
CS 421 — PROLOG Activity

Mattox Beckman

Introduction

PROLOG is an unusual language. Instead of telling the computer how to compute the solution, you tell the computer what the solution should look like, and let the computer figure out how to compute it.

This activity will have you make a few PROLOG programs to get used to it.

Questions

1. Write a PROLOG predicate `sum(X,S)` that is true when `S` is the sum of the elements in list `X`.

```
1 -? sum([1,2,3],S).  
2 S = 6
```

2. Write a PROLOG predicate `append(X,Y,Z)` that is true when `Z` is the result of appending `X` to `Y`.

```
1 -? append([1,2,3],[4,5,6],Z).  
2 Z = [1,2,3,4,5,6]
```

3. Write a PROLOG predicate `isprefix(X,Z)` that is true when `X` is a proper prefix of `Z`.

```
1 -? isprefix([1,2,3],[1,2,3,4,5,6]).  
2 Yes
```

4. Write a PROLOG predicate `pathFrom(A,B,L)` that is true when there is a path from A to B specified by L. E.g., if you have

```
1 connected(a,b) .  
2 connected(b,c) .  
3 connected(c,d) .  
4 connected(a,x) .
```

Then the query `pathFrom(a,d,L)` will return `L = [a,b,c,d]`.

5. Write a PROLOG predicate `pathCost(A,B,C)` that shows the cost of the path between two points. Assume your `connected` predicates have the form `connected(a,b,10)` — i.e., a is connected to b with cost 10.