Advanced Programming Exam

2021-22 Summer Session

Problem description

A *polynomial algebraic expression* is either:

- a *variable*;

- a constant value in $\mathbb{Z}$;

- the sum, the subtraction, or the product of two polynomial algebraic expressions.

For instance, $(x - 4 * y + 3) * x - 2$, $z * z$, and 0 are all polynomial algebraic expressions having 2, 1, and 0 variables, respectively.

The polynomial algebraic expressions e_1 and e_2 are *equivalent* whenever the former can be rewritten as the latter by:

- interpreting the sums, subtractions, and products between constants as the standard arithmetic over $\mathbb{Z}$, e.g., $-3 + 1 * 5$ and 2 are equivalent;

- assuming the commutative law, e.g., $x * (y + 2)$ and $x + 3$ are equivalent to $(2 + y) * x$ and $3 + x$, respectively;

- accepting the distributive law, e.g., $(x+3) * y$ is equivalent to $x * y + 3 * y$.

If e is a polynomial algebraic expression and x one of its variables, then e can be rewritten as an equivalent polynomial algebraic expression:

$$a_0 x^0 + a_1 x^1 + \ldots + a_{n_x} x^{n_x}$$

where x^i to denote the i-th natural power of x, i.e.,

$$x^i = \begin{cases} 1 & \text{if } i = 0 \\ x * x^{i-1} & \text{if } i > 0 \end{cases},$$

n_x is the maximum degree of x in e, and the a_i's are the (possible non-constant) x-coefficients of e of degree i. For instance, $(x - 4 * y + 3) * x - 2$ is equivalent to

$$-2 * x^0 + (3 - 4 * y) * x^1 + 1 * x^2.$$

Thus, -2, $(3/2 - 4 * y)$, and 1 are the expression x-coefficients of degree 0, 1, and 2, respectively.

Assignment

Design two classes `Var` and `Expr` to represent variables and polynomial algebraic expressions, respectively. In particular, beyond the constructors, the class `Expr` must provide the following public methods/friend functions:

- `std::vector<Var> get_variables() const` to get a list of the variables in an expression;

- `std::map<unsigned, Expr> get_coeffs(const Var& x) const` to get a map that relates the degree associated to non-null x-coefficients to the x-coefficients themselves;

- `Expr &replace(const std::map<Var, Expr>& repl)` to replace every occurrence of a variable by a specified polynomial algebraic expression;

- `std::ostream &operator<<(std::ostream&, const Expr&)` to print in a stream a polynomial algebraic expression;

- `bool equivalent(const Expr& e1, const Expr& e2)` to test whether two polynomial algebraic expressions are equivalent.

Few functions showing the classes features are also requested (e.g., by using the Boost test library).

Exam and Deadline

The exam must occur during the summer exam session 2021/22, i.e., between June 9 and July 29, 2022. The exam date must be individually scheduled in agreement with the teacher. To plan it, please write to acasagrande@units.it.

The assignment solution must be sent to acasagrande@units.it at least 10 days before the scheduled exam date.