Adding Negation

Jef Wijsen

April 16, 2008

1 Nonrecursive Datalog with Negation

If L is an atom, then $\neg L$ is a *negative literal* and L a *positive literal*. A *nr-datalog*$^\neg$ *rule* is defined as a conjunctive rule where the rule body can contain both positive and negative literals. The rule is *range restricted* if each variable occurring in the rule occurs in a positive literal in the rule body. For example,

$$\mathsf{Answer1}(x) \leftarrow \mathsf{Emp}(x,y,z), \neg\mathsf{Loc}(z, \text{“Charleroi”})$$

The *answer* to a range-restricted nr-datalog$^\neg$ rule $q : H \leftarrow B$ on a database I, denoted $q(I)$, is defined by:

$$q(I) = \{\theta(H) \mid \theta \text{ is a substitution such that} \left\{ \begin{array}{l} \text{for each } L \in B,\, \theta(L) \in I \\ \text{for each } \neg L \in B,\, \theta(L) \notin I \end{array} \right. \} .$$

A *nr-datalog*$^\neg$ *program* is a sequence of nr-datalog$^\neg$ rules:

$$\begin{array}{rcl} H_1 & \leftarrow & B_1 \\ H_2 & \leftarrow & B_2 \\ & \vdots & \\ H_n & \leftarrow & B_n \end{array}$$

such that the relation name that occurs in H_i does not occur in $B_1, B_2, \ldots, B_i$. The program is evaluated on input I by evaluating each rule in the given order and forming unions whenever two rules have the same relation name in their heads. For example,

$$\begin{array}{rcl} \mathsf{ChEmp}(x) & \leftarrow & \mathsf{Emp}(x,y,z), \mathsf{Loc}(z, \text{“Charleroi”}) \\ \mathsf{Answer1}(x) & \leftarrow & \mathsf{Emp}(x,y,z), \neg\mathsf{ChEmp}(x) \end{array}$$

What is the semantic difference between the two example queries?

2 Recursive Datalog with Negation

A *datalog*$^\neg$ *program* is a datalog program where negative literals can appear in rule bodies. The immediate consequence operator T_P can be naturally extended to a datalog$^\neg$ program P. However, the minimal fixpoints containing I may not be unique.

$$\begin{array}{rcl} \mathsf{GreenPath}(x,y) & \leftarrow & \mathsf{Green}(x,y) \\ \mathsf{GreenPath}(x,y) & \leftarrow & \mathsf{GreenPath}(x,z), \mathsf{GreenPath}(z,y) \\ \mathsf{RedMonopoly}(x,y) & \leftarrow & \mathsf{Red}(x,y), \neg\mathsf{GreenPath}(x,y) \end{array}$$

Let

$$\begin{aligned} I &= \{\mathsf{Green}(a,b), \mathsf{Red}(a,b), \mathsf{Red}(b,c)\} \\ J_1 &= I \cup \{\mathsf{GreenPath}(a,b), \mathsf{RedMonopoly}(b,c)\} \\ J_2 &= I \cup \{\mathsf{Green}(b,c), \mathsf{GreenPath}(a,b), \mathsf{GreenPath}(b,c), \mathsf{GreenPath}(a,c)\} \end{aligned}$$

Both J_1 and J_2 are minimal fixpoints. Note incidentally that the following database $J_3 \subsetneq J_2$ is not a fixpoint of T_P:

$$\begin{aligned} J_3 &= I \cup \{\mathsf{GreenPath}(a,b), \mathsf{GreenPath}(b,c), \mathsf{GreenPath}(a,c)\} \\ T_P(J_3) &= I \cup \{\mathsf{GreenPath}(a,b), \mathsf{GreenPath}(a,c)\} \\ T_P^2(J_3) &= I \cup \{\mathsf{GreenPath}(a,b), \mathsf{RedMonopoly}(b,c)\} \\ T_P^3(J_3) &= T_P^2(J_3) = J_1 \end{aligned}$$

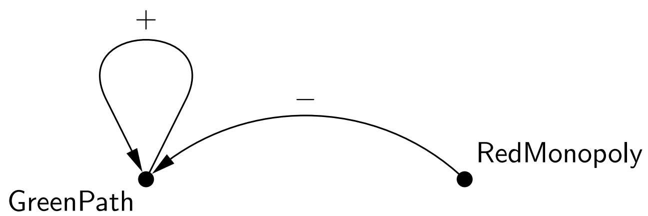
3 Stratified Semantics for Datalog$^\neg$

The *dependency graph* of a datalog$^\neg$ program P is the labeled graph whose nodes are the relation names of $idb(P)$. Its edges are the following:

- If $R(\vec{x}) \leftarrow \ldots S(\vec{y}) \ldots$ is a rule of P with $S \in idb(P)$, then there is an edge with label $+$ from R to S.
- If $R(\vec{x}) \leftarrow \ldots \neg S(\vec{y}) \ldots$ is a rule of P with $S \in idb(P)$, then there is an edge with label $-$ from R to S.

A program is *stratified* if its dependency graph has no cycle containing a negative edge. Given a stratified program P, the *stratum* of an intensional relation name R is the largest number of negative edges in a path from R, in the dependency graph.

For example,



The stratum of GreenPath is 0; the stratum of RedMonopoly is 1.

Under the stratified semantics, the strata $0, 1, \ldots$ are evaluated in order. Notice that for each rule

$$R(\vec{x}) \leftarrow \ldots \neg S(\vec{y}) \ldots \quad ,$$

where $R, S \in idb(P)$, the stratum of R is greater than the stratum of S. When evaluating this rule, all rules for S have already been evaluated, so S can be treated as an extensional relation name.

It can be shown that, given database I and stratified program P, the stratified semantics gives us a minimal fixpoint of T_P containing I. Intuitively, the stratified semantics is a natural choice from among several possible fixpoints.

4 A Note on Model-theoretic Semantics

We can associate with each nr-datalog$^\neg$ program a first-order logic theory, for example:

$$\forall x \forall y (\mathsf{Green}(x,y) \Rightarrow \mathsf{GreenPath}(x,y))$$
$$\forall x \forall y \forall z ((\mathsf{GreenPath}(x,z) \wedge \mathsf{GreenPath}(z,y)) \Rightarrow \mathsf{GreenPath}(x,y))$$
$$\forall x \forall y ((\mathsf{Red}(x,y) \wedge \neg \mathsf{GreenPath}(x,y)) \Rightarrow \mathsf{RedMonopoly}(x,y))$$

Given a database I, it seems natural to define the semantics of a program in terms of the minimal models of its first-order theory that contain I. (A model is any database that satisfies all the theory's sentences.) However, for programs with negation, there may be multiple minimal models containing a given database I, i.e. there may be no *unique* minimal model. For example, J_1 and J_3 are two minimal models containing $\{\mathsf{Green}(a,b), \mathsf{Red}(a,b), \mathsf{Red}(b,c)\}$. Note also that J_2 is a model, but not a minimal model (since $J_3 \subsetneq J_2$).

On the other hand, for programs without negation, the minimal model containing a given database is unique and coincides with the minimal fixpoint of T_P.

5 Exercises

1. Consider the following datalog$^\neg$ program P:

$$\begin{array}{rcl} \mathsf{Male}(x) & \leftarrow & \mathsf{Person}(x), \neg\mathsf{Female}(x) \\ \mathsf{Female}(x) & \leftarrow & \mathsf{Person}(x), \neg\mathsf{Male}(x) \end{array}$$

   Show that the immediate consequence operator T_P is not monotonic.

2. Write a stratified datalog program$^\neg$ to answer the following query:

   > Find pairs of cities (a,b) such that b can be reached from a by some combination of red or green edges, but not by red or green edges alone.

3. From [1]. Consider the following stratified datalog$^\neg$ program P:

$$\begin{array}{rcl} P(x,y) & \leftarrow & Q(x,y), \neg R(x) \\ R(x) & \leftarrow & S(x,y), \neg T(y) \\ R(x) & \leftarrow & S(x,y), R(y) \end{array}$$

   Let $I = \{S(a,b), S(b,c), S(c,a), T(a), T(b), T(c), Q(a,b), Q(b,c), Q(c,d), Q(d,e)\}$.

   (a) Find the minimal fixpoint given by the stratified semantics.

   (b) Find another minimal fixpoint.

References

[1] J. D. Ullman. *Principles of Database and Knowledge-Base Systems – Volume II.* Computer Science Press, Rockville, MD, 1989.