Logic Programming - Assignment 2

Grammars

Alípio Jorge, 2018

Synopsis

1. This assignment is part of the whole assignment based evaluation. It counts for 2/3 of the whole practical mark.

2. This part of the assignment is about writing a grammar that recognizes polynomials.

3. The assignment is to be done by the same groups.

Objectives

1. Each group implements the indicated predicates and all auxiliary predicates required.

2. The program should be able to understand plynomials written in "free" text (with some limits).

3. The main deliverable is the Prolog code to be submitted through moodle as a .pl file . The code should be runnable is swipl and ready to be tried by the evaluator. The code should be organized, comented and indented.

4. A README that describes how to run the code file should accompany the submisison and indicate successful and unsuccessful queries (if any) resolved by the submitted code.

Tasks

Define the predicates:

1. `text2poly/2` that recognizes a text (string) in the first argument and produces a polynomial as a term in the second argument.

- The free text can be someting like “two times x plus three times y raised to four”. This would yield polynomial $2 * x + 3 * y\hat{\ }4$.
- The recognizer is to be based on a DCG and can start by tokenizing the string (splitting by spaces) and producing a list of words. This is done with built-in predicate split_string/4.
- The free text should allow some liberties as in “two x squared” meaning $2 * x\hat{\ }2$.
- The set of possible variables can be limited or somehow defined (one or two letter words...).
- The set of possible numbers can be defined in increasingly more complete ways. Start with digits (one, two, three, ...). Move to integers. Move to floats.

2. Build a demo polyplay/0 that allows the input of polynomials and operations over polynomials and scalars as text and translates the input given to polynomials as terms and the shows the output as a polynomial.

    - The predicate asks for instructions (using a prompt and starting a specific environment, for example) and executes the instructions. One example is “add two times x to four times x”. Another is “simplify polynomial five x plus 3 times x”. The set of instructions is supposed to be simple and predictable, but you can extend as you wish.
    - There should be natural language instructions to ask for help and to stop the environment.
    - You can imagine other useful instructions such as defining legal variables or naming and storing polynomials for later use (“add polynomial p1 and polynomial p2 as p3”).

Evaluation

1. The predicates wil be evaluated according to ellegance, correctness, efficiency and completeness.

2. Part of the evaluation may be done as a practical or written test and/or oral defense.

3. Equal or unlikely equivalent answers by different groups may penalize both groups to a limit of both groups getting a zero mark.

Organization

1. Groups must be declared in moodle.

2. Students may be questioned in the labs and this counts for the individual evaluation. If groups are not present in the labs this may penalize the evaluation.

3. Marks are individual, although group elements tend to be homogeneous.

4. The deadline for this part of the assignment is December 20th. Further developments can be added until the day of the exam, but will be limited in value by 10%.

5. Remember that in order to obtain mininal attendance (frequência) you must have a note different from zero in the assignment evaluation component.

6. Although research is encouraged, all the code must be fully understood by the group members. To fail this, may decrease individual grades significantly.

Ethics

1. Upon submission the elements of the group commit to follow ethical principles. Any evidence of ethical violations will constitute grounds for marking the assignment with a zero.

2. The work is original and done solely by the elements of the group. Any external contribution must be declared explicitely in the README file and in the header of the code file.

Deadlines

1. The deadline is defined above.

2. Groups will have to defend their work in a session to be scheduled.