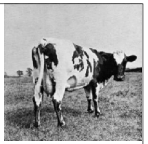
CS 330 Lecture 2

- Outline
  - Syntax = form, structure
  - Semantics = meaning

Describing Programming Languages

- Early days lengthy English and examples for both syntax and semantics
- 1950's Noam Chomsky – Context-Free Grammars for Linguistics
- Backus & Naur -> BNF diagrams
- EBNF
- Syntax diagrams

Let's invent a language for teaching arithmetic (moo)



Q: ooo + oo
A: ooooo or 5

Q: 3 + oo
A: ooooo or 5

Q: 3 + 2
A: ooooo or 5

Q: ooo + 2
A: ooooo or 5

Q: (ooo + oo) x 2
A: oooooooooo or 10

Q: (oo + oo) x oo + 2
A: oooooooooo or 10

Lexical structure

- Words = units of meaning in language
- Tokens = units of meaning in PLs
- Correspondance between the written representation of the language and the tokens in a grammar for the language

Lexical structure

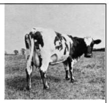
- Numbers
  - oo-format i.e oo or oooo
  - Num-format i.e 2 or 6
- Special symbols + - x
- Real PLs
  - Reserved words
  - Literals
  - Special Symbols
  - Identifiers

Algol code example

```
// the main program (this is a
comment)
begin
  integer N;
  Read Int(N);
  begin
    real array Data[1:N];
    real sum, avg;
    integer i;
    sum:=0;
    for i:=1 step 1 until N do
      begin real val;
        Read Real(val);
        Data[i]:=if val<0 then
-val else val
      end;
    for i:=1 step 1 until N do
      sum:=sum + Data[i];
    avg:=sum/N;
    Print Real(avg)
  end
end
```

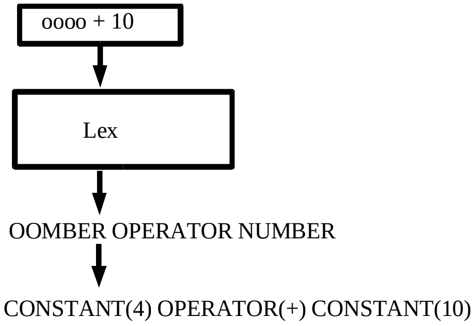
Find the reserved words, literals, special symbols and identifiers of this code

Back to Moo (longer munch + regular expr)



- OOMBER is one or more o symbols
  - o+   (why is ooooo not oo and ooo)
- NUMBER is one or more digits
  - (0|1|2|3|4|5|6|7|8|9)+
  - [0-9]+
- OPERATOR is (+|-|x)
- Regular expressions
  - concatenation, repetition, selection

Lexical Analysis



oooo + 10

Lex

OOMBER OPERATOR NUMBER

CONSTANT(4) OPERATOR(+) CONSTANT(10)

What about parentheses ?

Context-Free Grammars

- sentence -> noun-phrase verb-phrase
- noun-phrase -> article noun
- article -> a | the
- noun -> girl | dog
- verb-phrase -> verb noun-phrase
- verb -> sees pets

Some phrases (language of the grammar)

The girl sees a dog
A girl pets the dog
A dog sees a girl
The dog pets the girl ??

What about Moo ?

A grammar for Moo

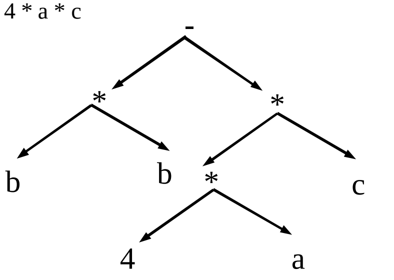
- moosing -> moosing + moosing |
  moosing * moosing |
  (moosing) |
  constant
- constant -> oomber | number
- ooomber -> oomber o | o
- number -> number digit | digit
- digit = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Expression notations

- Prefix
  - * + 20 30 60= ?
- Postfix
  - 20 30 + 60 * = ?
- Infix
  - 20 + 30 * 60 = ?\
- What about
  - if (exp) then exp else exp;

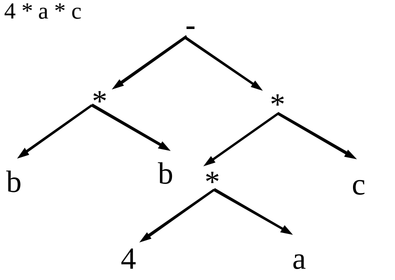
Abstract Syntax Trees

b * b – 4 * a * c



Adding semantics

b * b – 4 * a * c



a = 2
b = 3
c = 4

A grammar for Moo

- moosing -> moosing + moosing |
  moosing * moosing |
  (moosing) |
  constant
- 5 + 2 * 3 + 4 = ?
  - why does Moo answer 25 ?

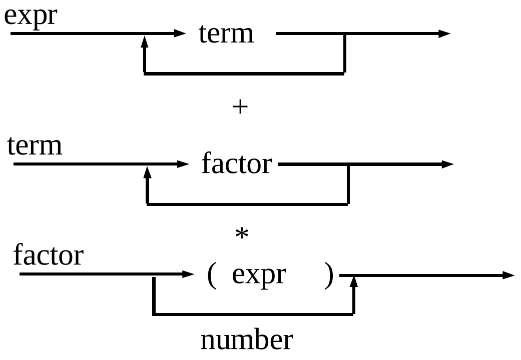
Handling precedence in Moo

- Expression is a list of moolts separated by +
- exp = exp + moolt
- moolt = moolt * const
- const = oomber | number | (e)

EBNF

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

Syntax chart



Next lecture:
More on Syntax and Semantics
+ Intro to Functional

- PLEASE don't start leaving the class when you read the title of this slide
- Recognizer, parser
- shift-reduce or bottom-up parsers
- top-down
- recursive decent parsing
- Functional programming