CS480
Translators
Top-down Parsing
Chap. 4

Monday at midnight
M3

Yes cases for good bets
Bad 15 pts

\( n = 1 \) if
Example of a non-term function w/ a production

```c
85 //T->[S] | empty
86 struct token* T(struct token *t, int depth) {
87     int i;
88     //check if token is } or empty production
89     if(t==NULL || t->tag==R_BRACKET){
90         return t; //do nothing
91     }
92     //check if token is [ for T->[S] production
93     else if(t->tag==L_BRACKET) {
94         //you want to print and add to tree!
95         for(i=depth; i>0; i--)
96             printf("\t");
97             printf("[\n");
98     //Get next token and call S production
99     t=lexer();
100     t=S(t);  
101     //Now process the ] after no more [, or we go to empty
102     if(t!=NULL && t->tag==R_BRACKET) {
103         //you want to print and add to tree!
104         for(i=depth; i>0; i--)
105             printf("\t");
106             printf("bye\n");
107             //Get next token again for nested brackets or empty
108             t=lexer();
109             return t;
110         }
111         //If we don't see a matching ] for our [
112         else {
113             printf("Error...\n");
114             exit(1);
115         }
116     }
117     //We can't match a production
118     else {
119         
```

Quiz #6

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

- Are you reading a token at a time or all at once? Pros/Cons - Know K lookahead

- $S' \rightarrow S S'$
- $S \rightarrow (S) S'$
- $S \rightarrow S A$
- $S' \rightarrow S'$
Defining an LL Grammar

• Need two definitions:
• **First** and **Follow**

Figure 4.15: Terminal $c$ is in $\text{FIRST}(A)$ and $a$ is in $\text{FOLLOW}(A)$
First(α)

• If α is any string in grammar, First(α) is set of terminals that begin strings derived from α.
• If α $\Rightarrow^* \varepsilon$, then $\varepsilon$ is in First(α).
• What does it mean if A $\rightarrow^* \alpha | \beta$, and First(α) and First(β) are disjoint?
Follow(A)

- If $A$ is a nonterminal in grammar, $\text{Follow}(A)$ is the set of terminals that can appear immediately to the right of $A$.
- If $A$ can be the rightmost symbol, then $\$$ is in $\text{Follow}(A)$.
- What is $\$$?
Compute First(X) for all symbols

- If X is a terminal, then First(X) = {X}
- If X is a nonterminal and $X \rightarrow Y_1 Y_2 ... Y_k$ is a production, then place a in First(X) if for some $i$, a is in First($Y_i$) and $\varepsilon$ is in all First($Y_1$), ..., First($Y_{i-1}$). If $\varepsilon$ is in First($Y_j$) for all $j=1, 2, ..., k$, then add $\varepsilon$ First(X).
- If $X \rightarrow \varepsilon$, then $\varepsilon$ is in First(X)
Compute Follow(A) for nonterminals

• Place $ in Follow(S), where S is start symbol and $ is the input endmarker.

• If there is a production $A \rightarrow \alpha B\beta$, then everything in First(β), except $\varepsilon$, is in Follow(B).

• If there is a production $A \rightarrow \alpha B$ or $A \rightarrow \alpha B\beta$, where $\varepsilon$ is in First(β), then everything in Follow(A) is Follow(B).
Example First and Follow

\[\begin{align*}
E & \to TE' \\
E & \to + TE' | \varepsilon \\
T & \to FT' \\
T' & \to * FT' | \varepsilon \\
F & \to (E) | id
\end{align*}\]

- First(E), First(E'), First(T), First(T'), First(F)?
- Follow(E), Follow(E'), Follow(T), Follow(T'), Follow(F)?

First(E) = \{ \varepsilon, id \}

First(T) = First(F)

First(E') = \{ \varepsilon + \varepsilon \}

First(T') = \{ \varepsilon + \varepsilon \}

Follow(E) = \{ \varepsilon, \$ \}

Follow(T) = \{ \varepsilon + \$ \}
Quiz #6

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

(a) \( S \rightarrow 0 S' \)
\( S' \rightarrow S \ 1 \ | \ 1 \)

(b) \( S \rightarrow ( S ) S \ | \ \epsilon \)