Artificial Intelligence

Programming in Prolog (‘A Short Tutorial’)

K. Buza, Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Economics and Information Systems
& Institute of Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de
Prolog

• Declarative Language
  – Statements: logic formulas
  – Query: is a formula consequence of the given formulas? At which substitution of variables is a formula a consequence of the given formulas?
  – Efficient Inference System

• Theoretical background: First Order Logic

• Several “Variants”
  – Sicstus Prolog, SWI Prolog, Mercury...

Before we start about Prolog syntax:
- Variables begin with upper case letters
- Constants begin with lower case letters
- Predicates begin with lower case letters
- Predicates end with ‘.’
- Load the knowledge base (load the Prolog program): via consult predicate (finish: CTRL-Z i.e. Strg-Z)
Before we start about Prolog syntax:

- Variables begin with upper case letters
- Constants begin with lower case letters
- Predicates begin with lower case letters
- Predicates end with '.'
- You should take these “rules” seriously! (Prolog “thinks” Anna is a variable.)
Predicates and Clauses

• person is a predicate
  – often written as person/1
  – “1” means it has one argument
  – friend/1 and friend/2 would be different predicates!

• Clauses of person/1
  – person(anna)
  – person(peter)
  – ...

K. Buza, Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany
Course on Artificial Intelligence, summer term 2008
First steps... (1)

- friend/1 and friend/2 are different predicates
- % means comment
- , (colon) means AND operation (see query)
- What does the query mean?
First steps... (2)

- All friends of Anna are friends of mine as well
- How to write it as FOL formula?
- Predicates are not the same as for example methods in Java: a method returns (at most) one value (array, object...), Prolog predicates may return several solutions
- Use ";'" to get all solutions of a predicate (i.e. all substitutions, whenever the predicate is true)

K. Buza, Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany Course on Artificial Intelligence, summer term 2008
First steps... (3)

• If you want to write longer programs, you can load them by the consult predicate (or File/Consult).

```
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 5.6.56)
Copyright (c) 1990-2008 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- consult('C:/Users/Krisztian.Buza/friends.pl').
% C:/Users/Krisztian.Buza/friends.pl compiled 0.00 sec, 1,160 bytes true.
2 ?-
```
Variables in Prolog

• No types
  – There aren’t different types for strings or numbers
• Variables like in logic, i.e. a variable is
  – either “free” (has not any value yet),
  – or substituted (has a value, which can not be changed anymore)
• Thus do not use expressions like X = X+1
• Values of the variables are atoms (or structures)
  – Atoms: anna, peter, christine,..., 1,2,3...
  – Structures: for example lists, like [1,2,3]
Lists in Prolog

• Example: [1, 2, 3, 4]

• Elements may be
  – *atoms* (1, 2, peter, anna...)
  – *structures* (for example other lists)
  – *variables* (unknown values)

• Examples:
  – [1, 2, anna, peter, X, 6, 2, 1]
  – [8, 2, [a, b, c, d], 3.14, 2.7]
  – [] (empty list)

• A list consists of
  – Head: the first element of the list
  – Tail: the list of the other elements of the list
  – Notation: [Head | Tail]
Arithmetic in Prolog

• Usual operators like +, -, *, / can be used, BUT in an unusual way: X is 8/2, Y is 10*X+3.

• Why is, why not ‘=’? ‘=’ tries to unify both structures (w.r.t. to unification algorithm)
Example: length of a list

```prolog
1 \?- consult(user).
\% list_len/2, list_len(L,N) - the length of the list L is N
\% list_len([],0).
\% list_len([H|T], N) :- list_len(T,N0), N is N0+1.
\% user://1:11:
    \% compiled 0.00 sec, 468 bytes
\% Singleton variables: [H]
\%
2 \?- list_len([a,b,c], 5).
true.
3 \?- list_len([a,b,c], 3).
fail.
4 \?- list_len([a,b,c], N).
N = 3.
5 \?- list_len(L, 5).
6 \?- list_len([anna|Tail], 5).
7 ?-```

K. Buza, Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany
Course on Artificial Intelligence, summer term 2008
Singleton Variables

• In the last example H is not used. (We have not done anything with H.)
• This may be a mistake → Warning
• In such cases you should rather use the “anonym” variable (_).

```
7 ?- consult(user).
   %size/2, size(L,N) - the length of the list L is N
   %size([],0).
   %size([_|T],N) :- size(T,N0), N is N0 + 1.
   % user://2 compiled 0.02 sec, 388 bytes
   true.
8 ?-
```
TWO + TWO = FOUR

• Knowledge Base

```prolog
1 ?- consult(user).
   : digit(0).
   : digit(1).
   : digit(2).
   : digit(3).
   : digit(4).
   : digit(5).
   : digit(6).
   : digit(7).
   : digit(8).
   : digit(9).

% user:~/1 compiled 0.00 sec, 756 bytes
```
• Use ‘=:=‘ to check whether the VALUE of two expressions are the same.
  – Remember: ‘=‘ tries to UNIFY two expressions, ‘is’ means assignment of a calculated VALUE to a variable.
TWO + TWO = FOUR

- We do not like writing long queries.
  - Remember: predicates (like solution, digit) are not functions!

```prolog
?- consult(user).
| : solution(T,W,O,F,U,R) :-
| : digit(F), digit(U), digit(R),
| : digit(T), digit(W), digit(O),
| : F \= U, F \= R, F \= T, F \= W, F \= O,
| : U \= R, U \= T, U \= W, U \= O,
| : R \= T, R \= W, R \= O,
| : T \= W, T \= O,
| : W \= O,
| : (100*T+10*W+O)+(100*T+10*W+O) =:= 1000*F+100*O+10*U+R.
| : % user://2 compiled 0.03 sec, 916 bytes
true.

8 ?- solution(T,W,O,F,U,R).
T = 3,
W = 5,
O = 7,
F = 0,
U = 1,
R = 4
```
An “IF”-like construction

```prolog
11  ?- consult(user).
   :- say_hello_if_at_least_five(X) :-
    ( X >= 5 ->
        write('five')
    ; write('sorry, this is less than 5')
    ).

% user://4 compiled 0.00 sec, 476 bytes
true.

12  ?- say_hello_if_at_least_five(5).
    five
    true.

13  ?- say_hello_if_at_least_five(3).
    sorry, this is less than 5
    true.

14  ?- say_hello_if_at_least_five(Z).
ERROR: user://4:307:
   >=/2: Arguments are not sufficiently instantiated
^ Exception: (8) _G368>=5 ? Unknown option (h for help)
^ Exception: (8) _G368>=5 ? creep
```

K. Buza, Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany
Course on Artificial Intelligence, summer term 2008
Negation

- Negation as failure (closed world semantic)
- \(+ \) predicate successes, if predicate fails
  
  \[
  \text{list\_len}( \text{[a,b,c]}, 3 ) \rightarrow \text{yes} \\
  \text{list\_len}( \text{[a,b,c]}, 5 ) \rightarrow \text{fail} \\
  \text{\(+ list\_len}( \text{[a,b,c]}, 3 ) \rightarrow \text{fail} \\
  \text{\(+ list\_len}( \text{[a,b,c]}, 5 ) \rightarrow \text{yes}
  \]
Some small exercises

• Generate list of given length
• Member of a list
• Append two lists
• Reverse list
Libraries

• Prolog has many libraries for example for list operations, CLP problems, etc.