

# Homework # 12

## due Tuesday, April 26, 3:30 PM

### 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

One useful search is to find the lowest cost path between two nodes in a graph. In CS 351, we investigate depth-first and breadth-first traversals for this problem. In Prolog, the problem is that “marking” a node after visiting it doesn’t fit well with side-effect free structures. But without marking, it is difficult to prevent endless loops from being investigated.

One way around this difficulty is to establish a bound on the cost and then cut short any path that strays beyond the bound. To find the best path, one collects all the paths that remain below the bound and then finds the cheapest.

The goal of this assignment is to implement a predicate `minpath` using this idea (a maximum bound):

```
?- minpath(ems,library,10,L).
L = [ems, chem, lap, bridge, union1, lubar, bolton, library] .

?- minpath(ems,curtin,10,L).
L = [ems, chem, lap, bridge, union1, lubar, bolton, library, music|...] .

?- minpath(ems,library,8,L).
L = [ems, chem, lap, bridge, union1, lubar, bolton, library] .

?- minpath(ems,curtin,8,L).
No.
```

Notice that if one chooses a bound that is too small, `minpath` may fail. if you choose too big a bound, it takes longer to execute.

This design requires a large number of helper predicates. We provide documentation for all helper predicates our solution uses. We recommend you implement these in the order we give them in the skeleton file, `path.pl.SKEL`.

Put your solution in the file `path.pl`.