Universidade do Minho
Escola de Engenharia
Departamento de Informática

Knowledge Representation

Prolog

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
Sistemas de Representação de Conhecimento e Raciocínio

Fevereiro 2020

ISLab
Synthetic Intelligence Lab

Arithmetic in Prolog

- Prolog provides a number of basic arithmetic tools;
- Integer and real numbers.

Arithmetic	Prolog
2 + 3 = 5	?- 5 is 2+3.
3 x 4 = 12	?- 12 is 3*4.
5 – 3 = 2	?- 2 is 5-3.
3 – 5 = -2	?- -2 is 3-5.
4 ? 2 = 2	?- 2 is 4/2.
1 is the remainder when 7 is divided by 2	?- 1 is mod(7,2).

Example queries

```
?- 10 is 5+5.
yes

?- 4 is 2+3.
no

?- X is 3 * 4.
X=12
yes

?- R is mod(7,2).
R=1
yes
```

Arithmetic

- It is important to know that +, -, / and * do not carry out any arithmetic;
- Expressions such as 3+2, 4-7, 5/5 are ordinary Prolog terms;
  - Functor: +, -, /, *
  - Arity: 2
  - Arguments: integers

ISLab
Synthetic Intelligence Lab

The is/2 predicate

- To force Prolog to actually evaluate arithmetic expressions, use:

  is

- This is an instruction for Prolog to carry out calculations;
- Because this is not an ordinary Prolog predicate, there are some restrictions.

ISLab
Synthetic Intelligence Lab

Restrictions on use of is/2

- Use variables on the left hand side of the **is** predicate;
- The variables must be instantiated with a variable-free Prolog term;
- This Prolog term must be an arithmetic expression.

Arithmetic and Lists

- How long is a list?
  - The empty list has length: zero;
  - A non-empty list has length: one plus length of its tail.

ISLab
Synthetic Intelligence Lab

Length of a list in Prolog

```
len([],0).
len([_|L],N):-
    len(L,X),
    N is X +1.

?- len([a,b,c,d,e,[a,x],t],X).
X=7
yes
?-
```

acclen/3

- The predicate acclen/3 has three arguments:
  - list whose length we want to find;
  - length of the list, an integer;
  - An accumulator, keeping track of the intermediate values for the length.

Length of a list in Prolog

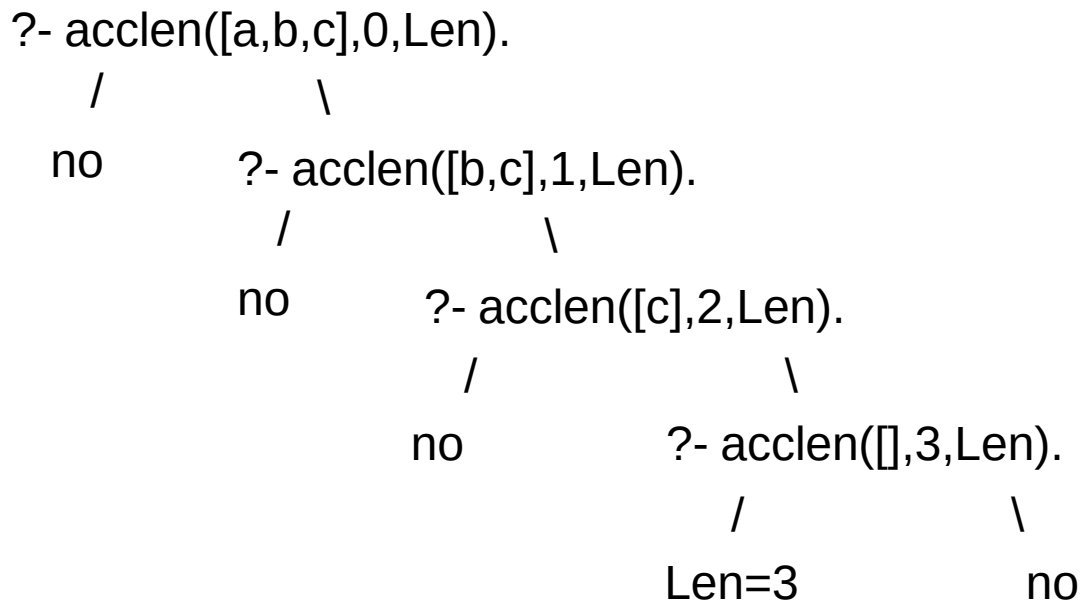
```
acclen([],Acc,Acc).

acclen([_|L],OldAcc,Length):-  NewAcc is
   OldAcc + 1,  acclen(L,NewAcc,Length).


?-acclen([a,b,c],0,Len).
Len=3
yes
?-
```

ISLab
Synthetic Intelligence Lab

Search tree for acclen/3



acclen([ ],Acc,Acc).

acclen([_|L],OldAcc,Length):- NewAcc is OldAcc + 1, acclen(L,NewAcc,Length).

ISLab
Synthetic Intelligence Lab

Adding a wrapper predicate

```
acclen([ ],Acc,Acc).

acclen([ _|L],OldAcc,Length):-   NewAcc is OldAcc +
   1,  acclen(L,NewAcc,Length).

length(List,Length):-  acclen(List,0,Length).

?-length([a,b,c], X).
X=3
yes
```

ISLab
Synthetic Intelligence Lab

Tail recursion

- Why is acclen/3 better than len/2 ?
  - acclen/3 is tail-recursive, and len/2 is not;
- Difference:
  - In tail recursive predicates the results is fully calculated once we reach the base clause;
  - In recursive predicates that are not tail recursive, there are still goals on the stack when we reach the base clause.

ISLab
Synthetic Intelligence Lab

Comparison Operators

- Have the obvious meaning;
- Force both left and right hand argument to be evaluated.

```
?- 2 < 4+1.
yes

?- 4+3 > 5+5.
no
```

ISLab
Synthetic Intelligence Lab

Comparison Operators

- Have the obvious meaning;
- Force both left and right hand argument to be evaluated.

```
?- 4 = 4.
yes
?- 2+2 = 4.
no
?- 2+2 =:= 4.
yes
```

ISLab
Synthetic Intelligence Lab

Definition of accMax/3

```
accMax([H|T],A,Max):-  H > A,
  accMax(T,H,Max).


accMax([H|T],A,Max):-  H =< A,
  accMax(T,A,Max).


accMax([],A,A).
```

```
?- accMax([1,0,5,4],0,Max).
Max=5
yes
```

ISLab
Synthetic Intelligence Lab

Adding a wrapper max/2

```
accMax([H|T],A,Max):-  H > A,
   accMax(T,H,Max).

accMax([H|T],A,Max):-  H =< A,
   accMax(T,A,Max).

accMax([],A,A).

max([H|T],Max):-
   accMax(T,H,Max).
```

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Knowledge Representation

Prolog

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
Sistemas de Representação de Conhecimento e Raciocínio

Fevereiro 2020