CS323 · Compilers

Assignment3

胡玉斌 / 11712121

October 28, 2020

1 (Grammar Basics): Consider the following context-free grammar G:

$$S \rightarrow SS + | SS * | a$$

1.1 Is the string $a + a * a$ in $L(G)$? [10points]

No. + must have at lease 2 terminal before, but it has only one.

1.2 Give a leftmost derivation for the string $aa * aa + *$. [10points]

$S \rightarrow SS* \rightarrow SS + S* \rightarrow aS + S* \rightarrow aa * S* \rightarrow aa * SS + * \rightarrow aa * aS + * \rightarrow aa * aa + *$

1.3 Give a rightmost derivation for the string $aa * aa + *$. [10points]

$S \rightarrow SS* \rightarrow SSS + * \rightarrow SSa + * \rightarrow Saa + * \rightarrow SS * aa + * \rightarrow Sa * aa + * \rightarrow aa * aa + *$

1.4 Give a parse tree for the string $aa * aa + *$. [10points]

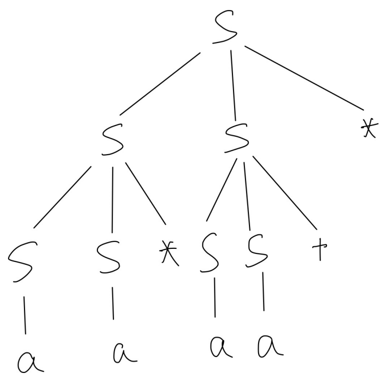

Figure 1: paser tree

1.5 Give an equivalent grammar without immediate left recursions. [10points]

$$S \rightarrow aS'$$

$$S' \rightarrow S + S' | S * S' | \epsilon$$

2 (Top-Down Parsing): Consider the following grammar G:

$$S \longrightarrow aB$$

$$B \longrightarrow S + B | \epsilon$$

2.1 Construct the predictive parsing table for G. Please put down the detailed steps, including the calculation of FIRST and FOLLOW sets. [25 points]

non-terminal = $\{S, B\}$

terminal = $\{a, +, \epsilon\}$

- FIRST sets
  - FISTR(S) = $\{a\}$
  - FIRST(B) = FIRST(S) $\cup \epsilon$ = $\{a, \epsilon\}$
- FOLLOW sets
  - FOLLOW(S) = $\{\$, +\}$
  - FOLLOW(B) = $\{\$, +\}$

	a	+	$	
S	$S \longrightarrow aB$			
B	$B \longrightarrow S + B$	$B \longrightarrow \epsilon$	$B \longrightarrow \epsilon$	

2.2 Is the grammar $LL(1)$? [5 points]

Recursive-descent parsers needing no backtracking can be constructed for a class of grammars called $LL(1)$. The grammar should hold:

- There is no terminal α such that α and β derive strings beginning with α
- At most one of α and β can derive the empty string
- If $\beta \Rightarrow \epsilon$, then does not derive any string beginning with a terminal in FOLLOW(A) and vice versa

More formally

- $FIRST(\alpha) \cap FIRST(\beta) = \emptyset$
- If $\epsilon \in FIRST(\beta)$, then $FIRST(\alpha) \cap FOLLOW(A) = \emptyset$ and vice versa

So, this grammar is $LL(1)$.

2.3 Can an $LL(1)$ parser accept the input string $aaaa+++$? If yes, please list the moves made by the parser; otherwise, state the reason. Before parsing, please resolve conflicts in the parsing table if any. [20 points]

Yes.

Matched	Stack	Input	Action
	S$	aaaa+++$	
	aB$	aaaa+++$	$S \to aB$
a	B$	aaa+++$	match a
	S+B$	aaa+++$	$B \to S + B$
	aB+B$	aaa+++$	$S \to aB$
a	B+B$	aa+++$	match a
	S+B+B$	aa+++$	$B \to S + B$
	aB+B+B$	aa+++$	$S \to aB$
a	B+B+B$	a+++$	match a
	S+B+B+B$	a+++$	$B \to S + B$
	aB+B+B+B$	a+++$	$S \to aB$
a	B+B+B+B$	+++$	match a
	+B+B+B$	+++$	$B \to \epsilon$
+	B+B+B$	++$	match +
	+B+B$	++$	$B \to \epsilon$
+	B+B$	+$	match +
	+B$	+$	$B \to \epsilon$
+	+$	$	match +
	$	$	$B \to \epsilon$
$	empty	empty	match $

3 Optional Exercise

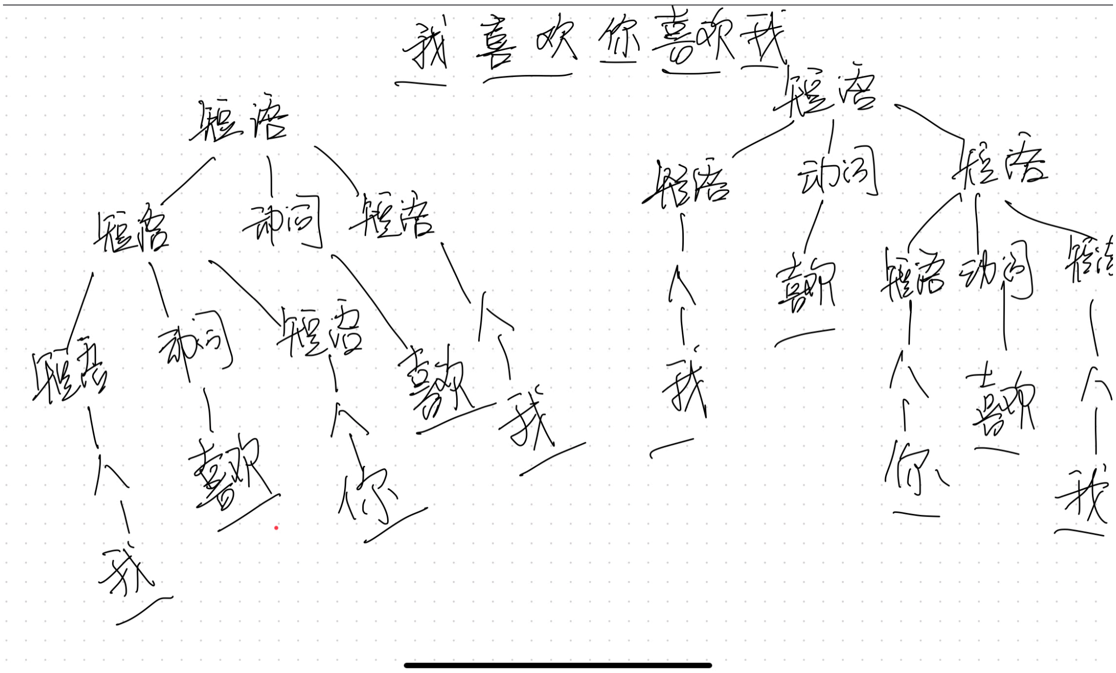

Figure 2: I like U like I