NP Datalog

Language Definition

NP Datalog is a language designed to express, in a simple and intuitive way, NP search and optimizations. Basically, NP Datalog extends positive Datalog with controlled forms of negation. These extensions are, stratified negation, a restricted form of exclusive disjunction (used to define relation partitions) and constraints.

Syntax

An NP Datalog program consists of three distinct sets of rules:

1. partition and subset rules defining guess (IDB) predicates,
2. standard stratified datalog rules defining standard (IDB) predicates, and
3. constraint rules.

where every guess predicate is defined by a unique subset or partition rule.

A partition rule is a disjunctive rule of the form:

\[ p_1(X) \oplus \cdots \oplus p_k(X) \leftarrow \text{Body}(X, Y) \quad (1) \]

or of the form

\[ p_0(X, c_1) \oplus \cdots \oplus p_0(X, c_k) \leftarrow \text{Body}(X, Y) \quad (2) \]

where \( p_0, p_1, \ldots, p_k \) are distinct IDB predicates not defined elsewhere in the program and \( c_1, \ldots, c_k \) are distinct constants and \( \text{Body}(X, Y) \) denotes a conjunction of literals and \( X \) and \( Y \) are vectors of range restricted variables.

The intuitive meaning of these rules is that the projection of the relation defined by \( \text{Body}(X, Y) \) on \( X \) is partitioned nondeterministically into \( k \) relations or \( k \) distinct sets of the same relation. Clearly, every rule of the form (2) can be rewritten into a rule of the form (1) and vice versa.

A generalized partition rule is a (generalized) disjunctive rule of the form:

\[ \bigoplus_L p(X, L) \leftarrow \text{Body}(X, Y), d(L) \quad (3) \]

where \( p \) is an IDB predicate not defined elsewhere and \( d \) is a database domain predicate specifying the domain of the variable \( L \).

The intuitive meaning of such a rule is that the projection of the relation defined by \( \text{Body}(X, Y) \) on \( X \) is partitioned into a number of subsets equal to the cardinality of the relation \( d \).

A subset rule is of the form

\[ s(X) \subseteq \text{Body}(X, Y) \quad (4) \]
where $s$ is an IDB predicate not defined elsewhere in the program. Observe that a subset rule of the form above corresponds to the generalized partition rule with $d = \{0, 1\}$. On the other hand, every generalized partition rule can be rewritten into a subset rule and constraints.

In the previous rules, $\text{Body}(X, Y)$ is a conjunction of literals not depending on predicates defined by partition or subset rules.

$\mathcal{NP} \text{ Datalog}$ allows us to also use a finite subset of the integer domain and the standard built-in arithmetic operators. More specifically, reasoning and computing over a finite set of integer ranges is possible with the unary predicate $\text{integer}$, which consists of the facts $\text{integer}(x)$, with $\text{MinInt} \leq x \leq \text{MaxInt}$, and the standard arithmetic operators defined over the integer domain. Moreover, the language allows arithmetic expressions which involve variables taking integer values to appear as operands of comparison operators.

A constraint (rule) is of the form:

$$\leftarrow \text{Body}(X) \quad (5)$$

A ground constraint rule is satisfied w.r.t. an interpretation $I$ if the body of the rule is false in $I$. Constraints can be also written under the form $A_1 \lor \ldots \lor A_k \leftarrow B_1, \ldots, B_m$ (or $B_1, \ldots, B_m \Rightarrow A_1 \lor \ldots \lor A_k$) to denote a constraint of the form $\leftarrow B_1, \ldots, B_m, \neg A_1, \ldots, \neg A_k$ (i.e. negative literals are moved from the body to the head). Here the symbol $\lor$ denotes inclusive disjunction and is different from $\oplus$ as the latter denotes exclusive disjunction. It should be recalled that inclusive disjunction allows more than one atom to be true while exclusive disjunction allows only one atom to be true.

According to the definition above, the set of IDB predicates of an $\mathcal{NP} \text{ Datalog}$ program can be partitioned into two distinct subsets depending on the rules used to define them. Observe that predicates defined by partition or subset rules are not recursive as the body of these rules cannot contain guess predicates or predicates depending on guess predicates.

An $\mathcal{NP} \text{ Datalog}$ search query is a pair $Q = \langle \mathcal{P}, g(t) \rangle$, where $\mathcal{P}$ is an $\mathcal{NP} \text{ Datalog}$ program and $g(t)$ is an IDB atom denoting the output relation. An optimization query is a pair $\langle \mathcal{P}, \text{opt}|g(t)\rangle$ where $\text{opt} \in \{\text{max}, \text{min}\}$.

For the sake of simplicity, the current implementation considers only optimization queries computing the maximum or minimum cardinality of the output relation, although any polynomial function can be used. It is worth noting that $\mathcal{NP} \text{ Datalog}$ has the same expressive power as $\text{DATALOG}^-$ (Datalog with negation).

**Semantics.**

The declarative semantics of a $\mathcal{NP} \text{ Datalog}$ query $\langle \mathcal{P}, g(t) \rangle$ is given in terms of an ‘equivalent’ $\text{DATALOG}^-$ query and stable model semantics. Specifically, given a $\mathcal{NP} \text{ Datalog}$ program $\mathcal{P}$, $\text{st}(\mathcal{P})$ denotes the standard $\text{DATALOG}^-$ program derived from $\mathcal{P}$ as follows:

1. Every standard rule in $\mathcal{P}$ belongs to $\text{st}(\mathcal{P})$,

2. Every partition rule $r \in \mathcal{P}$ of the form (1) is translated into $k$ rules of the form:

$$p_j(X) \leftarrow \text{Body}(X, Y), \neg p_1(X), \ldots, \neg p_{j-1}(X), \neg p_{j+1}(X), \ldots, \neg p_k(X_k)$$

with $j \in [1..k]$ (recall that each $p_j$ is not defined by some other rule).
3. Every generalized partition rule of the form (3) is translated into the two rules:

\[
\begin{align*}
  p(X, L) &\leftarrow Body(X, Y), d(L), \neg \text{diff}_p(X, L) \\
  \text{diff}_p(X, L) &\leftarrow Body(X, Y), d(L), p(X, L'), L' \neq L
\end{align*}
\]

where \text{diff}_p is a new predicate symbol and \( L' \) is a new variable. Here \text{diff}_p is used to avoid inferring two ground atoms \( p(x, l_1) \) and \( p(x, l_2) \) with \( l_1 \neq l_2 \) (observe that \( p \) is not defined by any other rule).

4. Every constraint rule of the form (5), is translated into a rule of the form:

\[
\begin{align*}
  c &\leftarrow Body(X), \neg c
\end{align*}
\]

where \( c \) is a new predicate symbol not appearing elsewhere.

For any \( \mathcal{NP} \) Datalog search query \( Q = \langle \mathcal{P}, g(t) \rangle \) (resp. optimization query \( OQ = \langle \mathcal{P}, \text{opt}[g(t)] \rangle \)), \( st(Q) = \langle st(\mathcal{P}), g(t) \rangle \) (resp. \( st(OQ) = \langle st(\mathcal{P}), \text{opt}[g(t)] \rangle \)) denotes the corresponding \text{DATALOG\textsuperscript{\neg}} (resp. optimization) query.

Definition 1 Given an \( \mathcal{NP} \) Datalog query \( Q \) and a database \( D \), the (nondeterministic) answer to the query \( Q \) over \( D \) is obtained by applying the \text{DATALOG\textsuperscript{\neg}} query \( st(Q) \) to \( D \), i.e. \( Q(D) = st(Q)(D) \).