CS310

Parsing with Context Free Grammars

Today's reference:

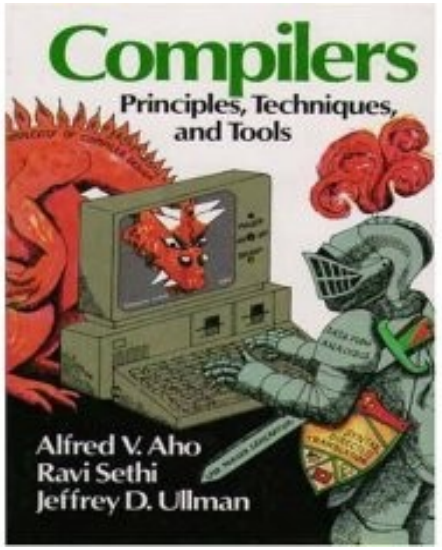
Compilers: Principles, Techniques, and Tools

by: Aho, Sethi, Ullman

aka: The Dragon Book

Section 4.4 – 4.8

Nov 3, 2010



Remove Left Recursion

- Immediate:
  $A \rightarrow Aa \mid a$

- Indirect:

  $S \rightarrow Aa \mid b$

  $A \rightarrow Sd \mid Ac \mid \varepsilon$

  Recursion: $S \rightarrow Aa \rightarrow Sda$

Algorithm (Dragon book, p 177 fig 4.7)

- Limits: no ε productions; no cycles
  Arrange non-terminals in some order: $A_1 ... A_n$

---

for i = 1 to n
    for j = 1 to i-1
        replace each production of the form
        $A_i \rightarrow A_j Y$ by the productions
        $A_i \rightarrow d_1 Y \mid d_2 Y \ldots d_k Y$
        where $A_j \rightarrow d_1 \mid d_2 \mid ... \mid d_k$ are all the
        current A_j productions
    end
    remove immediate recursion on A_i
end

Build Parse Table

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

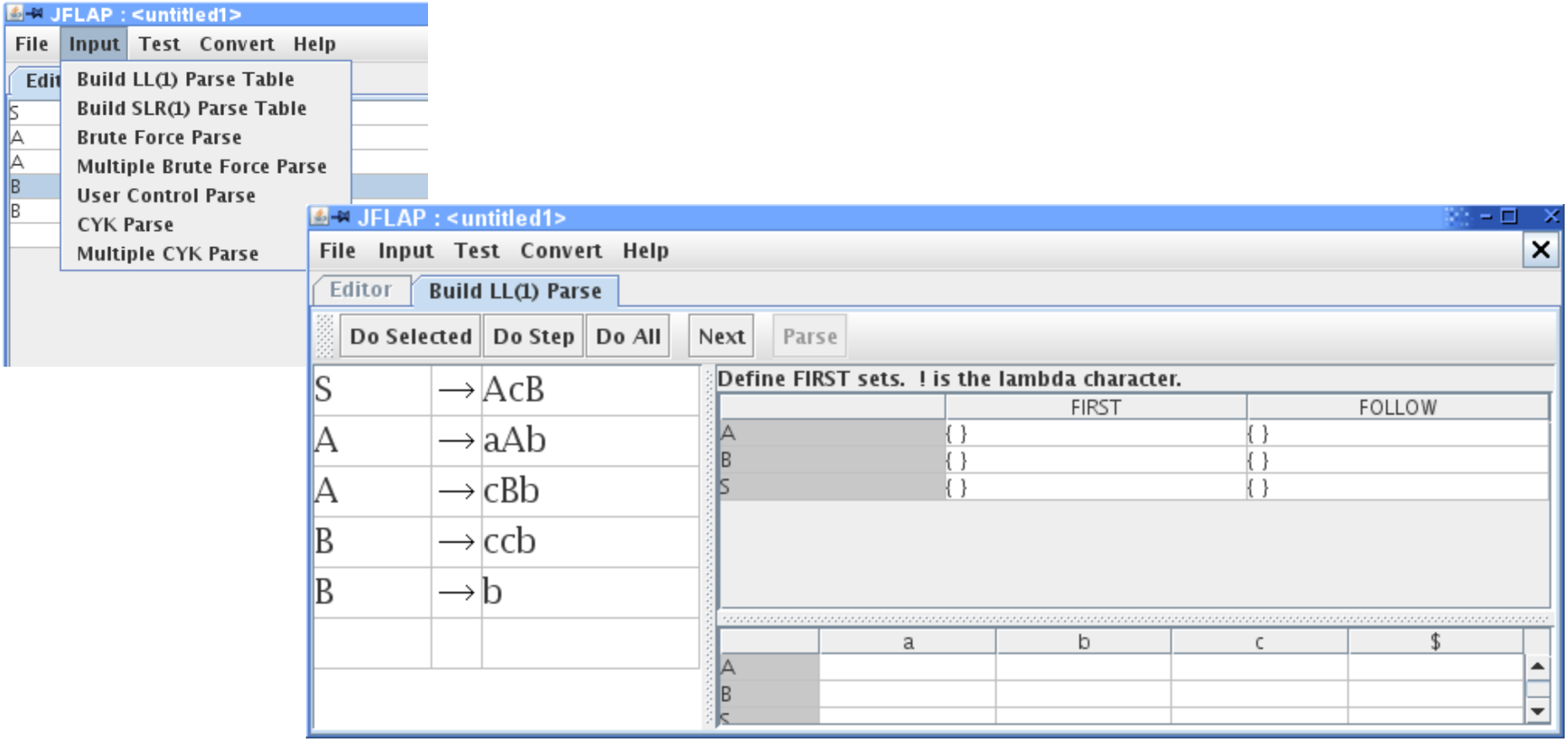
What transformations do we need to make?

What effects do these transformations have?

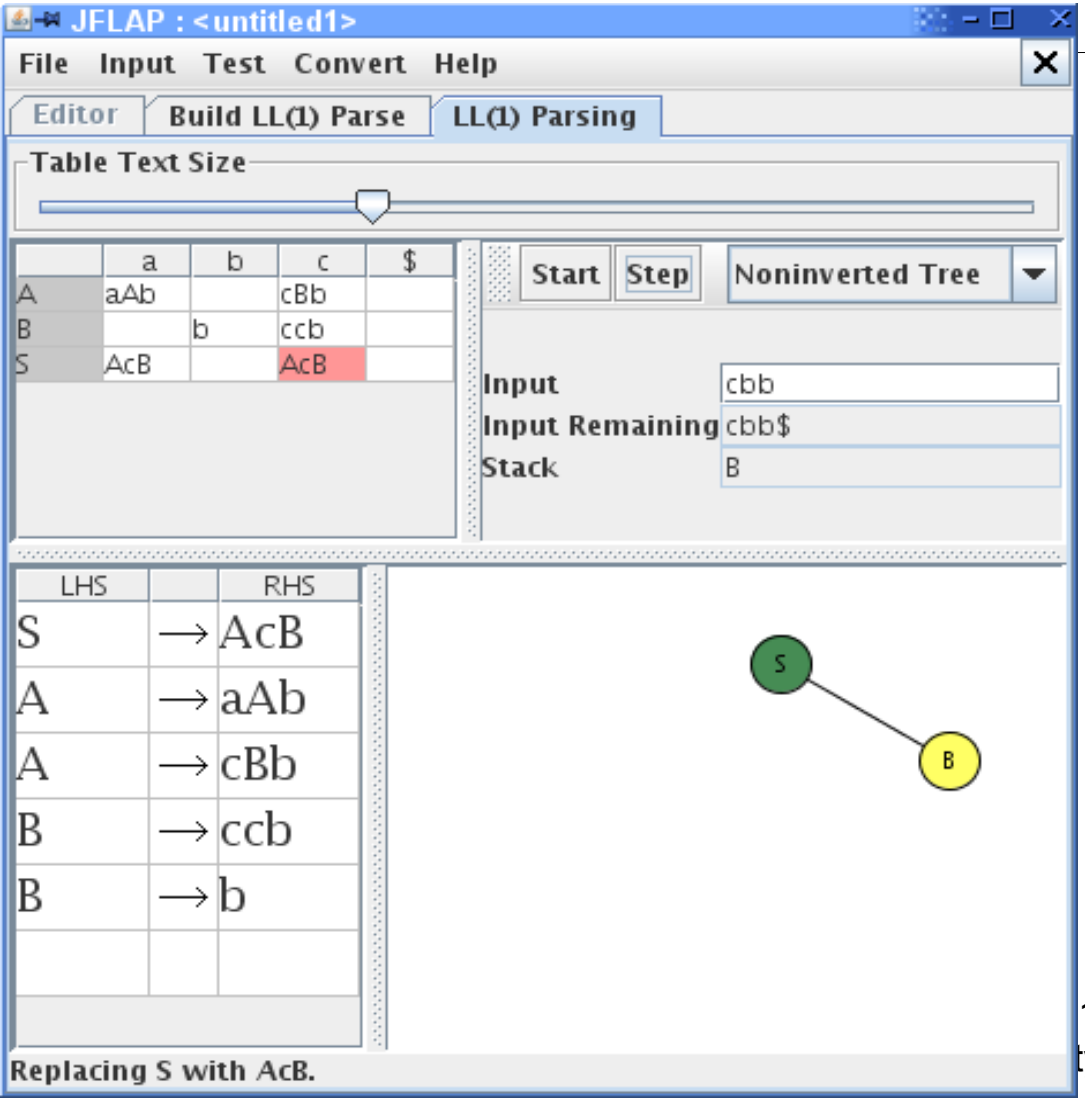
Parsing with JFLAP

- FIRST? FOLLOW? Parse Table?

(1) S -> AcB
(2) A -> aAb
(3) A -> cBb
(4) B -> ccb
(5) B -> b



Parse


JFLAP : <untitled1>
File Input Test Convert Help
Editor
Build LL(1) Parse
LL(1) Parsing
Table Text Size
a b c $
A aAb cBb
B b ccb
S AcB AcB
Start
Step
Noninverted Tree
Input cbb
Input Remaining cbb$
Stack B
LHS RHS
S → AcB
A → aAb
A → cBb
B → ccb
B → b
S
B
Replacing S with AcB.

LL(2) Parse Table

(1) S -> AcB
(2) A -> aAb
(3) A -> aBb
(4) B -> ccb
(5) B -> b

- Two lookahead symbols

	aa	ab	ac				
S							
A							

Bottom Up Parsing

(1) E -> E + E
(2) E -> E * E
(3) E -> (E)
(4) E -> x

- Shift-reduce parsing
  - used in many automatic parser generators,
- *Reduce* the string to the start symbol
- *Shift* a symbol from the string on to a stack

Stack	Input	Action
$	x + x * x $	shift
$ x	+ x * x $	reduce
$ E		

Shift/Reduce Conflict

stmt → IF expr THEN stmt
| IF expr THEN stmt ELSE stmt

Stack	Input	Action
... $ IF expr THEN stmt	ELSE .... $	???

Sipser 2.24: $E = \{a^i b^j \mid i \neq j \text{ and } 2i \neq j\}$

$L = \{w \mid a^x a^z b^z b^x ;\ x, z \geq 0\}$