# Prolog (1)

Comp3031 Lab 08

Fall 2020

Xibo SUN & Yongchao WANG

# Introduction to Prolog

- Prolog is short for PROgramming in LOGic.
- A “Relational Programming” language.

```
csl2wk01 ~ $ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
```

- Load file in Prolog
  - [filename].  %load “filename.pl”
  - [‘filename.xx’].  % load “filename.xx”

# Simple Terms

- Atom
  - ?- atom(‘John’). % true.
  - ?- atom(hongkong). % true.
  - ?- atom(_). % false.
- Variable
  - ?- var(_). % true.
  - ?- var(X). % true.
  - ?- var(Less). % true.
- Number
  - ?- number(1.2e24). % true.
  - ?- integer(23). % true.
  - ?- float(12.3). % true.

# Structures

- Structures are also called compound terms
- Syntax:
  - <functor> (<terms>)
  - <functor> is the name of relations

# Clauses

- Prolog programs describe relations by clauses
- There are two types of clauses: facts and rules
- Facts
  - Syntax:
    - <fact> := <functor>(<terms>). | <functor>.
    - <terms> := <term> |<terms>,<term>
    - <term> := <atom> | <variable> | <number> | <functor>(<terms>) | <functor>
  - Example
    - parent(‘John’, ‘James’).
    - male(‘John’).
    - age(‘John’, 60).

# Clauses

- Rules
  - Syntax:
    - Head :- Body.
    - Read as: “Head is true if Body is true”.
  - Body consists of calls to predicates.
  - The comma “,” is logical conjunction, meaning *and*.
    - p :- p1, p2, …, pn.
  - The semicolon “;” can be used in the body for *or*.
    - p :- p1; p2; …; pn.
  - Example
    - sibling(X, Y) :- parent(_1, X), parent(_1, Y).

# Predicates

- Built-in predicates in Prolog
  - File input/output predicates: read, write, etc.
  - Control predicates: ;, true, false, etc
  - Arithmetic predicates: +, *, is, etc.
  - List of built-in predicates
    - https://www.swi-prolog.org/pldoc/man?section=builtin
- Structures are predicates.
  - family.pl defines six predicates
    - parent/2
    - age/2
    - male/1
    - female/1
    - sibling/2
    - ancestor/2

# Comparison Predicates

- Term comparisons to compare the terms literally
  - ==, \==
- Arithmetic comparisons to compare the arithmetic values of the terms
  - =:=, =\=, <, =<, >, >=
- Example

```
?- monday == 'Monday'.
false.
?- monday == 'monday'.
true.
?- 2+1 == 3.
false.
```

```
?- 2+1 =:= 3.
true.
?- 2+1 \== 3.
true.
?- 2+1 =\= 3.
false.
```

# Query

- Load the file “family.pl”

```
?- [family].
true.
```

- *Is John a male?*
  - ?- male('John').
  - true.
- *Who is James’ parent?*
  - ?- parent(X, 'James').
  - X = 'John'.
- *Who is John’s child?*
  - ?- parent('John', Y).
  - Y = 'James' ;
  - Y = 'Mary'.

- Prolog can produce all of the possible answers
  - If the user types a semicolon ';' , Prolog will look for a next answer
  - If the user just hits Enter, then Prolog stops looking for answers

# Recursive Definition

- A predicate is recursively defined if it refers to itself in the rule definition.
- Example
  - ancestor(X, Y) :- parent(X, Y),
  - ancestor(X, Y) :- parent(_1, Y), ancestor(X, _1).
  - Query: *John is who's ancestor?*
    - ?- ancestor('John', X).
    - X = 'James' ;
    - X = 'Mary' ;
    - X = 'Judy' ;