Logic Programming - Assignment 1
Symbollic Manipulation

Alípio Jorge, 2018

Synopsis

1. This assignment is part of the whole assignment based evaluation. It counts for 1/3 of the whole practical mark.
2. This part of the assignment is about implementing a symbollic manipulator for polynomials.
3. The rest of the assignment will focus on other matters.
4. The assignment is to be done by groups of 2 or 3 students.

Objectives

1. Each group implements the indicated predicates and all auxiliary predicates required.
2. The program should be able to simplify polynomials using different representations as requested.
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. poly2list/2 that transforms a list representing a polynomial (second argument) into a polynomial represented as an expression (first argument) and vice-versa.

- The polynomial is represented as an expression as shown in lectures, using the usual arithmetic operators, ^for exponent, lower case letters such as x, y, ,z etc. for variables, and floating point notation positive and negative numbers for coefficients. Examples of polynomials are `2*x^2+5+y*2`, `7-z` and `0`.
- As a list, the polynomial is decomposed into its monomials and each is an element of the list. Examples of list polynomials are `[2*x 3^2, 5, y*2]`, `[7, -z]` and `[0]`

2. `s`impoly_list/2 that simplifies a polynomial represented as a list into another polynomial as a list.

3. `s`impoly/2 that simplifies a polynomial represented as an expression as another polynomial as an expression. This predicate can use the previous one.

4. `s`calepoly/3 that multiplies one polynomial as expression by a scalar resulting in a seconed polynomial. The two first arguments are assumed to be ground. The polynomial resulting from the sum is in simplified form.

5. `a`ddpoly/3 that adds two polynomials as expressions resulting in a third one. The two first arguments are assumed to be ground. The polynomial resulting from the sum is in simplified form.

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 first part of the assignment is November 25th. The full assignment should be due by the day of the exam.

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

6. This part of the assignment is worth 16.6% of the final mark (4 values).

7. 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.