathit{not}\ r_m$ is in P or a guarded atom $q : \{r_1, \ldots, r_m\}$ such that $q \leftarrow \mathit{not}\ r_1, \ldots, \mathit{not}\ r_m$ is in P. In the special case where the tree consist of a single node, we assume that the node is labeled with $q : \{r_1, \ldots, r_m\}$ where $q \leftarrow \mathit{not}\ r_1, \ldots, \mathit{not}\ r_m$ is in P. Note that a resolution proof, guards only grow as we proceed down the tree. That is, as we resolve, the guards are summed up. For that reason, the guard of the root of the proof contains the guards of *every* label in the tree.

We say that a set of atoms M *admits* a guarded atom $p : S$, if $M \cap S = \emptyset$ and that M admits a guarded proof $\mathcal{P}$ if it admits the label of the root of $\mathcal{P}$. The following statement follows from the containment properties of guards in the guarded proof.

Lemma 2.1 *If M admits the guarded proof $\mathcal{P}$, then M admits every guarded atom occurring as a label in $\mathcal{P}$ and M is disjoint from the guard of every guarded clause in $\mathcal{P}$.*

We then have the following proposition.

Proposition 2.1 *Let P be a propositional logic program and let M be a set of atoms. Then $GL_P(M)$ consists exactly of atoms p such that there exists a set of atoms Z where the guarded atom $p : Z$ is a conclusion of a guarded derivation $\mathcal{P}$ admitted by M.*

Proof: Let $Q = P_M$ and assume that $p \in GL_P(M)$. Then by definition, $p \in T_Q^\omega$. We claim that we can prove by induction on $n \in N$ that whenever $p \in T_Q^n$, then there exists a set of atoms Z such $p : Z$ possesses a guarded resolution derivation admitted by M. When $n = 1$ then it must be the case that the $p \leftarrow$ belongs to Q. But then for some $r_1, \ldots, r_m$,

$$p \leftarrow \mathit{not}\ r_1, \ldots, \mathit{not}\ r_m$$

belongs to P and $\{r_1, \ldots, r_m\} \cap M = \emptyset$. Therefore the guarded atom $p : \{r_1, \ldots, r_m\}$ is admitted by M and it possesses a guarded resolution proof; one that consists of the root labeled by $p : \{r_1, \ldots, r_m\}$.. Now, let us assume $p \in T_Q^{n+1}$. Then there is a clause $C = p \leftarrow q_1, \ldots, q_s$ in Q such that $q_i \in T_Q^h$ for $i = 1, \ldots, s$. Thus by induction, there are sets of atoms S_i, $1 \leq i \leq n$, such that $q_i : S_i$ possesses a guarded resolution derivation from P admitted by M. As $p \leftarrow q_1, \ldots, q_n$ belongs to Q, there must exist atoms $r_1, \ldots, r_m \notin M$ such that

$$p \leftarrow q_1, \ldots, q_n, \mathit{not}\ r_1, \ldots, \mathit{not}\ r_m,$$

is a clause in P. It is easy to combine the derivations of $q_i : Z_i$, $1 \leq i \leq n$ and the guarded clause $p \leftarrow q_1, \ldots, q_n : \{r_1, \ldots, r_m\}$ to obtain a derivation of the following guarded atom:

$$p : S_1 \cup \ldots \cup S_n \cup \{r_1, \ldots, r_m\}.$$

As all the sets occurring in the guard of this guarded atom are disjoint from M, the resulting derivation is admitted by M, as desired. This shows the inclusion $\subseteq$.

Conversely, suppose $p : Z$ has a guarded derivation $\mathcal{P}$ admitted by M. By the lemma, all the guards occurring in $\mathcal{P}$ are disjoint form M. We can then prove by induction of the the height of the tree $\mathcal{P}$ that $p \in GL_P(M)$. If the height of $\mathcal{P}$ is 0, then it must be the case that

$$p \leftarrow \textit{not } r_1, \ldots, \textit{not } r_m$$

belongs to P where $S = \{r_1, \ldots, r_m\}$. But then, as $M \cap S = \emptyset$ so that the clause $p \leftarrow$ belongs to Q. Hence $p \in GL_P(M)$.

Now, for the inductive step, assume that whenever $q : S$ has a guarded derivation of height smaller or equal to n admitted by M, then $q \in GL_P(M)$. We how that the same property holds for atoms $p : U$ with the guarded derivation admitted by M of the height $n + 1$. What does such a derivation look like? First the root must be the result of a guarded resolution:

$$\frac{p \leftarrow q : Z_1 \quad q : Z_2}{p : Z_1 \cup Z_2}.$$

As $(Z_1 \cup Z_2) \cap M = \emptyset$, $Z_1 \cap M = \emptyset$ and $Z_2 \cap M = \emptyset$. Now, $q : Z_2$ has a derivation of height at most n and so by inductive assumption $q \in GL_P(M)$. As the body of guarded clauses only get smaller as we progress down the tree and the guards of guarded clauses only get bigger, there must exist a path starting at the guarded clause $p \leftarrow q : Z_1$ which consists of guarded clauses

$$\begin{array}{l} p \leftarrow q, q_1, \ldots, q_p : Z_{p+1} \\ \vdots \\ p \leftarrow q, q_1 : Z_2 \\ p \leftarrow q : Z_1 \end{array}$$

such that $Z_{p+1} \subseteq Z_p \subseteq \cdots \subseteq Z_1$ and for each i, there is a node in the tree of the form $q_i : S_i$ such that the resolution of $p \leftarrow q, q_1, \ldots, q_i : Z_{i+1}$ and $q_i : S_i$ resulted in the clause $p \leftarrow q, q_1, \ldots, q_{i-1} : Z_i$. It follows that each $q_i : S_i$ is the root of resolution proof tree of height less than or equal to n and hence is in $GL_P(M)$. Moreover, $q : Z_2$ is also the root of resolution proof tree of height less than or equal to n and hence q is in $GL_P(M)$.

Moreover, since $p \leftarrow q, q_1, \ldots, q_p : Z_{p+1}$ is leaf, there must be a clause

$$p \leftarrow q, q_1, \ldots, q_p, \textit{not } r_1, \ldots, \textit{not } r_m$$

in P where $Z_{p+1} = \{r_1, \ldots, r_m\}$. Since M admits the proof tree, it must be the case that $\{r_1, \ldots, r_m\} \cap M = \emptyset$ and, hence, $p \leftarrow q, q_1, \ldots, q_p$ is in Q. But then since $q, q_1, \ldots, q_p$ are in $GL_P(M)$, it follows that $p \in GL_P(M)$. $\Box$

Proposition 2.1 tells us that the Gelfond-Lifschitz operator GL is, essentially, a proof-theoretic construct. Here is one consequence, this time a semantic one.

Corollary 2.1 *Let P be a propositional logic program and let M be a set of atoms. Then M is a stable model of P if an only if*

1. *For every $p \in M$, there is a set of atoms S such that there is a guarded derivation of $p : S$ from $g(P)$ admitted by M and*
2. *For every $p \notin M$, there is no set of atoms S such that there is a guarded derivation of $p : S$ admitted by M.*

Given a finite set of atoms S, we write $\neg S$ for the conjunction $\bigwedge_{q \in S} \neg q$. Next, let us call S such that $p : S$ has a guarded derivation from P a *support* of p with respect to P. We can then form an *equation* for p with respect to P, $eq_P(p)$, as follows:

$$p \Leftrightarrow (\neg S_1 \vee \neg S_2 \vee \ldots)$$

where $S_1, S_2, \ldots$ is the list of all supports of p with respect to P. Next, let E_P be the propositional theory consisting of $eq_P(p)$ for all $p \in At$. We then have the following theorem resembling Clark's completion theorem, except we get it for stable models, not supported models.

Proposition 2.2 *Let P be a propositional program and let M be a set of atoms. Then M is a stable model of P if and only if $M \models E_P$.*

Proof: First, assume that M is a stable model of P. Then if $p \in M$, it follows from Corollary 2.1 that there is an S such that $p : S$ has a guarded resolution derivation admitted by M. Hence $M \cap S = \emptyset$ and $M \models \neg S$. Thus M satisfies both p and one of the disjuncts on the right-hand side of $eq_P(p)$. Hence $M \models eq_P(p)$. Next assume that $p \notin M$. Then there is no Z, such that $p : Z$ has a guarded derivation admitted by M. It follows that either $eq_P(p)$ equals $\neg p$ or M satisfies both negation of p and of the negation of every disjunction on the right-hand-side of $eq_P(p)$. Thus again $M \models eq_P(p)$. This shows $\Rightarrow$.

For the other direction, suppose that $M \models eq_P(p)$. Then if $p \in M$, either $eq_P(p) = p$ or $eq_P(p) = p \Leftrightarrow (\neg S_1 \vee \neg S_2 \vee \ldots)$. In the form case, there is nothing to do. In the later case, there must be some S_i such that $M \models \neg S_i$. But by definition, $P : S_i$ is the root of some derivation $\mathcal{P}$ and since every guard in the tree is contained in S_i, then M admits $\mathcal{P}$. But then we have shown that $p \in GL_P(M)$. Thus $M \subseteq GL_P(M)$.

On the other hand, if $p \notin M$, then either $eq_P(p) = p$ or $eq_P(p) = p \Leftrightarrow (\neg S_1 \vee \neg S_2 \vee \ldots)$. In the former case, there is nothing to do and in the latter case, it must be that M does not satisfy any $\neg S_i$. This means that there is no derivation of the form $p : S$ such that M admits p and hence $p \notin GL_M(P)$. This implies $GL_P(M) \subseteq M$ and hence, $GL_p(M) = M$. Thus M is a stable model of P. □

I am not sure why this comment is here. It needs further explanation.

If we look carefully at what happens when we compute supports for atoms, we see that we can , essentially, unfold the atoms to conjunctions of negated atoms (see [BD99].)

3 Some applications

In this section we will use the results of Section 2 to derive the result of Erdem and Lifschitz [EL03]. This result generalizes a theorem by Fages [Fa94]. That last result is useful as a preprocessing for computation of stable models of programs. We will also prove a result of Dung [DK89] on stable models of purely negative programs.

We say that a set of atoms M *has levels* with respect to a program P if

1. M is a model of P, and
2. There is a function $rk : M \rightarrow ord$ such that, for every $p \in M$, there is a clause C such that
   (a) $p = head(C)$
   (b) $M \models body(C)$
   (c) For all $q \in posBody(C)$, $rk(q) < rk(p)$.

Clearly, the least model of a Horn program has levels; namely the function assigning to an atom $p \in M$ the least integer n such that $p \in T_P^n(\emptyset)$ is the desired rank function satisfying condition of (2).

We now prove the following proposition.

Proposition 3.1 *Let P be a propositional logic program and $M \subseteq At$. Then M is a stable model of P if and only if M has levels with respect to P.*

Clearly, when M is a stable model of P, then M has levels with respect to P. Namely, the rank function whose existence is postulated in (2)(ii) is the rank function inherited from the Horn program P_M.

Converse implication can be proved in a variety of ways. Our proof, in the spirit of proof-theoretic position adopted in this paper uses guarded resolution.

Our goal is to prove that $M = GL_P(M)$. First, let us observe that since M satisfies P, $GL_P(M) \subseteq M$. Thus, all we need to show is that when M possesses levels with respect to P, then $M \subseteq GL_P(M)$. In other words, following Corollary 2.1, given any $p \in M$, we need to find Z so that $p : Z$ possesses a guarded derivation out of P that is admitted by M.

We construct such set Z and the derivation using the rank function rk whose existence is assumed in (2). So, let $S = \{rk(p) : p \in M\}$, i.e. S be the range of function S. We proceed by transfinite induction on the elements of S. Let p be an atom in M such that $rk(p)$ is the least element of S. Now, let C be a clause in P, such that $M \models body(C)$, $p = head(C)$ and for all $q \in posBody(C)$, $rk(q) < rk(p)$. But p has a least rank in C and the putative q's in the positive part of the body of C would be in C and have a smaller rank. Thus there is no such q, and therefore the clause C has the following form:

$$p \leftarrow not\ r_1, \ldots not\ r_m.$$

As $M \models body(C)$, $r_1, \ldots, r_m \notin M$. But then $p : \{r_1, \ldots, r_m\}$ is a guarded atom admitted by M and so $p : \{r_1, \ldots, r_m\}$ has a guarded derivation from P.

Now, let us assume that whenever $\beta \in S$, and $\beta < \alpha$ and $rk(q) = \beta$ then there is a guarded derivation of q admitted by M. Let us assume that $p \in M$, $rk(p) = \alpha$. By our assumption, there is a clause C

$$p \leftarrow q_1, \ldots, q_n, not\ r_1, \ldots not\ r_m$$

with $M \models body(C)$ and $rk(q_1) < rk(p), \ldots, rk(q_n) < rk(p)$. By inductive assumption, for every q_i, $1 \leq i \leq n$, there is a finite set of atoms Z_i such that there is guarded derivation $\mathcal{D}_i$ of $q_i : Z_i$, with $\mathcal{D}_i$ admitted by M. In particular $Z_i \cap M = \emptyset$. We now transform our clause C into

$$p \leftarrow q_1, \ldots, q_n : \{r_1, \ldots, r_m\}$$

and by successive applications of guarded resolution rule (n applications) derive the following guarded atom:

$$p : Z$$

where $Z = Z_1 \cup \ldots \cup Z_n \cup \{r_1, \ldots, r_n\}$. Since all Z_is are disjoint from M and also $M \cap \{r_1, \ldots, r_m\} = \emptyset$, $M \cap Z = \emptyset$. Thus the constructed derivation is admitted by M, as desired. This completes the inductive argument and thus the proof of the Proposition. □

We observe that, in fact, it is sufficient to limit the functions rk to those that take values in N, the set of natural numbers.

We get, as promised, several corollaries. One of these is the result of Erdem and Lifschitz [EL03]. Following [EL03], we say that a program P is *tight* on a set of atoms M if there is a rank function rk defined on M such that whenever C is a clause in P and $head(C) \in M$, then for all $q \in posBody(C)$, $rk(q) < rk(head(C))$. Here is the result of Erdem and Lifschitz.

Corollary 3.1 *If M is tight on P and M is a supported model of P then M is a stable model of P.*

Indeed, tightness on M requires that for any $p \in M$, there is a support for p and that *all* clauses C that provide the support for the presence of p in M have the property that the atoms in the positive part of the body of C have smaller rank. In Proposition 3.1, we showed that it is enough to have just one such clause. Since tightness on M implies existence of such a supported clause, the corollary follows. □

Since all stable models are supported [GL88], one can express Erdem-Lifschitz Theorem in a more succinct way.

Corollary 3.2 ((Erdem-Lifschitz) *If M is tight on P, then the classes of supported and stable models of P coincide.*

Fages Theorem [Fa94] is a weaker version of Corollary 3.1 (but with assumptions that are easier to test). Specifically, we say that a program P is *tight* if there is a rank function rk such that for every clause C of P, the ranks of atoms occurring in the positive part of the body of C are smaller than the rank of the head of C. Clearly, if P is tight, then P is tight on any of its model - the same rank function witnesses to that. Thus one gets the following corollary.

Corollary 3.3 ((Fages) *If P is tight then the classes of supported and stable models of P coincide.*

Tightness is a syntactic property that can be checked in polynomial time by inspection of the positive call-graph of P. This is not the case for the stronger assumptions of Proposition 3.1 and Corollary 3.2.

Let $Stab(P)$ be the set of all stable models of P. We say that programs P, P' are *equivalent* if $Stab(P) = Stab(P')$. This notion and its strengthenings were studied by ASP community [LPV01], [Tr06]. We have the following fact.

Lemma 3.1 *If P, P' prove the same guarded atoms, then P and P' are equivalent.*

As a corollary we get the following result due to Dung [DK89]

Corollary 3.4 ((Dung)) *For every program P, there is purely negative program P' such that P, P' are equivalent.*

The program P' is quite easy to construct. That is, for each atom p, if

$$eq_P(p) = p \Leftrightarrow (\neg S_1 \vee \neg S_2 \vee \ldots),$$

then we add to P', all clauses of the form

$$p \leftarrow not\ r_{i,1}, \ldots, not\ r_{i,m_i}$$

where $S_i = \{r_{i,1}, \ldots, r_{i,m_i}\}$. If $eq_P(p) = p$, then we add $p \leftarrow$ to P'. Finally, if $eq_P(p) = \neg p$, then we add nothing to P'. It is then easy to see that $E_P = E_{P'}$ and hence P and P' are equivalent. □

4 Conclusions and Further Work

We showed that guarded resolution, a proof system that is a nonmonotonic version of resolution, is adequate for description of the Gelfond-Lifschitz operator GL_P and its fixpoints. That is, we can characterize stable models of logic programs in terms of guarded resolution.

There are several natural questions concerning extensions of guarded resolution in the context of Answer Set Programming. For example:
Is there an analogous proof system for the disjunctive version of logic programming?
or
Are there analogous proof systems for logic programming with constraints?
We believe that availability of such proof systems could help with finding a variety of results on the complexity of reasoning in nonmonotonic logics. An interesting case is that of Default Logic. While it is, clearly, a proof-theoretic system, its reasoning technique does not appear to be purely proof-theoretical. In particular, there is no known general *schematic* reasoning system (of the kind considered here) for Default Logic.

References

[ABW88] Apt, K., Blair, H.A., and Walker, A. Towards a theory of Declarative Knowledge. In *Foundations of Deductive Databases and Logic Programming*, J. Minker, Ed. Morgan Kaufmann, 89–148, 1988.

[Ba03] Baral, C., *Knowledge Representation, Reasoning and Declarative Problem Solving*, Cambridge University Press, 2003.

[BD99] Brass, S. and Dix, J. Semantics of (disjunctive) Logic Programs Based on Partial Evaluation. *Journal of Logic Programming* 40:1–46, 1999.

[DK89] Dung, P.M. and Kanchansut, K. A fixpoint approach to declarative semantics of logic programs. *Proceedings of North American Conference on Logic Programming* pages 604–625, 1989.

[EL03] Erdem, E. and Lifschitz, V. Tight Logic Programs. *Theory and Practice of Logic Programming* 3:499–518, 2003.

[Fa94] Fages, F. Consistency of Clark's completion and existence of stable models. *Journal of Methods of Logic in Computer Science*, 1:51–60, 1994.

[GL02] Gelfond, M. and Leone, N. Logic Programming and Knowledge Representation – A-Prolog perspective. *Artificial Intelligence* 138:3–38, 2002.

[GL88] Gelfond, M. and Lifschitz, V. The stable model semantics for logic programming. In *Proceedings of the International Joint Conference and Symposium on Logic Programming*. MIT Press, 1070–1080, 1988.

[KK71] Kowalski, R. and Kuehner, D. Linear Resolution with Selection Rule. *Artificial Intelligence* 2:227–260, 1971.

[Li08] Lifschitz, V. Twelve definitions of a stable model. in: *Proceedings of International Conference on Logic Programming*, pages 37–51, 2008.

[LPV01] Lifschitz, V., Pearce, D. and Valverde, A. Strongly equivalent logic programs. *ACM Transactions on computational logic* 2:526–541, 2001.

[MT99] Marek, V.W., and Truszczyński, M. Stable Models and an Alternative Logic Programming Paradigm. *The Logic Programming Paradigm*, pp. 375–398. Series Artificial Intelligence, Springer-Verlag, 1999.

[Nie99] Niemelä, I. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence 25,* 3,4, 241–273, 1999

[Tr06] Truszczynski, M. Strong and Uniform Equivalence of Nonmonotonic Theories An algebraic Approach. *Proceedings of the Conference on Principles of Knowledge Representation and Reasoning*, pages 389–399, 2006.