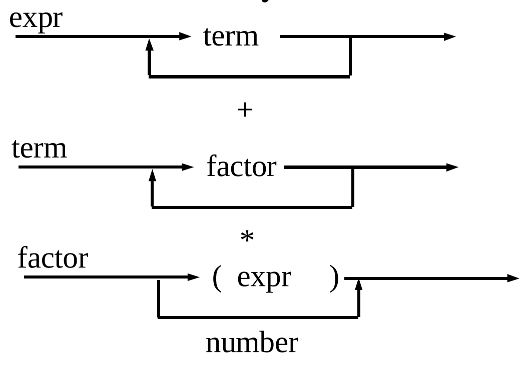
CS 330 Lecture 3

- Outline
  - Quickly finish last lecture
  - Brief overview of semantics
  - Intro to functional programming
  - Assignment I

EBNF

- <statement-list> := { <statement> }
- <st-list> := <empty>
  | <statement>; <st-list>
- Basically shorthands and metasymbols for commonly used CFG structures

Syntax chart



Parsing

- Recognizer, parser
- shift-reduce or bottom-up parsers
- top-down
- recursive decent parsing

Names and attributes

- const int n = 5;
  - name n
  - type attribute = const int
  - value attribute = 5
- double f(int n) {.....}
  - name f
  - type attribute = function of 1 int argument that returns a double
  - body attribute = the actual code

Binding

- Associate attribute to a name
- Static binding
  - Translation
  - Linking
  - Loading
- Dynamic binding

Let's think about adding variables to Moo

Symbol Table or Environment

- Function that expresses the bindings of attributes to names
- Compilers – Symbol Table
- Interpreters – Environment
- Variable dictionary
  - Insert
  - Lookup
  - Delete

Scope

- Region of program where a binding is maintained
- Let's draw some symbol tables
- Static and dynamic scoping
- Name resolution and overloading

Functional Languages

- Black box view
- Function y=f(x)   f: X->Y
  - domain X, range Y
  - x = independent variable, y = dependent variable
  - partial vs total function
- Function definition, application

Functional Languages

- Scheme, ML, Haskell
- AI, prototyping, proof-systems
- Advantages
  - Uniform view of programs as functions
  - Automatic memory management
  - Great flexibility, conciseness of notation and simple semantics
- Drawback (used to be)
  - Inefficiency

Then why people don't use them

- Persistence of established technology
- More abstract and mathematical
- Object-oriented programming mirrors everyday experience so for simple programs it is easier (that's why it doesn't work very well :-))
- Less libraries although they are catching up

Side-effects – the enemy

- In pure functional language there are no assignments only bindings.
- Referential transparency
  - Function that its value depends only on the values of its arguments
- Value semantics
- Functions are first class citizens

Assignment 1

- Postfix calculator for rational number
- 1|2 + 3|2 = 5|2
- Postfix (reverse Polish)
  - 1|2 3|2 +
- More details on the web page
- Any language you want
- Write for someone to read

Next week

- Introduction to SML with emphasis on learning important concepts in Programming Languages