CS3300 - Assignment 1 CS21B040

K. Puneeth Kumar

2) Let $G = (N, \Sigma, P, S)$ be a CFG.

We want to convert G to an ε-free grammar

Let G' be the grammar (ε-free)

Let $P' = P$ be the production rules of G'.

$\Rightarrow G' = (N, \Sigma, P', S)$

Now,

i) If $A \to \alpha B \beta$ and $B \to \varepsilon$ are in P' then add $A \to \alpha\beta$ in P'.

ii) Repeat the above step until no new production rule can be added.

And remove all the ε-productions from P' (except start symbol.)

Example: $S' \to S$.

$S \to aSbS \mid bSaS \mid \varepsilon$

Adding new rules:

$\to S' \to S;\ S \to aSbS \mid bSaS \mid abS \mid aSb \mid bSa \mid baS \mid \varepsilon$.

$\to S' \to \varepsilon \mid S;\ S \to aSbS \mid bSaS \mid abS \mid aSb \mid bSa \mid baS \mid ab \mid ba \mid \varepsilon$

Remove ε-productions:

* $S' \to S \mid \varepsilon;\ S \to aSbS \mid bSaS \mid abS \mid aSb \mid bSa \mid baS \mid ab \mid ba$

Now, the grammar is ε-free.

1) a) i) \/* ((Σ – "\n")* – Σ*(*\/)Σ*)* *\/

ii) \" (Σ – \")* \"

b) (a+b)* abb (a+b)*

c) Integer → (+|-|ε) (0|[1-9][0-9]*)

Decimal → Integer . ([0-9]*)

Numeric → (Integer|Decimal) E (+|-) ([0-9]*)

d) (b+B)(e+E)(g+G)(i+I)(n+N)|(e+E)(n+N)(d+D)|
(f+F)(u+U)(n+N)(c+C)(t+T)(i+I)(o+O)(n+N)

e) [1|2|3|12|13|23|11|22|33|123|321|132|312|213|231|112|121|
211|113|131|311|221|212|122|223|232|322|331|313|133|
1332|323|233|1123|1213|2113|1312|3112|2113|3112|2131|
3121|2311|3211|2231|2123|1223|2321|3221|1223|3221|1232|
3212|1322|3122|3321|3132|1332|3231|2331|1332|2331|1323|
2313|1233|2133|1132|2213|3312]

3) b) <CFG> ::= <Rules> <CFG>
| <Rules>

<Rules> ::= <non-terminal>
<non-terminal> ::= <symbol> | <non-terminal><symbol>
<symbol> ::= ε | <terminal> | <non-terminal>

a) <RE> ::= <term> <RE> | <term>
<term> ::= <factor> <term> | <factor>
<factor> ::= <atom> | <atom> <symbol>
<atom> ::= <RE> | <char>
<char> ::= "a" | "b" | --- "z" | "0" | --- | "9" |
<symbol> ::= "*" | "+" | "^" | "$" | "?"

c) S → 0S0 | 1S1 | 2S2 | 0 | 1 | 2 | ε

d) <stmt> ::= "if" "(" <expr> ")" <stmt> "fi"
| "while" "(" <expr> ")" "do" <stmt> "od"
| <Assignment statements>

<Assignment statements> ::= <variable> "=" <expr>
<variable> ::= [a-zA-Z_][a-zA-Z0-9_]*

4)

a) S → 1S0|01   String: 110100

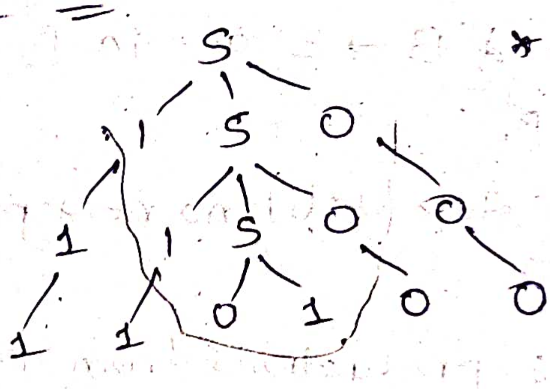
Leftmost Derivation:

S → 1S0 → 11S00 → 110100

Rightmost Derivation:

S → 1S0 → 11S00 → 110100

Parse tree:



* Grammar is unambiguous. Only one non-terminal is present. So, unambiguous.

b) S → S+S | SS | (S) | S* | a   String (a+a)*a

Leftmost Derivation:

S → SS → S*S → (S)*S → (S+S)*S → (a+S)*S → (a+a)*S → (a+a)*a

Rightmost Derivation:

S → SS → Sa → S*a → (S)*a → (S+S)*a → (S+a)*a → (a+a)*a

Grammar is ambiguous:

a+a*a has two leftmost derivations possible.

Case 1

$S \to S + S$

$S \to a + S$

$S \to a + S S$

$S \to a + S * S$

$S \to a + a * S$

$S \to a + a * a$

Case 2

$S \to S * S$

$S \to S + S * S$

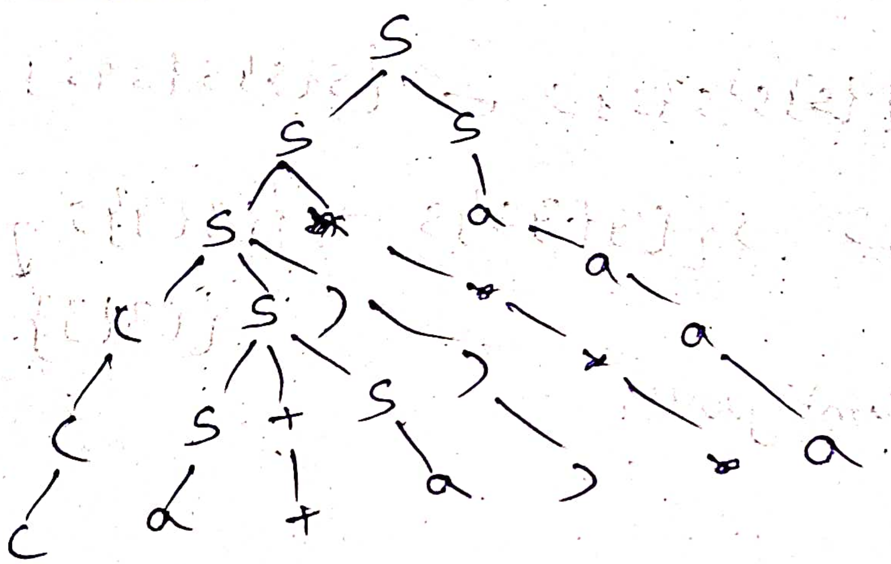
$S \to a + S * S$

$S \to a + a * S$

$S \to a + a * a$

Therefore, grammar is ambiguous.

Parse tree of (a+a)*a



c) S → S{S}S | ε   String {{}{}}

Leftmost Derivation:

S → S{S}S → {S}S → {S{S}S}S → {{S}S}S → {{}S}S → {{}S{S}S}S → {{}{S}S}S → {{}{}S}S → {{}{}}S → {{}{}}

Rightmost Derivation:

S → S{S}S → S{S} → S{S{S}S} → S{S{S}S{S}S} → S{S{S}S{S}} → S{S{S}S{}} → {{}{}}

Grammar is ambiguous, two different leftmost derivations possible for given input.

Case 1: One possibility of leftmost derivation is mentioned above.

Case 2: S → S{S}S → {S}S → {S{S}S}S

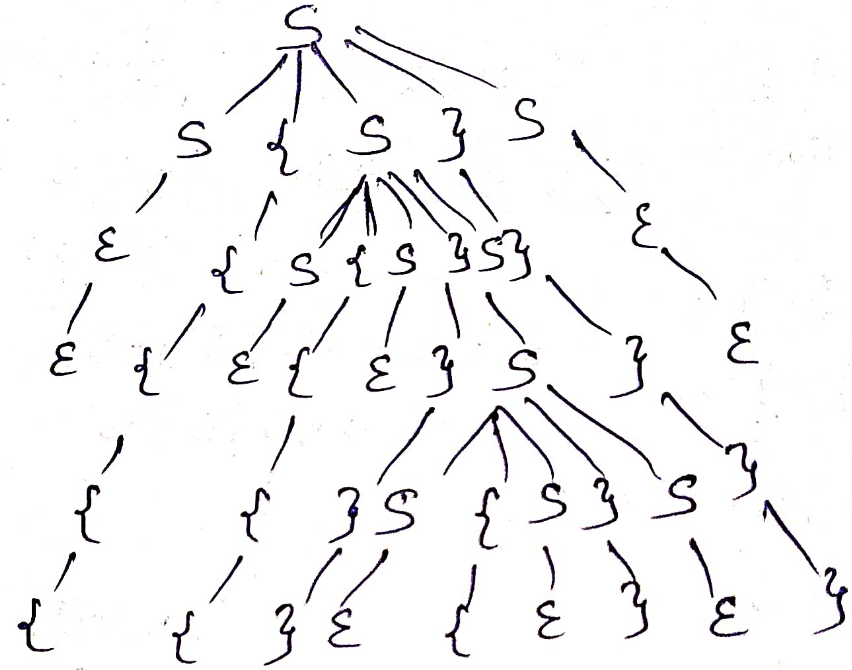
{{}S{S}S}S ← {{S}S{S}S}S ← {S{S}S{S}S}S

→ {{}{S}S}S → {{}{}S}S → {{}{}}S

{{}{}}

So, grammar is ambiguous.

Parse tree for {{}{}}



5) 1) $S' \to S\$$

2) $S \to S+S$

3) $\quad | \; SS$

4) $\quad | \; (S)$

5) $\quad | \; S*$

6) $\quad | \; a$

I_0: $S' \to \cdot S\$$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_1: $S' \to S \cdot \$$

$S \to S \cdot +S$

$\quad S \cdot S$

$\quad S \cdot *$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_2: $S' \to S\$ \cdot$

I_3: $S \to SS \cdot$

$S \to S \cdot +S$

$\quad S \cdot S$

$\quad S \cdot *$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_4: $S \to S + \cdot S$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_5:

$S \to S+S \cdot$

$S \to S \cdot +S$

$\quad S \cdot S$

$\quad S \cdot *$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_6: $S \to S* \cdot$

I_7: $S \to (\cdot S)$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_8: $S \to (S \cdot)$

$S \to S \cdot +S$

$\quad S \cdot S$

$\quad S \cdot *$

$S \to \cdot S+S$

$\quad \cdot SS$

$\quad \cdot (S)$

$\quad \cdot S*$

$\quad \cdot a$

I_9: $S \to (S) \cdot$

I_{10}: $S \to a \cdot$

Follow of S : $\{+, a, (,), *, \$\}$

NOT SLR(1) Grammar

Conflicts: I_3 [Reduce/shift Conflict]

Reason: Follow of S contains all terminals. So, shift/reduce conflict occurs. Similarly in I_5 [Shift/Reduce conflict]

ii) 1) $S' \rightarrow S\$$

2) $S \rightarrow S\{S\}S \mid \varepsilon$

Ignoring ε transition

I_0: $S' \rightarrow \cdot S\$$
$S \rightarrow \cdot S\{S\}S$

I_1: $S' \rightarrow S \cdot \$$
$S \rightarrow S \cdot \{S\}S$

I_2: $S' \rightarrow S\$ \cdot$

I_3: $S \rightarrow S\{ \cdot S\}S$
$S \rightarrow \cdot S\{S\}S$

I_4: $S \rightarrow S\{S \cdot\}S$
$S \rightarrow S \cdot \{S\}S$

I_5: $S \rightarrow S\{S\} \cdot S$
$S \rightarrow \cdot S\{S\}S$

I_6: $S \rightarrow S\{S\}S \cdot$
$S \rightarrow S \cdot \{S\}S$

follow of $S = \{ "\{", \$, "\}" \}$

$\boxed{\text{NOT SLR(1) Grammar}}$

I_6 [shift/Reduce] conflict $\rightarrow$ follow of S contains "{".

6)
1) $S' \to S\$$
2) $S \to +SS$
3) $\quad | *SS$
4) $\quad | T$
5) $T \to id$
6) $\quad | num$

I_0: $S' \to \cdot S\$$
$S \to \cdot +SS$
$S \to \cdot *SS$
$S \to \cdot T$
$T \to \cdot id$
$T \to \cdot num$

I_1: $S' \to S \cdot \$$

I_2: $S' \to S\$ \cdot$

I_3: $S \to + \cdot SS$
$S \to \cdot +SS$
$S \to \cdot *SS$
$S \to \cdot T$
$T \to \cdot id$
$T \to \cdot num$

I_4: $S \to +S \cdot S$
$S \to \cdot +SS$
$S \to \cdot *SS$
$S \to \cdot T$
$T \to \cdot id$
$T \to \cdot num$

I_5: $S \to +SS \cdot$

I_6: $S \to * \cdot SS$
$S \to \cdot +SS$
$S \to \cdot *SS$
$S \to \cdot T$
$T \to \cdot id$
$T \to \cdot num$

I_7: $S \to *S \cdot S$
$S \to \cdot +SS$
$S \to \cdot *SS$
$S \to \cdot T$
$T \to \cdot id$
$T \to \cdot num$

I_8: $S \to *SS \cdot$

I_9: $S \to T \cdot$

I_{10}: $T \to id \cdot$

I_{11}: $T \to num \cdot$

State	Action					Goto		
	id	num	+	*	$	S'	S	T
0	S10	S11	S3	S6	–	–	1	9
1	–	–	–	–	S2	–	–	–
2	acc	acc	acc	acc	acc	–	–	–
3	S10	S11	S3	S6	–	–	4	9
4	S10	S11	S3	S6	–	–	5	9
5	r2	r2	r2	r2	r2	–	–	–
6	S10	S11	S3	S6	–	–	7	9
7	S10	S11	S3	S6	–	–	8	9
8	r3	r3	r3	r3	r3	–	–	–
9	r4	r4	r4	r4	r4	–	–	–
10	r5	r5	r5	r5	r5	–	–	–
11	r6	r6	r6	r6	r6	–	–	–

* This is LR(0) grammar.

b) Input: *2+a+bc

Stack	Input	Action
$0	*2+a+bc$	S6
$06	2+a+bc$	S11
$0611	+a+bc$	r6
$069	+a+bc$	r4
$067	+a+bc$	S3
$0673	a+bc$	S10
$067310	+bc$	r5
$06739	+bc$	r4
$06734	+bc$	S3
$067343	bc$	S10
$06734310	c$	r5
$0673439	c$	r4
$0673434	c$	S10
$067343410	$	r5
$06734349	$	r4
$06734345	$	r2
$067345	$	r2
$0678	$	r3
$01	$	S2
$012	$	acc

7) Crownoise → kaa Crownoise
| kaa

Let S = Crownoise, a = kaa

1) $S' \to S \$$
2) $S \to a S$
3) $| a$

I_0: $S' \to \cdot S, \$$
$S \to \cdot aS, \$$
$S \to \cdot a, \$$

I_1: $S' \to S \cdot, \$$

I_2: $S \to a \cdot S, \$$
$S \to a \cdot, \$$
$S \to \cdot aS, \$$
$S \to \cdot a, \$$

I_3: $S \to aS \cdot, \$$

State	Action		Goto	
	a	\$	S'	S
0	S2	—	—	1
1	—	acc	—	—
2	S2	r3	—	3
3	—	r2	—	—

Stack	Input	Action
\$0	aaa\$	S2
\$02	aa\$	S2
\$022	a\$	S2
\$0222	\$	r3
\$0223	\$	r2
\$023	\$	r2
\$01	\$	acc