

# CS2910 Assessed Coursework 2

~~Original deadline: 23:59 on the 25/03/20.~~ **Revised deadline:** 23:59 on the 15/04/20.

**Feedback by:** 28/04/20.

## 1 Learning outcomes assessed

This assignment covers some of the topics under *knowledge representation*, *default* and *temporal reasoning*. In particular, the outcomes assessed are knowledge and understanding on:-

- what to represent given a domain specific problem;
- how to translate descriptions of problems, including *semantic networks* and *narratives* into logic-based symbolic representations;
- implement the resulting computational inferences required to reason about knowledge representation problems using Prolog.

## 2 Instructions

You will submit your work electronically by means of the Department's submission script called `submitCoursework`. You may resubmit your script any number of times, though only the last submission will be kept. The submission will occur on `linux.cim.rhul.ac.uk` and the protocol is `submitCoursework <your directory>`. For example, assuming the directory with the solutions is `assignment1`, then you should submit by typing in the following command:

```
submitCoursework assignment2
```

and then follow the instructions accordingly. Your directory should contain the following Prolog files (see detailed assignment specification later in Section 3):

- `students.pl`;
- `semnet.pl`;
- `sitcalc.pl`.

The files you submit cannot be overwritten by anyone else, and they cannot be read by any other student. You can, however, overwrite your submission as often as you like, by resubmitting, though only the last version submitted will be kept. Submission after the deadline will be accepted but it will automatically be recorded as being late and is subject to College Regulations on late submissions. Please note that all your submissions will be graded anonymously, so please do not put any references of your name or College identifier anywhere in your submission.

**NOTE:**

All the work you submit should be solely your own work. Coursework submissions are routinely checked for this. Any assessment offence will be investigated subject to the College regulations.

### 3 Tasks for this assignment

1. Consider the following facts:

- Mary, Pat, Bob and Tony are all first year Computer Science students.
- They were awarded 75, 80, 63, and 57, respectively, for coursework.
- They were awarded 70, 25, 85, and 65, respectively, in programming.
- In exams cs1111, cs1112, cs1113, they were awarded 60, 65, 70 (Mary), 75, 80, 63 (Pat), 40, 50, 80 (Bob), 20, 80, 90 (Tony).

The regulations of the University that Mary, Pat, Bob and Tony attend state:

- In order to pass the year, students need to have a minimum mark of 30 in their programming.
- Students with one exam below 30 need to resit in September (thus they do not immediately pass).
- Students with two exams below 30 fail the year.
- Students must have an average of 40 or more in order to pass the year.
- Students with less than 50 marks in the coursework can pass the year only if they have an exam average of 50 or more.

Represent the above knowledge as a Prolog program and store it in a file called `students.pl`. Compile your program and use it to prove whether Mary, Pat, Bob and Tony will pass and progress to the second year or whether they must resit an exam in September. [40%]

2. Represent the semantic network of Fig. 1 as a Prolog program saved in a file called `semnet.pl`. Make sure you represent both the `isa` hierarchy as well as the default values for the *height* and *has nose* properties.

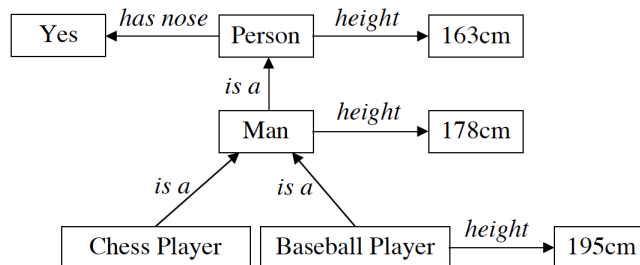


Figure 1: A semantic network.

Exemplify the use of the network by considering the following three instances:

- Gregory is an instance of a Baseball Player;

- Dave is an instance of a Chess Player;
- Sara is an instance of a Person.

Compile your program and ensure that the inferences provide the expected answers for the default values of the *height* and *has nose* properties of these instances. [30%]

3. Consider the following narrative:

Mary and Bob live in the UK. Mary owns a Prolog book (**b1**), entitled “Prolog primer”, originally bought for the value of 20 GBP. She now sells the book to Bob at the price she bought it for. Bob can afford it, he has 50 GBP, while Mary only has 10 GBP prior to the sale. Bob is happy after the sale because he owns a Prolog book. When the book was hers, Mary was happy too. After the sale, the book lost 20% of its original value.

Represent the above story in the Situation Calculus using Prolog and store it in a file called `sitcalc.pl`. In your representation, decide which parts of the story can be specified as rigid facts, what constitutes a reasonable initial state, how to represent the sale’s preconditions and effects, and how you formulate the frame axiom. Compile your program and see which parts of the state, according to your representation, change after the sale of the book and which ones remain the same. [30%]

## Marking criteria

- Full marks will be given for implementations that address the requirements of all the tasks as specified in this document.
- Marks will be allocated to the representational adequacy (ability to represent the required knowledge), the inferential adequacy (ability to manipulate the knowledge represented to produce new knowledge corresponding to that inferred from the original) and the inferential efficiency (ability to direct the inferential mechanisms into the most productive directions) of the representations proposed as well as their implementation.
- Marks will also be allocated in solutions which show understanding of Prolog unification, especially on the use of lists, the use of existing primitives that manipulate and generate data structures (like `append/3` and `findall/3`), including the relevance of all these to the specific task at hand.
- Code quality: indentation, comments, variable naming, use of ‘\_’ variables, and appropriate use of Prolog control operators (e.g. the cut operator (!)).
- It is expected that the files you will submit are created using Linux and not Windows, as the marking will be done on a Linux machine. The code should run in SWI Prolog (Version 7.6.4), the one we use at `linux.cim.rhul.ac.uk`. Implementations in any other Prolog or programming language will not be accepted.

- Your code should compile successfully for full marks. If part of your code does not compile, then wrap it in a comment of the form:

```
/* Partial Code:  
....  
End of Partial Code */
```

and we will try to mark any logic that is relevant to the required task.

KS-21/03/20.