UCS1524 – Logic Programming

Semantics of Logic Program

Session Meta Data

Author	**Dr. D. Thenmozhi**
Reviewer	
Version Number	1.2
Release Date	27 August 2022

Session Objectives

- Understanding semantics of logic programming
- Learn the concept of procedural semantics and model-theoretic semantics

Session Outcomes

- At the end of this session, participants will be able to
  - explain the concept of semantics namely procedural semantics and model-theoretic semantics of logic program.

Agenda

- Procedural interpretation
- Procedural semantics
- Model theoretic semantics
  - Declarative semantics

Procedural interpretation

- The procedural interpretation of Horn clause programs is given by the presentation of an abstract *interpreter for such programs.*
- *A configuration of* this interpreter is any pair *(G, sub) where G is a goal clause and sub is a* substitution.
- Let F be a logic program (set of definite Horn clauses).
- The *transition relation for configurations is then defined as follows.*

$$(G_1, sub_1) \vdash_{F} (G_2, sub_2)$$

- if and only if G1 has the form

$$G_1 = \{\neg A_1, \neg A_2, \ldots, \neg A_k\} \qquad (k \geq 1)$$

Procedural interpretation

- The program clause in F (rename variables so that G, and C do not have a variable in common) $C = \{B, \neg C_1, \neg C_2, \ldots, \neg C_n\} \qquad (n \geq 0)$

- Let a most general unifier be the substitution s. Then G2 has the form

$$G_2 = \{\neg A_1, \ldots, \neg A_{i-1}, \neg C_1, \ldots, \neg C_n, \neg A_{i+1}, \ldots, \neg A_k\}s$$

- and *sub2 has the form* $sub_2 = sub_1 s.$

- *A computation of F on input* $G = \{\neg A_1, \ldots, \neg A_k\}$ *is a (finite or infinite)* sequence of the form

$$(G, [\,]) \vdash_{\mathrm{F}} (G_1, sub_1) \vdash_{\mathrm{F}} (G_2, sub_2) \vdash_{\mathrm{F}} \cdots$$

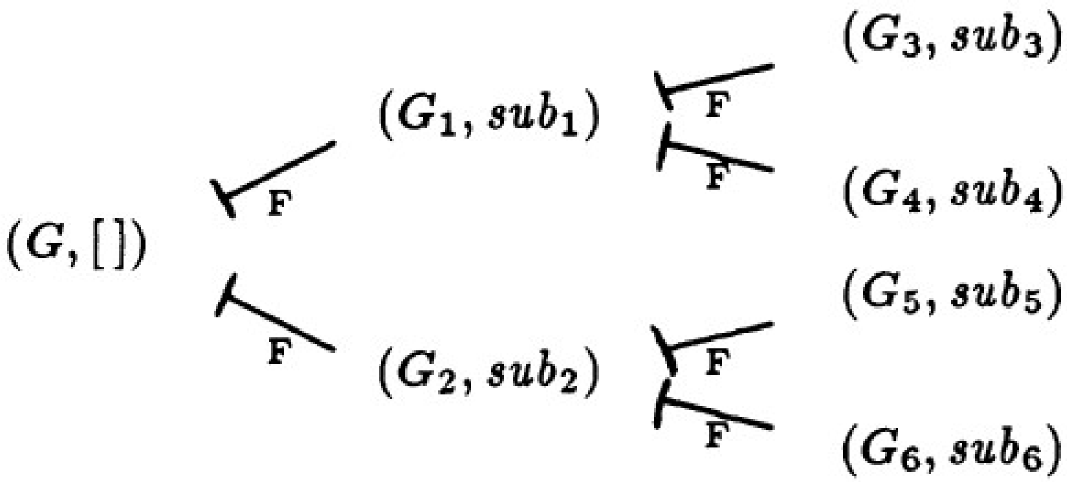
- If the sequence is finite, and the last configuration of it has the form ([], sub), then this computation is called successful, and in this case the formula

$$(A_1 \wedge \cdots \wedge A_k)sub$$

is called the result of the computation.

Procedural interpretation

- The possible computations from a given input G can be represented as a tree.



Example

- Consider the following logic program

$$
\begin{aligned}
F \quad = \quad & \{\{P(x,z), \neg Q(x,y), \neg P(y,z)\}, \\
& \{P(u,u)\}, \\
& \{Q(a,b)\}\}
\end{aligned}
$$

which in PROLOG notation is

$$
\begin{aligned}
& P(x,z) \;:\!-\; Q(x,y), P(y,z). \\
& P(u,u). \\
& Q(a,b).
\end{aligned}
$$

The goal clause $G = \{\neg P(v,b)\}$ (resp. $?-\ P(v,b)$) as input leads to a non-successful computation

$$
\begin{aligned}
& (\{\neg P(v,b)\}, [\,]) \\
& \quad \vdash_F \quad (\{\neg Q(v,y), \neg P(y,b)\}, [x/v][z/b]) \\
& \quad \vdash_F \quad (\{\neg P(b,b),\}, [x/v][z/b][v/a][y/b]) \\
& \quad \vdash_F \quad (\{\neg Q(b,y), \neg P(y,b)\}, [x/v][z/b][v/a][y/b][x/b][z/b]) \\
& \quad \vdash_F \quad (\{\neg Q(b,b)\}, [x/v][z/b][v/a][y/b][x/b][z/b][y/b])
\end{aligned}
$$

which cannot be continued. Here the first, third, first, and second program clause have been used in the SLD-resolution steps.

Example

There are also two successful computations with different results. These are

$$(\{\neg P(v,b)\},[\,])$$

$$\vdash_{\mathbf{F}} \quad (\{\neg Q(v,y), \neg P(y,b)\}, [x/v][z/b])$$ (with the 1. program clause)

$$\vdash_{\mathbf{F}} \quad (\{\neg P(b,b)\}, [x/v][z/b][v/a][y/b])$$ (with the 3. program clause)

$$\vdash_{\mathbf{F}} \quad (\square, [x/v][z/b][v/a][y/b][u/b])$$ (with the 2. program clause)

and

$$(\{\neg P(v,b)\},[\,])$$

$$\vdash_{\mathbf{F}} \quad (\square, [v/b]).$$ (with the 2. program clause)

The first computation leads to the result

$$P(v,b)[x/v][z/b][v/a][y/b][u/b] = P(a,b),$$

and the second,

$$P(v,b)[v/b] = P(b,b).$$

Types of semantics

- Two ways to read a Prolog program.
  - Procedural
  - Declarative
- The Prolog rule:

  a :- b1, b2, …, bm.
- Declarative semantics (based on model theoretic semantics)
  - "a is true if b1 and b2 and … and bm are all true".
  - Declarative meaning does not depend on the order of the clauses and the order of the goals in clauses
- Procedural semantics
  - "To do a, first do b1, then do b2, …, then do bm"
  - Procedural meaning does depend on the order of goals and clauses. An unsuitable order may even lead to infinite recursive calls
- The Prolog system reads it procedurally

Procedural semantics

- Let F be a logic program and G a goal clause. The *procedural semantics of* (F, G) is defined by the set of ground instances of the computation results of F on input G which the abstract logic program interpreter can produce.

$$S_{proc}(F,G) = \{\ H \mid \text{there is a successful computation of } F \text{ on input } G \text{ such that } H \text{ is a ground instance of the computation result}\ \}$$

Example

- Consider this program:
  - male(philip).
  - female(elizabeth).
  - parent(elizabeth, charles).
  - parent(elizabeth, anne).
  - parent(philip, anne).
  - father(X, Y) :- parent(X, Y), male(X)

- Query
  - ?- father(X, anne).

Procedural semantics

1. male(philip).
2. female(elizabeth).
3. parent(elizabeth, charles).
4. parent(elizabeth, anne).
5. parent(philip, anne).
6. father(X, Y) :- parent(X, Y), male(X)

Query
?- father(X, anne).

- Prolog acts as follows:
  - Find a clause with the same predicate as the goal.
  - Copy this clause with new variable names:
    - father(X1, Y1) :- parent(X1, Y1), male(X1).
  - Unify the corresponding arguments:
    - X = X1, anne = Y1.
  - Replace the goal with the body of the clause (resolution step): (G & 6)
    - ?- parent(X1, anne), male(X1). (->7)
  - Select the first goal from the new list:
    - parent(X1, anne)
  - Find the first clause with the same predicate. (3)
  - Copy this clause with new variable names (if any):
    - parent(elizabeth, charles).
  - Attempt to unify the corresponding arguments:
    - X1 = elizabeth, anne = charles – FAILS

Procedural semantics

1. male(philip).
2. female(elizabeth).
3. parent(elizabeth, charles).
4. parent(elizabeth, anne).
5. parent(philip, anne).
6. father(X, Y) :- parent(X, Y), male(X)

Query
?- father(X, anne).

- Find the next clause with the same predicate: (4)
  - parent(elizabeth, anne).
- Attempt to unify the corresponding arguments:
  - X1 = elizabeth, anne = anne.
- Replace this goal with the body of the clause.
- (This clause was a fact, so the body is empty and Prolog moves to the next goal in the list). (4 & 7)
- The new list of goals is now:
  - ?- male(elizabeth). (->8)
- Select the first goal from the new list:
  - male(elizabeth)
- Find the first clause with the same predicate.
- Copy this clause with new variable names (if any):
  - male(philip). (1)
- Attempt to unify the corresponding arguments:
  - elizabeth = philip – Fails.

Procedural semantics

1. male(philip).
2. female(elizabeth).
3. parent(elizabeth, charles).
4. parent(elizabeth, anne).
5. parent(philip, anne).
6. father(X, Y) :- parent(X, Y), male(X)

Query
?- father(X, anne).

- Find the next clause with the same predicate.
- There are none, so go back to the last choice and undo unifications.
- The goal list again becomes:
  - parent(X1, anne), male(X1).
- Select the first goal of the list:
  - parent(X1, anne).
- Find the clause with the same predicate that comes after the last one tried:
  - parent(philip, anne). (5)
- Attempt to unify the corresponding arguments:
  - X1 = philip, anne = anne.
- Replace this goal with the body of the clause.
- (This clause was a fact, so the body is empty and Prolog moves to the next goal in the list).
- Continue until the goal list is empty, or all choices are tried.

Model Theoretic semantics

- The *model theoretic semantics of a logic program F and a given goal clause* G = .?-A1,...,Ak is the set of ground instances of (A1∧A2--∧Ak) that are consequences of F.

$$S_{mod}(F, G) = \{H \mid H \text{ is a ground instance of } (A_1 \wedge \cdots \wedge A_k) \text{ and } H \text{ is a consequence of } F\}.$$

- Declarative semantics is based on model theoretic semantics which is based on the principles
  - Herbrand Universe
  - Herbrand Base
  - Interpretation
  - Model
  - Minimal model

Declarative semantics

- Instance
  - An **instance of a clause C is the clause C with each of its variables** substituted by some terms
- Variant
  - A **variant of a clause C is such an instance of the clause C where** each variable is substituted by another variable
- Example:
  - Consider the clause
    - hasachild(X) :- parent(X,Y).
  - A variant of the clause:
    - hasachild(A) :- parent(A,B).
  - An instance of the clause:
    - hasachild(sandro) :- parent(sandro,D).

Herbrand universe

- Let P be a logic program. The **Herbrand universe of P, denoted by U(P), is the set of all ground** terms that can be formed from the constants and function symbols appearing in P.
- Let P1
  - parent(tukul,budi).
  - parent(budi,doni).
  - parent(doni,harto).
  - parent(harto,tomi).
  - ancestor(X,Y) :- parent(X,Y).
  - ancestor(X,Z) :- parent(X,Y),ancestor(Y,Z).

U(P1) := {tukul, budi, doni, harto, tomi}

- Let P2
  - nat(0).
  - nat(s(X)) :- nat(X).

U(P2) := {0, s(0), s(s(0)), s(s(s(0))), …}

Herbrand base

- The **Herbrand Base, denoted by B(P), is the set of all ground goals** that can be formed from the predicates in P and the terms in the Herbrand universe.

- For the previous examples
  - B(P1) := {parent(tukul, tukul), parent(tukul, budi), …,
    parent(tomi, doni), parent(tomi, tomi),
    ancestor(tukul, tukul), ancestor(tukul, budi),…,
    ancestor(tomi, doni), ancestor(tomi, tomi)}
  - Since there are two predicates with arity 2 and |U(P1)| = 4, then |B(P1)| = 2 * 4^2 = 32. (predicate * term ^{arity})
  - B(P2) := {nat(0), nat(s(0)), nat(s(s(0))), …}

Interpretation and Model

- An **interpretation for a logic program is a subset of the Herbrand base**
- It assigns truth and falsity to the elements of the Herbrand base.
  - A goal in the Herbrand base is true w.r.t. an interpretation if it is a member of it.
  - Otherwise, false.
- An interpretation I is a **model for a logic program if for each ground** instance of a clause in the program $A \leftarrow B1,\ldots,Bn$, A is in I if B1,…. Bn are in I.
- Intuitively, models are interpretation that respect the declarative reading of the clauses of a program.

Interpretation and Model

- M(P1) := {parent(tukul, budi), parent(budi, doni), parent(doni, harto), parent(harto, tomi), ancestor(tukul, budi), ancestor(budi, doni), ancestor(doni, harto), ancestor(harto, tomi), ancestor(tukul, doni), ancestor(budi, harto), ancestor(doni, tomi), ancestor(tukul, harto), ancestor(budi, tomi), ancestor(tukul, tomi)}

- M(P2) := {nat(0), nat(s(0)), nat(s(s(0))), …}

Minimal model

- The model obtained as the intersection of all models is known as the **minimal model and denoted by M(P).**
- The idea of this semantics is to minimize positive information.
- What is implied as true by the program is true; everything else is false.
- The minimal model is the **declarative meaning of a logic** program.
- ?-ancestor(tukul, tomi)

Example

- Given logic program
  - Ablemathematician(X) ← physicist(X)
  - Physicist(Einstein)
  - President(Trump)
- Query: ?-Ablemathematician(Trump)

- Minimal Model is : {Physicist(Einstein), President(Trump), Ablemathematician(Einstein)}

Summary

- Procedural interpretation
- Procedural semantics
- Model theoretic semantics
  - Declarative semantics

Check your understanding

- Draw procedural interpretation in a tree structure for the logic program with program clauses

$$
\begin{aligned}
&P(x,z) \text{ :- } Q(x,y), P(y,z).\\
&P(u,u).\\
&Q(a,b).
\end{aligned}
$$

- And a goal clause

$$G = \{\neg P(v,b)\}$$

Check your understanding

- Draw the tree structure for the program using procedural semantics
  - male(philip).
  - female(elizabeth).
  - parent(elizabeth, charles).
  - parent(elizabeth, anne).
  - parent(philip, anne).
  - father(X, Y) :- parent(X, Y), male(X)

  with a Query
  - ?- father(X, anne).

Check your understanding

- Find the declarative semantics of the program
  - male(philip).
  - female(elizabeth).
  - parent(elizabeth, charles).
  - parent(elizabeth, anne).
  - parent(philip, anne).
  - father(X, Y) :- parent(X, Y), male(X)

  with a Query
  - ?- father(X, anne).

Check your understanding

- Show in detail what the procedural semantics and model theoretic semantics of
  - P(a,a).
  - P(a,b).
  - P(x, y) :- P(y, x)

with the given goal clause
  - ?- P(a, z), P(z, a)
- Are.