Constructing an LL(1) Parse Table
CSC 2710

Assume a context-free grammar $G = (N, \Sigma, P, S)$.

**First Sets**
Let $x$ be a string in $(N \cup \Sigma)^*$. $\text{First}(x)$ is the set of terminal symbols or $\lambda$ that can appear as a prefix of any string derivable from $x$. $\lambda \in \text{First}(x)$ if $x \Rightarrow^* \lambda$.

**Inductive/Recursive Rules:**
1. $\text{First}(\lambda) = \{ \lambda \}$
2. $\text{First}(aw) = \text{First}(a) = \{ a \}$ if $a \in \Sigma$.
3. If $A \rightarrow w_1 | w_2 | \ldots | w_n$ are rules in $P$, then $\text{First}(A) = \text{First}(w_1) \cup \text{First}(w_2) \cup \ldots \cup \text{First}(w_n)$.
4. If $w \neq \lambda$
   a. If $\lambda \not\in \text{First}(A)$, then $\text{First}(Aw) = \text{First}(A)$
   b. If $\lambda \in \text{First}(A)$, then $\text{First}(Aw) = (\text{First}(A) - \{ \lambda \}) \cup \text{First}(w)$

**Follow Sets**
Let $A \in N$. Let $B \in N$. Let $x, y \in (N \cup \Sigma)^*$. $\text{Follow}(A)$ is the set of terminal symbols that can appear after $A$ in a sentential form. Assume that $\$$ \in \Sigma$ is a special terminal symbol that does not appear in any rule in $P$. It is an end of string marker.
1. If $S$ is the start symbol, $\$$ \in \text{Follow}(S)$.
2. If $A \rightarrow xB$ is a rule in $P$, then $\text{Follow}(A) \subseteq \text{Follow}(B)$.
3. If $A \rightarrow xBy$ is a rule in $P$, then $(\text{First}(y) - \{ \lambda \}) \subseteq \text{Follow}(B)$.
4. If $A \rightarrow xBy$ is a rule in $P$ and $\lambda \in \text{First}(y)$, then $\text{Follow}(A) \subseteq \text{Follow}(B)$.

Now we can use these sets to generate our LL(1) parse table.

**LL(1) Parse Table**
Our table $T$ has one row for each nonterminal symbol, and one column for each terminal symbol (including $\$$). The entries of the table will either be empty or hold one rule. We need to consider each rule $A \rightarrow w$ in $P$. A given rule may go into multiple entries of the table.
1. For each terminal $a \in \text{First}(w)$, put $A \rightarrow w$ in $T[A,a]$.
2. If $\lambda \in \text{First}(w)$, then for each terminal $a \in \text{Follow}(A)$, put $A \rightarrow w$ in $T[A,a]$.
3. If $\lambda \in \text{First}(w)$ and $\$$ \in \text{Follow}(A)$, put $A \rightarrow w$ in $T[A,\$$].

If we ever try to put a second rule in an entry of the table, the grammar is not LL(1), and we need to attempt to fix the grammar.

**Using the LL(1) Parse Table to Parse**
Instead of parsing just $w$, we parse $w\$$$. We use a stack, and push $S$, our start symbol to begin.
// assume w$ = w[1..n]
i ← 1
create stack and push S on stack
while ((stack not empty) and (i <= n))
    top ← pop(stack)
    while top is a terminal symbol
        if top does not match w[i]
            break;
        end if
        i++
        if (stack not empty)
            top ← pop(stack)
        end if
    end while
if top is a nonterminal symbol and T[top,w[i]] has a rule
    push reverse of RHS of rule at T[top,w[i]] on stack one symbol at a time
    // if RHS is $\lambda$, don’t push anything
    // remember LIFO property of stack
else
    break;
end if
end while
if ((i == n) and (stack empty))
    accept string
else
    reject string
end if