Notes lecture 2

Prolog language

* Prolog Program
  + program - collection of logical statements, each being a Horn clause of the form
  + conclusion to be demonstrated - in form
* Control structure
  + ***fact*** - what is known to be true
  + ***rule*** - what can be deduced from given facts (indicates a conclusion that is known to be true when other conclusions or facts are true)
  + ***goal*** – conclusion to be proven
* Translations

|  |  |
| --- | --- |
|  | “q :- p.” (𝑞 if p.) |
|  | “,” |
|  | “;”/ using 2 separate clauses |

* Constant – starts with lowercase letter
* Variable
  + starts with uppercase letter
  + the anonymous variable is represented by the underline character (\_)
* Comments
  + % …
  + /\* … \*/
* Operators

|  |  |
| --- | --- |
| X = Y | Checks if X and Y can be unified |
| X \= Y | Cannot be unified |
| X == Y | Checks if X and Y can be bound to the same value |
| X \== Y | Have not been bound to the same value |
| =:= | * Test arithmetic equality * Forces arithmetic evaluation of both sides * Operands must be numeric * Variables are BOUND |
| =\= | “different” arithmetic operator |
| <variable> is <bound&numeric> | * If var is bound, check numerical equality (like =:=) * If not, evaluate right side and var is related to the result |

* Predefined arithmetic functions:

|  |
| --- |
| mod(X,Y)  X mod Y |
| div(X,Y)  X div Y  X // Y |
| abs(X) |
| sqrt(x) |
| round |
| truncate |
| floor |
| ceiling |
| random |
| between |

* Standard order of terms:
  + …