

Homework # 12

due Monday, April 27, 11:00 AM

1 Arithmetic

As mentioned in the textbook, arithmetic is not well integrated into Prolog. Suppose `<` and other relation operators were fully invertible predicates that would work on variables (assuming integers only for now):

```
?- X>2, X<5.
X = 3;
```

```
X = 4
```

Yes

Suppose we wanted a predicate `eval` that would evaluate arithmetic terms, which would work even with variables:

```
?- eval(X*X+1,5).
X = 2;
```

```
X = -2
```

Yes

```
?- eval(X,2).
X = 2;
```

```
X = 0+2;
```

```
% many, many more solutions
```

Yes

Write (on paper) the definition of `eval` that can handle arbitrary expressions involving addition and multiplication with variables and constants. Your definition may use (unimplemented) predicates `add(_,_,_)`, `mul(_,_,_)`, and `isNum(_)`.

Then answer the following question:

Why do you think Prolog does not do things this way?

Hint: think about how `add(_,_,_)` and `isNum(_)` would be implemented and how they would work. Your discussion should include information on the cost model for these (hypothetical) predicates.

2 Optimization

Do Exercises 22.2 and 22.8. Your solution should `subseq` (from the textbook) and `minlist` (analogous to your answer to 22.2). It is permitted to return the same (optimal) cover more than once.

Put your solutions in the file `homework12.pl`.