CSc 372
Comparative Programming Languages

13: Prolog — Execution

Christian Collberg
collberg@cs.arizona.edu

Department of Computer Science
University of Arizona

Copyright © 2004 Christian Collberg
Now that we know about matching, we can take a closer look at how Prolog tries to satisfy goals.

In general, to solve a goal

\[ G = G_1, G_2, \ldots, G_m, \]

Prolog will first try to solve the sub-goal \( G_1 \).

It solves a sub-goal \( G_1 \) it will look for a rule

\[ H_i :\!-\! B_1, \ldots, B_n \]

in the database, such that \( G_1 \) and \( H_i \) will match.

Any variable substitutions resulting from the match will be stored in a variable \( \theta \).
Executing Prolog...

A new goal will be constructed by replacing $G_1$ with $B_1, \ldots, B_n$, yielding

$$G' = B_1, \ldots, B_n, G_2, \ldots, G_m.$$  

If $n = 0$ the new goal will be shorter and we’ll be one step closer to a solution to $G$!

Any new variable bindings from $\theta$ are applied to the new goal, yielding $G''$.

We recursively try to find a solution to $G'''$. 
FUNCTION Execute \((G = G_1, G_2, \ldots, G_m; \text{ Result})\);

IF Is.Empty\((G)\) THEN Result := Yes
ELSE
    Result := No;
    \(i := 1;\)
    WHILE Result=No \& \(i \leq \text{NoOfClauses}\) DO
        Clause := \(H_i :: B_1, \ldots, B_n;\)
        IF Unify\((G_1, \text{Clause, } \theta)\) THEN
            \(G' := B_1, \ldots, B_n, G_2, \ldots, G_m;\)
            \(G'' := \text{substitute}(G', \theta);\)
            Execute\((G'', \text{Result});\)
        ENDIF;
        \(i := i + 1;\)
    ENDDO
ENDIF
Executing Prolog

Goal
G1, G2, ..., Gm

Empty?
Yes
No
Succeed

Scan database
Unify(Hi, G1)

Match?
No
Yes

Database

| (1) H1 :- A1, ..., An |
| (2) H2 :- B1, ..., Bn |
| (3) H3 :- C1, ..., Cn |

Hi :- X1, ..., Xn
θ = {⋯}

Replace G1 by X1, ..., Xn
X1, ..., Xn, G2, ..., Gm
Substitute vars from θ
X1′, ..., Xn′, G2′, ..., Gm′
% From the Northern Exposure FAQ
% friend(of, kind(name, regular)).
friend(maggie, person(eve, yes)).
friend(maggie, moose(morty, yes)).
friend(maggie, person(harry, no)).
friend(maggie, person(bruce, no)).
friend(maggie, person(glenn, no)).
friend(maggie, person(dave, no)).
friend(maggie, person(rick, no)).
friend(maggie, person(mike, yes)).
friend(maggie, person(joel, yes)).
Northern Exposure Example...

cause_of_death(morty, copper_deficiency).
cause_of_death(harry, potato_salad).
cause_of_death(bruce, fishing_accident).
cause_of_death(glenn, missile).
cause_of_death(dave, hypothermia).
cause_of_death(rick, hit_by_satellite).
cause_of_death(mike, none_yet).
cause_of_death(joel, none_yet).

male(morty). male(harry). male(bruce).
male(glenn). male(dave). male(rick).
male(mike). male(joel). female(eve).
Northern Exposure Example...

alive(X) :- cause_of_death(X, none_yet).

pastime(eve, hypochondria).
pastime(mike, hypochondria).
pastime(X, golf) :- job(X, doctor).

job(mike, lawyer).
job(adam, chef).
job(maggie, pilot).
job(joel, doctor).

?- friend(maggie, person(B, yes)),
   male(B),
   alive(B),
   pastime(B, golf).
Northern Exposure Example...

friend(maggie, p(B, yes)).
male(B), alive(B),
pastime(B, golf).

friend(m, p(eve, yes)).
friend(m, m(morty, yes)).
friend(m, p(harry, no)).
friend(m, p(mike, yes)).
friend(m, p(joel, yes)).
cause_od(mike, none).
cause_od(joel, none).
alive(X) :- cause_od(X, none).
male(mike). male(joel).
female(eve).
pastime(eve, hypocondriac).
pastime(mike, hypocondriac).
pastime(X, golf) :- job(X, doctor).
job(adam, chef).
job(joel, doctor).

Scan database
Unify(Hi, G1)

Match?

Hi := X1, ..., Xn
θ = \{B=eve\}

Replace G1 by <empty>
Substitute vars from θ
male(eve), alive(eve)
pastime(eve, golf).
Northern Exposure Example...

```
male(eve), alive(eve),
pastime(eve, golf).
friend(m, p(eve, yes)).
friend(m, m(morty, yes)).
friend(m, p(harry, no)).
friend(m, p(mike, yes)).
friend(m, p(joel, yes)).
cause_od(mike, none).
cause_od(joel, none).
alive(X):−cause_od(X, none).
job(adam, chef).
job(joel, doctor).
pastime(mike, hypocondriac).
pastime(mike, hypocondriac).
pastime(X, golf):−job(X, doctor).
female(eve).
```

(1)
friend(maggie, p(B, yes)).
male(B), alive(B),
pastime(B, golf).

friend(m, p(eve, yes)).
friend(m, m(morty, yes)).
friend(m, p(harry, no)).
friend(m, p(mike, yes)).
friend(m, p(joel, yes)).
cause_od(mike, none).
cause_od(joel, none).
avive(X) :- cause_od(X, none).
male(mike). male(joel).
female(eve).
pastime(eve, hypocondriac).
pastime(mike, hypocondriac).
pastime(X, golf) :- job(X, doctor).
job(adam, chef).
job(joel, doctor).

G1

Empty?

Hi

No more Hi

fail

Succeed

No

Match?

Yes

Hi := X1, ..., Xn

θ = {B=mike}

Replace G1 by <empty>
Substitute vars from θ

male(mike), alive(mike),
pastime(mike, golf).
Northern Exposure Example...

```
friend(m,p(eve,yes)). friend(m,m(morty,yes)).
friend(m,p(harry,no)). friend(m,p(mike,yes)).
friend(m,p(joel,yes)).
cause_od(mike,none).
cause_od(joel,none).
male(mike).
female(eve).
pastime(eve, hypocondriac).
pastime(mike, hypocondriac).
pastime(X,golf):-job(X,doctor).
job(adam,chef).
job(joel,doctor).
```

G1

male(mike), alive(mike), pastime(mike, golf).

Hi

friend(m,p(eve,yes)). (1)
friend(m,m(morty,yes)).
friend(m,p(harry,no)).
friend(m,p(mike,yes)).
friend(m,p(joel,yes)).
cause_od(mike,none).
cause_od(joel,none).
alive(X):-cause_od(X, none).
male(mike).male(joel).
female(eve).
pastime(eve, hypocondriac).
pastime(mike, hypocondriac).
pastime(X,golf):-job(X,doctor).
job(adam,chef).
job(joel,doctor).
Northern Exposure Example...

friend(m,p(eve,yes)). friend(m,m(morty,yes)). friend(m,p(harry,no)). friend(m,p(mike,yes)). friend(m,p(joel,yes)). cause_od(mike,none). cause_od(joel,none). male(mike). male(joel). female(eve).

pastime(X,golf):−job(X,doctor).

alive(X):−cause_od(X,none).
pastime(eve, hypocondriac). pastime(mike, hypocondriac).

G1

alive(mike), pastime(mike, golf).

Hi

Scan database
Unify(Hi, G1)

Match?

Empty?

Yes

Succeed

No

No more Hi

fail

Hi :− X1, ..., Xn

θ = {X=mike}

Replace G1 by

cause_od(mike, none), pastime(mike, golf).
Northern Exposure Example...

friend(m,p(eve,yes)).
friend(m,m(morty,yes)).
friend(m,p(harry,no)).
friend(m,p(mike,yes)).
friend(m,p(joel,yes)).
cause_od(mike,none).
cause_od(joel,none).
alive(X):-cause_od(X, none).
male(mike).
male(joel).
female(eve).
pastime(eve, hypoc).
pastime(mike, hypoc).
pastime(X,golf):-job(X,doctor).
job(adam,chef).
job(joel,doctor).

cause_od(mike, none),
pastime(mike, golf).

Hi

G1

cause_od(mike, none),
pastime(mike, golf).

Scan database
Unify(Hi, G1)

Match?

Empty?

Yes

Succeed

No

No more Hi
fail

Hi :- X1, ..., Xn
θ = {}

Replace G1 by <empty>
pastime(mike, golf).

Replace G1 by <empty>
Northern Exposure Example...

Scan database
Unify(Hi, G1)

Match?

Hi ::= X1, ..., Xn
θ = {}
Northern Exposure Example...

- We skip a step here.
- `pastime(mike, golf)` unifies with

  `pastime(X, golf) :- job(X, doctor).`

- However, `job(mike, doctor)` fails, and we backtrack all the way up to the original query.
Northern Exposure Example...

\[
\text{match} \\
\text{Unify}(\text{Hi}, \text{G1}) \\
\text{Scan database} \\
\text{Hi} : - X_1, ..., X_n \\
\theta = \{B=\text{joel}\}
\]

Replace G1 by <empty>
Substitute vars from \( \theta \)

male(\text{joel}), alive(\text{joel}),
pastime(\text{joel}, \text{golf}).
Northern Exposure Example...

friend(m,p(eve, yes)).  (1)
friend(m,m(morty, yes)).
friend(m,p(harry, no)).  (4)
friend(m,p(mike, yes)).  (2)
friend(m,p(joel, yes)).  (3)
cause_od(mike, none).
cause_od(joel, none).
alive(X):-cause_od(X, none).
male(mike). male(joel).
female(eve).
pastime(eve, hypoc).
pastime(mike, hypoc).
pastime(X, golf):-job(X, doctor).
job(adam, chef).
job(joel, doctor).

G1

pastime(joel, golf).

Empty?

Yes

Succeed

No

No more Hi

fail

Scan database

Unify(Hi, G1)

Match?

No

Hi :- X1, ..., Xn
θ = \{X=joel\}

Replace G1 by
job(joel, doctor)
Northern Exposure Example...

```
friend(m,p(eve,yes)). (1)
friend(m,m(morty,yes)). (4)
friend(m,p(harry,no)).
friend(m,p(mike,yes)).
friend(m,p(joel,yes)). (2)(3)
cause_od(mike,none).
cause_od(joel,none)
alive(X):-cause_od(X, none).
male(mike).male(joel). (5)
female(eve).
pastime(eve, hypoc).
pastime(mike, hypoc)
pastime(X,golf):-job(X,doctor)
job(adam,chef).
job(joel,doctor).
```

Hi := X1, ..., Xn
θ = {}

**Diagram:**
- **Empty?**
  - Yes → Succeed
  - No → Unify(Hi, G1)

- **Match?**
  - Yes
    - Replace G1 by <empty>
  - No

"Hi" is applied to the database and unified with G1. If the database is empty, G1 is replaced with an empty string. Otherwise, if there's a match, G1 is replaced.
Northern Exposure Example...

friend(m,p(eve,yes)).
friend(m,m(morty,yes)).
friend(m,p(harry,no)).
friend(m,p(mike,yes)).
friend(m,p(joel,yes)).
cause_od(mike,none).
cause_od(joel,none).
aprose(X):-cause_od(X, none).
male(mike).male(joel).
female(eve).
pastime(eve, hypoc).
pastime(mike, hypoc).
pastime(X, golf):-job(X, doctor).
job(adam, chef).
job(joel, doctor).

Hi :- X1, ..., Xn
θ = { }
A term is either a
- a constant (an atom or integer)
- a variable
- a structure

Two terms match if
- there exists a variable substitution $\theta$ which makes the terms identical.

Once a variable becomes instantiated, it stays instantiated.

Backtracking undoes variable instantiations.

Prolog searches the database sequentially (from top to bottom) until a matching clause is found.