

Quiz 5

1. What are the main steps of the predictive parsing algorithm?

1. Construct the parsing table from the given grammar.
2. Apply the predictive parsing algorithm to construct the parse tree.

2. Give the parsing table construction algorithm?

Input: a grammar G

Output: the corresponding parsing table M

Method: For each production $A \rightarrow \alpha$ of the grammar do the following steps:

1. For each terminal a in $\text{FIRST}(\alpha)$, add $A \rightarrow \alpha$ to $M[A, a]$.
2. If λ in $\text{FIRST}(\alpha)$, add $A \rightarrow \alpha$ to $M[A, b]$ for each terminal b in $\text{FOLLOW}(A)$.
3. If $\lambda \in \text{FIRST}(\alpha)$ and $\$$ in $\text{FOLLOW}(A)$, add $A \rightarrow \alpha$ to $M[A, \$]$

3. Give the predictive parsing algorithm?

set ip to point to the first symbol of the input string $w\$$

repeat

 if $\text{Top}(\text{Stack})$ is a terminal or $\$$ then

 if $\text{Top}(\text{stack}) = \text{Current-Input}(ip)$ then

$\text{Pop}(\text{stack})$ and advance ip

 else null

 else if $M[X, a] = X \rightarrow Y_1 Y_2 \dots Y_k$ then

 begin

$\text{Pop}(\text{stack})$;

 Push $Y_1; Y_2; \dots; Y_k$ onto the stack, with Y_1 on top;

 Output the production $X \rightarrow Y_1 Y_2 \dots Y_k$

 end

 else null

until $\text{Top}(\text{stack}) = \$$ (i.e. the stack become empty)

4. When the predictive parsing algorithm repeat/until loop will stop?

When the stack become empty, the predictive parsing algorithm loop will stop.

5. Define LL(k) and LL(1) parsers and give two properties of LL(1) grammars?

Definition:

First L means parsing input from Left to right. Second L means making a Leftmost-derivation.

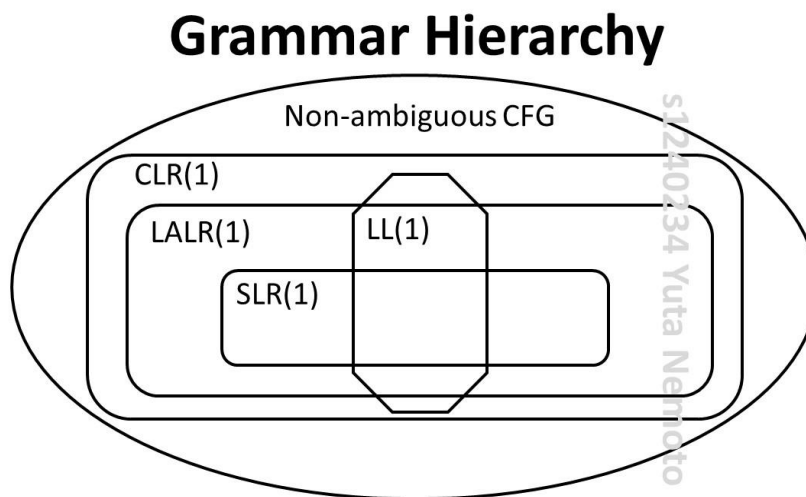
LL(k) parser looks k symbols ahead to decide its action.

LL(1) parser looks up 1 symbol ahead to choose its next action.

Properties of LL(1) grammar:

1. A grammar whose parsing table has no multiply-defined entries
2. Grammars enjoys several nice properties: for example they are not ambiguous and not left recursive.

6. Show the relationship between grammars (ie. Define the grammar hierarchy)?



7. Explain how the bottom-up parsing generally works?

It works from tokens to start-symbol

Repeat below until reducing to start-symbol:

- identify handle – reducible sequence:
 - non-terminal is not constructed but
 - all its children have been constructed
- reduce – construct non-terminal and update stack

In a grammar tree, a bottom-up parser begins at the leaves and works up to the top of the tree.

The reduction steps trace a rightmost derivation on reverse.

8. For the grammar

$E \rightarrow (E) \mid E+E$

$E \rightarrow i$

Where $i = 0, 1, 2, \dots, 9$, give the bottom-up parsing for the input $(3+2)+(1+4)$

