

Section: LL Parsing

LL(k) Parser:

- top-down parser - starts with start symbol on stack, and repeatedly replace nonterminals until string is generated.
- predictive parser - predict next rewrite rule
- first L of LL means - read input string left to right
- second L of LL means - produces leftmost derivation
- k - number of lookahead symbols

LL parsing process:

- convert CFG to PDA (different method than before)
- Use the PDA and lookahead symbols
- Lookahead symbol is next symbol in input string

Convert CFG to NPDA

The constructed NPDA:

- Three states: s , q , f
 - start in state s
 - push S on stack, move into q
 - all rewrite rules in state q : If lhs of rewrite rule on top of stack, replace it with rhs of rewrite rule and stay in state q
 - additional rules in q to recognize terminals: read input symbol, pop input symbol, stay in state q
 - pop z from stack, move into f , accept

Example:

$$L = \{a^n b b^n : n \geq 0\}$$

```

state = s
push(S)
state = q
read(symbol)
while top-of-stack  $\neq$  z do
  case top-of-stack of
    S: if symbol == a then
        {pop(); push(aSb)}
      else if symbol == b then
        {pop(); push(b)}
      else error
    a: if symbol  $\neq$  a, then error
        else {pop(); read(symbol)}
    b: if symbol  $\neq$  b, then error
        else {pop(); read(symbol)}
  end case
end while
pop()
if symbol  $\neq$  $ then error
state = f

```

LL Parse Table - 2-dim array

- rows - variables
- cols - terminals, \$ (end of string marker)
- $LL[i,j]$

Example: Parse table for

$$L = \{a^n b b^n : n \geq 0\}$$

$$S \rightarrow aSb \mid b$$

A generic parsing routine

```
push(S)
read(symbol)
while stack not empty do
  case top-of-stack of
    terminal:
      if top-of-stack == symbol
        then {pop(); read(symbol)}
      else
        error
    variable:
      if LL[top-of-stack, symbol] ≠ error
        then {pop()
              push(LL[top-of-stack,symbol])}
      else
        error
  end case
end while
if symbol ≠ $, then error
```

Example:

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow c \end{aligned}$$

	a	b	c	\$
S	aSb	error	c	error

Example:

$$\begin{aligned} S &\rightarrow Ac \mid Bc \\ A &\rightarrow aAb \mid \lambda \\ B &\rightarrow b \end{aligned}$$

To construct an LL parse table
 $LL[\text{rows},\text{cols}]$:

1. For each rule $A \rightarrow w$
 - (a) For each a in $\text{FIRST}(w)$
add w to $LL[A,a]$
 - (b) If λ is in $\text{FIRST}(w)$
add w to $LL[A,b]$ for each b in
 $\text{FOLLOW}(A)$
2. Each undefined entry is error.

Example:

$$\begin{aligned} S &\rightarrow aSc \mid B \\ B &\rightarrow b \mid \lambda \end{aligned}$$

	FIRST	FOLLOW
S	a,b, λ	\$,c
B	b, λ	\$,c

To Compute the LL Parse Table for this example:

- For $S \rightarrow aSc$,
 $\text{FIRST}(aSc) =$
- For $S \rightarrow B$,
 $\text{FIRST}(B) = \{b, \lambda\}$
 $\text{FOLLOW}(S) = \{\$, c\}$
- For $B \rightarrow b$,
 $\text{FIRST}(b) =$
- For $B \rightarrow \lambda$
 $\text{FIRST}(\lambda) =$

LL(1) Parse Table:

	a	b	c	\$

Parse string: aacc

Trace aabcc

				a					
		a		S	S	B	b		
		S	S	c	c	c	c	c	
Stack:	<u>S</u>	<u>c</u>	<u>c</u>	<u>c</u>	<u>c</u>	<u>c</u>	<u>c</u>	<u>c</u>	
symbol:	a	a	a'	a'	b	b	b	c	c'

Example: Construct Parse Table for:

$$S \rightarrow AcB$$

$$A \rightarrow aAb$$

$$A \rightarrow \lambda$$

$$B \rightarrow aBb$$

$$B \rightarrow c$$

$$\text{FIRST}(A) =$$

$$\text{FIRST}(S) =$$

$$\text{FIRST}(B) =$$

$$\text{FOLLOW}(A) =$$

$$\text{FOLLOW}(S) =$$

$$\text{FOLLOW}(B) =$$

To compute the parse table:

- For $S \rightarrow AcB$,
 $FIRST(AcB) =$
- For $A \rightarrow aAb$,
 $FIRST(aAb) =$
- For $A \rightarrow \lambda$,
 $FIRST(\lambda) =$
- For $B \rightarrow aBb$,
 $FIRST(aBb) =$
- For $B \rightarrow c$,
 $FIRST(c) =$
- All other entries are errors.

LL(1) Parse Table:

	a	b	c	\$

Example:

$S \rightarrow AcB$

$A \rightarrow aAb \mid ab$

$B \rightarrow aBb \mid acb$

	FIRST	FOLLOW

	a	b	c	\$

LL(2) Parse Table:

	aa	ab	ac	a\$	b	c	\$
S	AcB	AcB	error	error	error	error	error
A	aAb	ab	error	error	error	error	error
B	aBb	error	acb	error	error	error	error

parse string: aabbcacb

			a		a						
			A	A	b	b					
		A	b	b	b	b	b			a	
		c	c	c	c	c	c	c		c	c
Stack:	<u>S</u>	<u>B</u>	<u>B</u>	<u>B</u>	<u>B</u>	<u>B</u>	<u>B</u>	<u>B</u>	<u>B</u>	<u>b</u>	<u>b</u>
symbol:	aa	aa	aa	ab	ab	bb	bc	ca	ac	ac	cb

Example:

$$L = \{a^n : n \geq 0\} \cup \{a^n b^n : n \geq 0\}$$

$$\mathbf{S} \rightarrow \mathbf{A}$$

$$\mathbf{S} \rightarrow \mathbf{B}$$

$$\mathbf{A} \rightarrow \mathbf{aA}$$

$$\mathbf{A} \rightarrow \lambda$$

$$\mathbf{B} \rightarrow \mathbf{aBb}$$

$$\mathbf{B} \rightarrow \lambda$$

Example:

$$L = \{a^n : 0 \leq n \leq 10\} \cup \{a^n b^n : 0 \leq n \leq 10\}$$

Example

$S \rightarrow bbCd \mid Bcc$

$B \rightarrow bB \mid b$

$C \rightarrow cC \mid c$