# CS 421 --- Dynamic PROLOG Activity

| | | |
|---|---|---|
| Manager | Keeps team on track | |
| Recorder | Records decisions | |
| Reporter | Reports to class | |
| Reflector | Assesses team performance | |

## 1 Solving MP 6

For MP 6 you are going to write a type inferencer in HASKELL. I mentioned I would show you how to do it in PROLOG, but this is a POGIL classroom! That means *you* get to do it.

Consider this database:

```
lookup(K,V,[(K,V)|_]) :- !.
lookup(K,V,[_|XS]) :- lookup(K,V,XS).

tc(Gamma,X,int,pt(const,Gamma,X,int)) :- integer(X).
tc(Gamma,X,T,pt(var,Gamma,X,T)) :- atom(X), lookup(X,T,Gamma).

tc(Gamma,X+Y,int,pt(plus,Gamma,X+Y,int,[XPT,YPT])) :-
    tc(Gamma,X,int,XPT), tc(Gamma,Y,int,YPT).
```

Sample Run...

```
?- tc([],10,T,P).
T = int,
P = pt(const, [], 10, int).

?- tc([(x,int)],x+3,T,P).
T = int,
P = pt(plus, [(x, int)], x+3, int, [pt(var, [(x, int)], x, int), pt(const, [(x, int)], 3, int)]) ;

?- tc([(x,int)],x + 3.1, T, P).
false.
```

**Problem 1)** The `tc` predicate has four parameters. What are they for?

**Problem 2)** For the `plus` rule, how does it correspond to the big step semantics rule?

**Problem 3)** Add a rule to handle equality checks.

**Problem 4)** Now add a rule that type checks `if` expressions. Use `if(C,T,E)` as your representation.

## Modifying Gamma

Consider this new clause.

```
tc(Gamma,fun(X,Body),T1->T2,pt(fun,fun(X,Body),T1->T2,[PTB])) :-
    tc([(X,T1)|Gamma],Body,T2,PTB).

?- tc([],fun(x,x+3),T,PT).
T =  (int->int),
PT = pt(fun, fun(x, x+3),  (int->int),
        [pt(plus, [(x, int)], x+3, int,
            [pt(var, [(x, int)], x, int),
             pt(const, [(x, int)], 3, int)])])
```

**Problem 5)** What just happened here? Did you see how the value of `Gamma` was updated?

**Problem 6)** Try to write the proof checking rule for function applications. Use `app(F,X)` to represent it.

## 2 Let's Write an AI!

Here is some base code to write a bot. This one will remember things you tell it!

```
check(L) :- known(L), !,
            writef("I already know about that."), nl.
check(L) :- assert(known(L)), assert(L).
forget(L) :- retract(known(L)), retract(L).

newclause(L) :- aclause(L), !.
newclause(L) :- assert(aclause(L)), assert(L).

respond([bye]) :- known(name(X)), writef("Bye, "),
                  writef(X), nl, abort.
respond([bye]) :- writef("Bye!"), nl, abort.

respond([my,name,is,X]) :-
   forget(known(name(_))), fail.

respond([my,name,is,X]) :-
   !, check(known(name(X))),
   writef("Hello, "), writef(X), nl.

respond([what,is,my,name]) :-
   !, known(name(X)) ->
   (writef("Your name is "),
      writef(X), nl, !);
      !, writef("I don't know."), nl.
```

There is a parser that we don't show here (but you can find in your repository). There's also a REPL that calls `respond` with the sentence you enter.

**Problem 7)** What are `check` and `forget` trying to do?

**Problem 8)** What happens if I enter ``my name is master.'' into the REPL? How does the AI remember this? Does it behave differently as a result?

```
respond([X,is,Y]) :- !, L =.. [Y,X], check(L), newclause(Y).

respond([X,is,not,Y]) :- !, L =.. [Y,X],
              check(false(L)), newclause(false(Y)).

respond([all,X,are,Y]) :- !,
   C1 =.. [Y,Z],
   C2 =.. [X,Z],
   L =.. [(:-),C1,C2],
   newclause(X),
   newclause(Y),
   check(L),
   writef("Okay."), nl.

respond([is,X,Y]) :-
   !, L =.. [Y,X],
   call(L) -> (!, writef("Yes."), nl);
     (!, writef("I don't know."), nl).
```

**Problem 9)** What is the `L =.. [Y,X]` syntax doing in `respond`?

**Problem 10)** The `[is,X,Y]` code is using `call` instead of `check`. Why is that?

**Problem 11)** Describe how the `[all,X,are,Y]` code works.

**Problem 12)** The `[is,X,Y]` code does not know how to check for falsehood. Can you add that?

# Dynamic Prolog Activity --- Team's Assessment (SII)

Manager or Reflector: Consider the objectives of this activity and your team's experience with it, and then answer the following questions after consulting with your team.

1. What was a **strength** of this activity? List one aspect that helped it achieve its purpose.

2. What is one things we could do to **improve** this activity to make it more effective?

3. What **insights** did you have about the activity, either the content or at the meta level?

# Dynamic PROLOG Activity--- Reflector's Report

| | | |
|---|---|---|
| Manager | Keeps team on track | |
| Recorder | Records decisions | |
| Reporter | Reports to Class | |
| Reflector | Assesses team performance | |

1. What was a strength of your team's performance for this activity?

2. What could you do next time to increase your team's performance?

3. What insights did you have about the activity or your team's interaction today?