Fraction Calculator: Prolog Assignment

Grao en Intelixencia Artificial, Lógica

March 2025

1 Introduction

In this assignment, we will implement a fraction calculator in Prolog that supports basic arithmetic operations, fraction simplification and aggregate functions. To that aim, we will implement a new infix opertor `is_frac/2` that accepts arithmetic expressions with fractions. We will also implement a set of arithmetic and aggregate functions that operate on lists of fractions.

2 Fractions

The `is_frac` operator must accept either an integer number or fractions of the form N/D where both the numerator and denominator are integer numbers. Any other type of input should be considered invalid. Division by zero should also be considered invalid, that is, the denominator should be greater than 0 ($D > 0$).

2.1 Examples

Below are given some examples of queries that the operator should accept:

```
?- X is_frac 3.
X = 3/1.

?- X is_frac 4/3.
X = 4/3.

?- X is_frac -5/7.
X = -5/7.

?- X is_frac 7/0. % Denominator should be greater than 0.
false.
```

3 Arithmetic Operations

The system should support the following operations with fractions:

- Addition: `X is_frac A + B`
- Subtraction: `X is_frac A - B`
- Multiplication: `X is_frac A * B`
- Division: `X is_frac A div B`
- Negation: `X is_frac -A`

```
?- X is_frac -2/10 + 1/3.
X = 2/15.

?- X is_frac (1 / -3) div (6 / -5).
X = 5/18.

?- X is_frac 3 * -(4 div 2).
X = -6/1.
```

Note that the result is always a simplified fraction.

3.1 Simplification

The fractions will be simplified to their lowest terms when possible. This simplification should be done automatically after each operation.

```
?- X is_frac 5/10.
X = 1/2.

?- X is_frac 4/6 + 4/3.
X = 2/1.
```

4 Aggregate Functions

The calculator should also support aggregate functions over lists of fractions:

- `Min is_frac min(L)`: Computes the minimum fraction in the list.
- `Max is_frac max(L)`: Computes the maximum fraction in the list.
- `Sum is_frac sum(L)`: Computes the sum of all elements in the list.
- `Avg is_frac avg(L)`: Computes the average value of the elements.

4.1 Examples

```
?- L=[1/2, 3/4 - 5, 2, 3/5], N is_frac min(L), X is_frac
    max(L), S is_frac sum(L), A is_frac avg(L).
L = [1/2, 3/4 - 5, 2, 3/5],
N = -17/4,
X = 2/1,
S = -23/20,
A = -23/80.
```

Note that the syntax for the use of the aggregates must adhere to how they are defined in the examples. Yet, auxiliary predicates `min/2`, `max/2`, `sum/2` and `avg/2` might be defined, that will compute the minimum, maximum, sum and average of a list of fractions, respectively.

Important Notes

- We will use the arithmetic operators already defined by Prolog. For avoiding priority clashes, it is important to define the `is_frac` operator with a high priority number.

  ```
  :- op(700, xfx, is_frac).
  ```

- You are not required to implement your own version of `gcd/2` to compute the greatest common divisor of two numbers.
- We will assume that the "-" operator is always applied on the numerator. For example, 1/-3 should be interpreted as -1/3.
- You are allowed to use already existing predicates to test the type of a Prolog term, e.g. `number(X)` (see slide number 47 of the Prolog notes).
- **A task will be enabled in moodle for the delivery of the practice, dated 28 March at 23:59 hours. Submissions after this deadline will not be evaluated.**