Logic Programming

Georg Moser

Institute of Computer Science @ UIBK

Summer 2015

Summary of Last Lecture

Observation

the basic operations of relational algebras, namely:

1. union
2. difference
3. cartesian product
4. projection
5. selection
6. intersection

can easily be expressed within logic programming

Definition

- a type is a (possible infinite) set of terms
- types are conveniently defined by unary relations

Outline of the Lecture

Logic Programs

introduction, basic constructs, database and recursive programming, theory of logic programs

The Prolog Language

programming in pure Prolog, arithmetic, structure inspection, meta-logical predicates, cuts, extra-logical predicates, how to program efficiently

Advanced Prolog Programming Techniques

nondeterministic programming, incomplete data structures, definite clause grammars, meta-programming, constraint logic programming

Theory of Logic Programs

Definitions

- goal clause

  $$\leftarrow B_1, \dots, B_n$$

  consists of sequence $B_1, \dots, B_n$ of goals
- empty goal clause $\leftarrow$ is denoted by $\square$
- resolvent of goal clause $\leftarrow B_1, \dots, B_i, \dots, B_m$ and rule $A \leftarrow A_1, \dots, A_n$ is goal clause

  $$\leftarrow B_1\theta, \dots, B_{i-1}\theta, A_1\theta, \dots, A_n\theta, B_{i+1}\theta \dots, B_m\theta$$

  provided B_i (selected goal) and A unify with mgu θ

Selective Linear Definite Clause Resolution

Definitions

- SLD-derivation of logic program P and goal clause G consists of
  1. maximal sequence $G_0, G_1, G_2, \ldots$ of goal clauses
  2. sequence $C_0, C_1, C_2, \ldots$ of variants of rules in P
  3. sequence $\theta_0, \theta_1, \theta_2, \ldots$ of substitutions

  such that
  - $G_0 = G$
  - G_{i+1} is resolvent of G_i and C_i with mgu θ_i
  - C_i has no variables in common with $G, C_0, \ldots, C_{i-1}$
- SLD-refutation is finite SLD-derivation ending in $\square$
- computed answer substitution of SLD-refutation of P and G with substitutions $\theta_0, \theta_1, \ldots, \theta_m$ is restriction of $\theta_0\theta_1\cdots\theta_m$ to variables in G

Example

```
plus(0,X,X).
plus(s(X),Y,s(Z)) ← plus(X,Y,Z).
times(0,X,0).
times(s(X),Y,Z) ← times(X,Y,U), plus(U,Y,Z).

← times(X,X,Y)
```

SLD-refutation

G_0: ← times(X,X,Y)
- C_0: times(s(X_0),Y_0,Z_0) ← times(X_0,Y_0,U_0), plus(U_0,Y_0,Z_0)
- θ_0: X ↦ s(X_0), Y_0 ↦ s(X_0), Z_0 ↦ Y

G_1: ← times(X_0,s(X_0),U_0), plus(U_0,s(X_0),Y)
- C_1: times(0,X_1,0).
- θ_1: X_0 ↦ 0, X_1 ↦ s(0), U_0 ↦ 0

G_2: ← plus(0,s(0),Y)
- C_2: plus(0,X_2,X_2).
- θ_2: X_2 ↦ s(0), Y ↦ s(0)

G_3: $\square$ computed answer substitution: X ↦ s(0), Y ↦ s(0)

Remark

selected goal in goal clause is determined by selection function

Theorem

∀ logic programs P and goal clause G
∀ computed answer substitutions θ
∀ selection functions $\mathcal{S}$
∃ computed answer substitution θ' using $\mathcal{S}$
such that θ' is at least as general as θ (with respect to variables in G)

One Choice

goal in sequence of goals – "∀ selection functions $\mathcal{S}$"

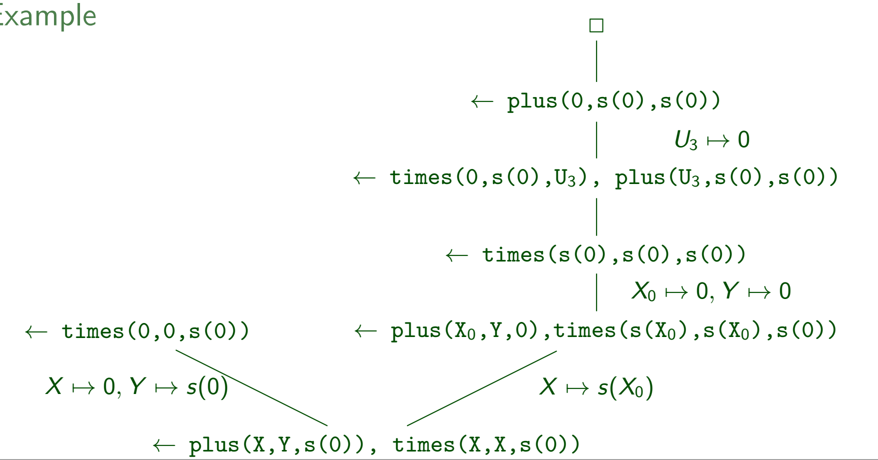
2 rule in logic program

substitution – avoid choice by always taking mgu

Definition

a SLD tree captures all SLD derivations wrt a given selection function

Example



Semantics

Definitions

- the Herbrand universe for a program P is the set of all closed terms built from constants and function symbols appearing in the program
- the Herbrand base is the set of all ground goals formed from predicates in P and terms in the Herbrand universe
- an interpretation is a subset of the Herbrand base
- an interpretation I is a model if it is closed under rules:

$$\forall A \leftarrow B_1, \dots, B_n \quad A \in I, \text{ if } B_1, \dots, B_n \in I$$

- the minimal model of P is the intersection of all models; the minimal model is unique

Declarative, Operational, and Denotational Semantics

Definition

- the declarative semantics of P (aka its meaning) is the minimal model of P
- we also say that the meaning of a logic program P, is the set of (ground unit) goals deducible from P

Definitions

the operational semantics describes the meaning of a program procedurally

Definition

the denotational semantics assign meanings to programs based on associating with the program a function over the domain computed by the program

Correctness and Completeness

Definition

the intended meaning of a logic program is a set of ground facts

Definition

a program P is called

- correct with respect to the intended meaning M, if the meaning of P is a subset of M
- complete if the intended meaning M is a subset of the meaning of P

Example

```
natural_number(0).
natural_number(s(X)) ← natural_number(X).
```

Lemma

the program is complete wrt the set of facts

$$M := \{\texttt{natural\_number}(s^i(0)) \mid i \geqslant 0\}$$

Proof of Completeness.

1. let N be a natural number
2. we show that $\texttt{natural\_number}(s^N(0))$ is deducible by given a explicit SLD tree
3. case distinction on $N = 0$ and $N > 0$

■

Lemma

the program is correct wrt the set of facts

$$M := \{\texttt{natural\_number}(s^i(0)) \mid i \geqslant 0\}$$

Proof of Correctness.

1. suppose $\texttt{natural\_number}(s^m(0))$ is deducible in n deductions
2. we use induction on n
3. $n = 0$: then $\texttt{natural\_number}(s^m(0))$ implies $m = 0$
4. $n > 0$: the goal must be of form $\texttt{natural\_number}(s(t))$
5. thus $\texttt{natural\_number}(t)$ is deducible with $n - 1$ deductions
6. $t = s^{m'}(0)$ for some $m' \in \mathbb{N}$
7. $\texttt{natural\_number}(s^{m'1}(0)) \in M$ and $m = m' + 1$

∎

Example

is the program is correct wrt the following set?

$$M := \{\texttt{natural\_number}(s^i(0)) \mid 0 \leqslant i \leqslant K\}$$

Attempted Proof

1. suppose $\texttt{natural\_number}(s^m(0))$ is deducible in n deductions
2. we use induction on n
3. $n = 0$: then $\texttt{natural\_number}(s^m(0))$ implies $m = 0$
4. $n > 0$: the goal must be of form $\texttt{natural\_number}(s(t))$
5. thus $\texttt{natural\_number}(t)$ is deducible with $n - 1$ deductions
6. $t = s^{m'}(0)$ for some $0 \leqslant m' \leqslant K$ and $m = m' + 1$
7. $\texttt{natural\_number}(s^m(0)) \in M$ iff $m \leqslant K$
8. what happens for $m > K$?

Example

```
natural_number(0).
natural_number(s(X)) ← natural_number(X).

plus(0,X,X) ← natural_number(X).
plus(s(X),Y,s(Z)) ← plus(X,Y,Z).
```

Lemma

the program is correct and complete wrt to the definition of addition

Proof Sketch.

1. completeness: suppose $X + Y = Z$; then we give an SLD tree of $\texttt{plus}(X, Y, Z)$
2. correctness: suppose $\texttt{plus}(X, Y, Z)$ is deducible; then we prove by induction on the length of this deduction that $X + Y = Z$

∎

Definitions

- a proof tree for a program P and a goal G is a tree, whose nodes are goals and whose edges represent reduction of goals; the root is the query G
- a proof tree for a conjunction of goals $G_1, \ldots, G_n$ is the set of all proof trees for G_i

Definitions

- a search tree is the same as an SLD tree
- in a search tree $\square$ is called a success node
- leafs in the search tree $\neq \square$ are called failure node

Remark

a proof tree is a different representation of one successful solution represented by a search tree