CSC 372, Spring 2025

Starting LA2 Type Inference and Prolog Example Problems

Michelle Strout

THE UNIVERSITY OF ARIZONA

March 4, 2025

Plan

- **Announcements**
  - LA2 has been posted
- **Last time**
  - Type inference for SML on ASTs
- **Today**
  - Starting LA2
  - More Prolog functionality
  - Example Prolog problems

For reference: SML Type Inference Rules

- **Constants**
  - `true, false`, type is `bool`
  - Integer literals (e.g. `42`), type is `int`
- **Variable use**
  - If x : tau is in the environment, then x has type tau
- **Lambda functions `(fn x => e)`**
  - If `x : tau1` and `e : tau2`, then `fn x => e : tau1 -> tau2`
- **Function application `(e1 e2)`**
  - If `e1 : tau1 -> tau2` and `e2 : tau1`, then `e1 e2 : tau2`
- **Addition, or multiplication `(e1 + e2)`**
  - Both `e1` and `e2` must be `int`, and the result is `int`
- **If expression `(if e1 then e2 else e3)`**
  - If `e1 : bool, e2 : tau, e3 : tau,` **result is** `tau`
- **Let expression `(let val x = e1 in e2 end)`**
  - Infer `e1 : tau1`, then infer `e2` with `x : tau1` in environment

Getting started with LA2

- Type inference in SML
- Check out the test cases and the questions in the README
  - Which test cases are examples we have already done in class?
  - Which test cases are already working?
  - What are some possible next steps for doing the assignment?

AST-based approach for earlier examples

- Earlier examples

```
fun square x = x * x;
fun baz f x = f (f x);

fun pairself x = (x,x)
```

- Which of the above show up in the LA2 test cases and where?

Getting started with LA2

- Type inference in SML
- Check out the test cases and the questions in the README
  - Which test cases are examples we have already done in class?
  - Which test cases are already working?
  - What are some possible next steps for doing the assignment?
- Once you have implemented the type inference, when do we know there is a type error?
- How would we do a key, value or dictionary like data structure in Prolog?

Other Prolog predicates

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `append([1,2,3],[4,5,6,7],X).`
- `append(X,[2,4],[0,2,4]).`
- `append([1,2],X,[1,2,3,4]).`
- `append(X,Y,[1,2,3,4]).`

What does `append(X,Y,Z)` mean?

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `select(4, [1,2,4,3,5,4], X).`
- `select(1, X, [0,2,4]).`
- `select(X, [1,2,3,4], [1,3,4]).`
- `select(X, [1,2,3,4], Y).`

What does `select(X,Y,Z)` mean?

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `nth0(0,[1,2,3,4],X).`
- `nth0(1,[1,2,3,4],X).`
- `nth0(5,[1,2,3,4],X).`
- `nth0(X,[1,2,3,1,2,3],2).`
- `nth0(4,X,4).`

What does `nth0(X,Y,Z)` mean?

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `nth1(0,[1,2,3,4],X).`
- `nth1(1,[1,2,3,4],X).`
- `nth1(5,[1,2,3,4],X).`
- `nth1(X,[1,2,3,1,2,3],2).`
- `nth1(4,X,4).`

What does `nth1(X,Y,Z)` mean?

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `length([1,2,3,4],X).`
- `length([],X).`
- `length([_],X).`
- `length(X,5).`

What does `length(X,Y)` mean?

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `reverse([1,2,3,4],X).`
- `reverse(X,[0,2,4,6,7]).`
- `reverse(X,Y).`
- `reverse([2,3|X], Y).`

What does `reverse(X,Y)` mean?

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `sort([2,6,1,7,8,9,0],X).`
- `sort(X,[1,2,3,4,5]).`

(a) What does `sort(X,Y)` mean?

(b) Explain why the second query causes an error.

Exercise.

Try the following queries and report the results. Don't forget to press ";" when there are multiple answers possible.

- `member(3,[1,2,3,4,5]).`
- `member(X,[1,2,3,4,5]).`
- `member(3,X).`

What does `member(X,Y)` mean?

Exercise.

Given the definition below, what does `foo(X,Y)` mean?

```
foo(0,1).

foo(X,Y) :- X > 0, X1 is X-1, foo(X1,Y1), Y is
X*Y1.
```

Problem-solving in Prolog

Note: Answers to these will not be posted in these slides. Posting the answers in piazza is encouraged and will be rewarded with extra credit points.

Remember: Prolog is *declarative*.

Instead of thinking about *how* to find a solution, we need to think about *what* a solution looks like.

And then we let Prolog do the rest.

Other reminders

- Everything is unification (i.e. pattern matching).
- You do not have to specify when predicates should fail. You only need to specify what makes them true.

Man-Wolf-Goat-Cabbage

Problem Definition

A man has a wolf, a goat, and a cabbage. He also has a boat that can carry at most himself and one of his items at a time. He and all his items are on the west side of a river, and he needs to get them all over to the east side of the river. However, if at any point he leaves the wolf and the goat together unsupervised, the wolf will eat the goat. Similarly, if he ever leaves the goat and the cabbage together unsupervised, the goat will eat the cabbage. How can he get everyone safely across the river?

What does a solution to this problem look like?