

# CS480

# Translators

What is Bottom-Up Parsing?

Chap. 4

# Odds and Ends

- Milestone 3 due tonight!
  - Are you allowed to change the grammar?
  - ... the language? no
- Make sure you have enough ~~test cases~~, i.e. programs.
- Milestone 4...

constants

Convince me  
that you did  
this

# Quiz #5

$$S \rightarrow [^B \text{ } ]^B | [^B S ]^B | \cancel{S'}^{\alpha} \text{expr}^B \quad \begin{cases} S \rightarrow [S' | [S S'] | \text{expr} S' \\ S' \rightarrow S S' | \epsilon \end{cases}$$

$$S \rightarrow (S) S \mid S \rightarrow \epsilon \mid S \rightarrow \epsilon$$

- Eliminate left recursion from the S production in IBTL.

- For each grammar below, calculate First and Follow sets for each nonterminal and construct a parsing table.

(a)  $S \rightarrow 0 S'$

$S' \rightarrow S 1 \mid 1$

(b)  $S \rightarrow (S) S \mid \epsilon$

$First(S) = \{0\}$   $Follow(S) = \{1, \$\}$   
 $(S') = \{0, 1\}$   $S' = \{1, \$\}$

S	$S \rightarrow 0S'$
S'	$S' \rightarrow S1 \mid S \rightarrow 1$

- What do we need to do to our grammar to use top-down parsing? Is it LL(1), LL(2), etc.?

$First(S) = \{ (, \epsilon \}$   $Follow(S) = \{ ), \$ \}$

left to right scan of input  
rightmost derivation

# Bottom-Up Parsing

CFG

$K \leq 1$

- What is it? and How is it useful?

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

id \* id

$F * \text{id}$   
|  
id

$T * \text{id}$   
|  
 $F$   
|  
id

$T * F$   
|     |  
 $F$    id  
|  
id

$T$   
/   |   \  
 $T$    \*    $F$   
|       |  
 $F$      id  
|  
id

$E$   
|  
 $T$   
/   |   \  
 $T$    \*    $F$   
|       |  
 $F$      id  
|  
id

LALR(1)  
SLR(1)  
LR(0)  
LL

Allows us  
to parse a more  
complex grammar

# Reductions

← opposite of a derivation

- Example:

id\*id, F\*id, T\*id, T\*F, T, E

Vs.

$E \Rightarrow T \Rightarrow T * F \Rightarrow T * id \Rightarrow F * id \Rightarrow id * id$

leftmost reduction

rightmost

- What do you notice?

# theory → Handles <sup>②</sup> shifting & reducing

- What is the Right Sentential Form?
- What is the handle ?

valid derivation from rightmost

RIGHT SENTENTIAL FORM	HANDLE	REDUCING PRODUCTION
① $id_1 * id_2$	$id_1$	$F \rightarrow id$
② $F * id_2$	$F$	$T \rightarrow F$
③ $T * id_2$	$id_2$	$F \rightarrow id$
$T * F$	$T * F$	$T \rightarrow T * F$
$T$	$T$	$E \rightarrow T$

Figure 4.26: Handles during a parse of  $id_1 * id_2$

what is being reduced

CFG

# Shift/Reduce Parser

- Stack – track position in parse
- Table – what to do
  - Shift – Put next input on stack.
  - Reduce – If the handle is on the stack, then reduce.
  - Accept – Successful completion.
  - Error – Discover Syntax error.

# Shift/Reduce Parser cont...

- Stack, input, and action

STACK	INPUT	ACTION
\$	<u>id<sub>1</sub> * id<sub>2</sub></u> \$	shift
\$ <u>id<sub>1</sub></u>	* id <sub>2</sub> \$	reduce by $F \rightarrow \text{id}$ ✓
\$ <u>F</u>	* id <sub>2</sub> \$	reduce by $T \rightarrow F$ ✓
\$ <u>T</u>	<u>* id<sub>2</sub></u> \$	shift ←
\$ <u>T *</u>	id <sub>2</sub> \$	shift
\$ <u>T * id<sub>2</sub></u>	\$	reduce by $F \rightarrow \text{id}$
\$ <u>T * F</u>	\$	reduce by $T \rightarrow T * F$
\$ <u>T</u>	\$	reduce by <u><math>E \rightarrow T</math></u>
\$ <u>E</u>	\$	accept

Handwritten notes and corrections:

- $E \rightarrow \bar{E} + T$
- $T \rightarrow T * F$
- $F \rightarrow (E) \text{id}$
- ? instead of  $E \rightarrow$