MA410 Prolog Practical 1 - Knowledge Representation

1 Facts & Rules w/o variables

Create a directory `lab1` and save all files relating to this lab in this directory.

Example 1

1. Download file `lab1and.pl`.
2. Record the follow facts:
   - "I am sleepy." : `i_am_sleepy.`
   - "It is nighttime." : `it_is_nighttime.`
3. And the rule:
   "If I'm sleepy ***and*** *it's nighttime then I'll go to bed."*
   `i_go_to_bed :- i_am_sleepy, it_is_nighttime.`
4. Go into prolog & load the file by typing:
   `?- consult('*lab1and.pl').`
   where `*` stands for file pathname.
5. Query the knowledge base:
   ```
   ?- it_is_nighttime.
   ?- i_go_to_bed.
   ```

Example 2

1. Download file `lab1or.pl`.
2. Record the follow facts:
   - "It is raining" : `it_is_raining.`
   - "The Referee is sick" : `ref_is_sick.`
3. And the rule:
   - "If ref is sick **or** it's raining then no football."
   `no_football :- it_is_raining; ref_is_sick.`
4. Ask the knowledge base if there is no football:
   ```
   ?- no_football.
   ```
5. Define the rule:
   ```
   football :- not(no_football).
   ```
6. Ask the knowledge base if there is football.

Task: Create amusing examples based on example 1 and 2 above and e-mail the code with subject heading "KR Lab - Examples 1 and 2".

2 Facts & Rules with variables

1. Download file `lab1vars.pl`.
2. Record the follow facts:
   "the book is on the table" : `on(book, table).`
   "the cup is on the book" : `on(cup, book).`
3. And the rule:
   - *"If X is on Y and Y is on Z then X is on Z."*
   `on(X,Z) :- on(X,Y), on(Y,Z).`
4. Query what is on what: `?- on(X,Y).`
5. Verify that the cup is on the table.
6. Create a rule: *"If X is on Y then Y is under X."*
7. Query what is under what.

3 Knowledge Representation Issues

Download file `lab1iss.pl` for this section.

Relationships

In regard to relationships, we consider attributes of an object such as inverses & existence. Consider the inverse example in `band(bono, u2).` We can treat this as Bono plays in band U2 or else Bono's band is U2.

Granularity

At what level should the knowledge be represented and what are the primitives? We need to choose the granularity of representation (i.e. which words are our primitives). Let's take an example, the barman serves a customer: `serves(barman, customer).` but we can also have the barman gives the customer a pint: `gives(barman, customer, pint).` Are these the same? If so, we need to define a rule to ensure they are: `give(X, Y, pint) :- serves(X,Y).`

Task: Create an example for each that illustrates the two issues. E-mail the code with subject heading "KR Lab - Relationships and Granularity".

3.1 Towards Natural Language

See `http://www.cse.unsw.edu.au/~billw/prologdict.html` and read about the `op` command.

1. Add the following into `lab1vars.pl`:
   ```
   eats(dog, food).
   :- op(600, xfy, eats).
   ```
2. Query in prolog: `?- dog eats food.`
3. Now enter in: `the(eats(dog, food)).`
   `:- op(700, fx, the).`
4. Query in prolog: `?- the dog eats food.`

Task: Write code so user can check such things as
`?- cup on table.` `?- dog under table.`
Create 3 new relationships between new objects. E-mail code with heading "KR Lab - Towards NL".