Warm Up Problem

Consider the following Grammar

$$
\begin{array}{llll}
S' & \rightarrow & \vdash \; S \; \dashv & (0) \\
S & \rightarrow & bSd & (1) \\
S & \rightarrow & pSq & (2) \\
S & \rightarrow & C & (3) \\
C & \rightarrow & rC & (4) \\
C & \rightarrow & \epsilon & (5)
\end{array}
$$

Compute the nullable, first, follow and predict tables for non-terminal symbols of this grammar. Is the grammar LL(1)?

CS 241 Lecture 12

Top Down Parsing, First and Follow Continued

With thanks to Brad Lushman, Troy Vasiga and Kevin Lanctot

Recall

Our notation:

$$First(\beta) = \{a \in \Sigma' : \beta \Rightarrow^* a\gamma, \text{ for some } \gamma \in V^*\}$$
$$Nullable(\beta) = \text{ true iff } \beta \Rightarrow^* \epsilon \text{ and false otherwise}$$
$$Follow(A) = \{b \in \Sigma' : S' \Rightarrow^* \alpha A b \beta \text{ for some } \alpha, \beta \in V^*\}$$

Definition

We say that that a $\beta \in V^*$ is **nullable** if and only if Nullable$(\beta) =$ true.

Updated Predictor Table Definition

Definition

$$\begin{aligned}\text{Predict}(A, a) = &\{A \to \beta : a \in \text{First}(\beta)\}\\ &\cup \{A \to \beta : \beta \text{ is nullable and } a \in \text{Follow}(A)\}\end{aligned}$$

This definition is correct and is the one we want to use. Notice that this still requires that the table only have one member of the set per entry to be useful as a deterministic algorithm.

Notes on Nullable

- Note that Nullable(β) $=$ false whenever β contains a terminal symbol.
- Further, Nullable(AB) $=$ Nullable(A) $\wedge$ Nullable(B)
- Thus, it suffices to compute Nullable(A) for all $A \in N'$.

Computing Nullable

Algorithm 1 Nullable(A) for all $A \in N'$ (Simplified)

```
Initialize Nullable(A) = false for all A ∈ N'.
repeat
    for each production in P do
        if (P is B → ε) or (P is B → B_1...B_k and ⋀_{i=1}^{k} Nullable(B_i) = true) then
            Nullable(B) = true
        end if
    end for
until nothing changes
```

Computing Nullable

Algorithm 2 Nullable(A) for all $A \in N'$ (With Flags)

Initialize Nullable(A) = false for all $A \in N'$.
flag_changed = true
while flag_changed == true **do**
  flag_changed = false
  for each production in P **do**
    if (P is $B \to \epsilon$) or (P is $B \to B_1 ... B_k$ and $\bigwedge_{i=1}^{k}$ Nullable(B_i) = true) **then**
      Nullable(B) = true
      flag_changed = true
    end if
  end for
end while

Example of Nullable

$$
\begin{array}{llll}
S' & \to & \vdash S \dashv & (0) \\
S & \to & c & (1) \\
S & \to & QRS & (2) \\
Q & \to & R & (3) \\
Q & \to & d & (4) \\
R & \to & \epsilon & (5) \\
R & \to & b & (6)
\end{array}
$$

Nullability Table

Iter	0	1	2	3
S'	F	F	F	F
S	F	F	F	F
Q	F	F	T	T
R	F	T	T	T

Thus, Nullable(S') = Nullable(S) = F and Nullable(Q) = Nullable(R) = T

Notes About First

- Main idea: Keep processing $B_1 B_2 ... B_k$ from a production rule until you encounter a terminal or a symbol that is not nullable. Then go to the next rule. Repeat until no changes are made during the processing.
- For us $\epsilon \notin \text{First}(A)$ for any $A \in N'$ since $\text{First}(A) \subseteq \Sigma'$.
- For First, we will ignore **trivial productions** of the form $A \to \epsilon$ based on the above observation.
- Further, $\text{First}(S') = \{\vdash\}$ always.
- We first compute $\text{First}(A)$ for all $A \in N'$ and then we compute $\text{First}(\beta)$ for all $\beta \in V^*$

Computing First

Algorithm 3 First(A) for all $A \in N'$ (Simplified)

Initialize First(A) $= \{\}$ for all $A \in N'$.
repeat
    for each non-trivial production $A \rightarrow B_1 B_2 ... B_k$ in P **do**
        for $i \in \{1, .., k\}$ **do**
            if $B_i \in \Sigma'$ **then**
                First(A) $=$ First(A) $\cup \{B_i\}$; **break**
            else
                First(A) $=$ First(A) $\cup$ First(B_i)
                if Nullable(B_i) $==$ False **then break**
            end if
        end for
    end for
until nothing changes

Computing First

Algorithm 4 First(A) for all $A \in N'$ (With Flags)

Initialize First(A) $= \{\}$ for all $A \in N'$.
flag_changed $=$ true
while flag_changed $==$ true **do**
    flag_changed $=$ false
    for each non-trivial production $A \rightarrow B_1 B_2 ... B_k$ in P **do**
        for $i \in \{1, .., k\}$ **do**
            if $B_i \in \Sigma'$ **then**
                if First(A) $\neq$ First(A) $\cup \{B_i\}$ **then**
                    flag_changed $=$ true
                end if
                First(A) $=$ First(A) $\cup \{B_i\}$; **break**
            else
                if First(A) $\neq$ First(A) $\cup$ First(B_i) **then**
                    flag_changed $=$ true
                end if
                First(A) $=$ First(A) $\cup$ First(B_i)
                if Nullable(B_i) $==$ False **then break**
            end if
        end for
    end for
end while

Example of First

$$
\begin{aligned}
S' &\rightarrow \vdash \; S \; \dashv \\
S &\rightarrow c \\
S &\rightarrow QRS \\
Q &\rightarrow R \\
Q &\rightarrow d \\
R &\rightarrow \epsilon \\
R &\rightarrow b
\end{aligned}
$$

First Table:

Iter	0	1	2	3
S'	$\{\}$	$\{\vdash\}$	$\{\vdash\}$	$\{\vdash\}$
S	$\{\}$	$\{c\}$	$\{b, c, d\}$	$\{b, c, d\}$
Q	$\{\}$	$\{d\}$	$\{b, d\}$	$\{b, d\}$
R	$\{\}$	$\{b\}$	$\{b\}$	$\{b\}$

Recall, Nullable(S') $=$ Nullable(S) $= F$ and Nullable(Q) $=$ Nullable(R) $= T$

Hence First(S') $= \{\vdash\}$, First(S) $= \{b, c, d\}$, First(Q) $= \{b, d\}$, First(R) $= \{b\}$,

Computing First For Any $\beta \in V^*$

Algorithm 5 First(β) where $\beta = B_1 ... B_n \in V^*$

result $= \emptyset$
for $i \in \{1, .., n\}$ **do**
    if $B_i \in N'$ **then**
        result $=$ result $\cup$ First(B_i)
        if Nullable(B_i) $==$ False **then**
            break
        end if
    else
        result $=$ result $\cup$ $\{B_i\}$ (Note: $B_i \in \Sigma'$ here)
        break
    end if
end for

For example, First(RcQ) $= First(R) \cup First(cQ) = \{b, c\}$.

Note: Sometimes, this is denoted as First*.

Computing Follow (Simplified)

Algorithm 6 Follow(A) where $A \in N$ (Note: Exclude $A = S'$)

```
Initialize Follow(A) = {} for all A ∈ N.
repeat
    for each production A → B_1 B_2 ... B_k in P do
        for i ∈ {1, .., k} do
            if B_i ∈ N then
                Follow(B_i) = Follow(B_i) ∪ First(B_{i+1} ... B_k)
                if ⋀_{m=i+1}^{k} Nullable(B_m) == True or i == k then
                    Follow(B_i) = Follow(B_i) ∪ Follow(A)
                end if
            end if
        end for
    end for
until nothing changes
```

Computing Follow (With Flags)

Algorithm 7 Follow(A) where $A \in N$ (Note: Exclude $A = S'$)

```
Initialize Follow(A) = {} for all A ∈ N.
flag_changed = true
while flag_changed == true do
    flag_changed = false
    for each production A → B_1 B_2 ... B_k in P do
        for i ∈ {1, .., k} do
            if B_i ∈ N then
                if Follow(B_i) ≠ Follow(B_i) ∪ First(B_{i+1} ... B_k) then
                    flag_changed = true
                end if
                Follow(B_i) = Follow(B_i) ∪ First(B_{i+1} ... B_k)
                if ⋀_{m=i+1}^{k} Nullable(B_m) == True or i == k then
                    if Follow(B_i) ≠ Follow(B_i) ∪ Follow(A) then
                        flag_changed = true
                    end if
                    Follow(B_i) = Follow(B_i) ∪ Follow(A)
                end if
            end if
        end for
    end for
end while
```

Example of Follow

$$\begin{array}{lcl} S' & \rightarrow & \vdash S \dashv \\ S & \rightarrow & c \\ S & \rightarrow & QRS \\ Q & \rightarrow & R \\ Q & \rightarrow & d \\ R & \rightarrow & \epsilon \\ R & \rightarrow & b \end{array}$$

Follow Table:

Iter	0	1	2
S	$\{\}$	$\{\dashv\}$	$\{\dashv\}$
Q	$\{\}$	$\{b, c, d\}$	$\{b, c, d\}$
R	$\{\}$	$\{b, c, d\}$	$\{b, c, d\}$

The above makes use of the fact that

$$\text{First}(RS) = \text{First}(R) \cup \text{First}(S) = \{b, c, d\}$$

where the first equality holds since R is nullable.

Recap

$$
\begin{array}{lcll}
S' & \to & \vdash S \dashv & (0) \\
S & \to & c & (1) \\
S & \to & QRS & (2) \\
Q & \to & R & (3) \\
Q & \to & d & (4) \\
R & \to & \epsilon & (5) \\
R & \to & b & (6)
\end{array}
$$

Data:

	Nullable	First	Follow
S'	False	$\{\vdash\}$	$\{\}$
S	False	$\{b, c, d\}$	$\{\dashv\}$
Q	True	$\{b, d\}$	$\{b, c, d\}$
R	True	$\{b\}$	$\{b, c, d\}$

Definition

$$
\begin{aligned}
\text{Predict}(A, a) = \{&A \to \beta : a \in \text{First}(\beta)\} \\
&\cup \{A \to \beta : \beta \text{ is nullable and } a \in \text{Follow}(A)\}
\end{aligned}
$$

Compute the Predict Table for the grammar.

Predict Table

Definition

$$\text{Predict}(A, a) = \{A \to \beta : a \in \text{First}(\beta)\}$$
$$\cup \{A \to \beta : \beta \text{ is nullable and } a \in \text{Follow}(A)\}$$

S'	$\to$	$\vdash S \dashv$	(0)
S	$\to$	c	(1)
S	$\to$	QRS	(2)
Q	$\to$	R	(3)
Q	$\to$	d	(4)
R	$\to$	ϵ	(5)
R	$\to$	b	(6)

	$\vdash$	b	c	d	$\dashv$
S'	$\{0\}$				
S		$\{2\}$	$\{1, 2\}$	$\{2\}$	
Q		$\{3\}$	$\{3\}$	$\{3, 4\}$	
R		$\{5, 6\}$	$\{5\}$	$\{5\}$	

Cheat Sheet and Examples

Nullable:

- $A \to \epsilon$ implies that Nullable(A) $=$ true. Further Nullable(ϵ) $=$ true.
- If $A \to B_1 ... B_n$ and each of Nullable(B_i) $=$ true then Nullable(A) $=$ true.

First:

- $A \to a\alpha$ then $a \in$ First(A)
- $A \to B_1 ... B_n$ then First(A) $=$ First(A) $\cup$ First(B_i) for each $i \in \{1, .., n\}$ until Nullable(B_i) is false.

Follow:

- $A \to \alpha B \beta$ then Follow(B) $=$ First(β)
- $A \to \alpha B \beta$ and Nullable(β) $=$ true, then Follow(B) $=$ Follow(B) $\cup$ Follow(A)

$$\text{Predict}(A, a) = \{A \to \beta : a \in \text{First}(\beta)\}$$
$$\cup \{A \to \beta : \beta \text{ is nullable and } a \in \text{Follow}(A)\}$$

For More Practice

Check out

`http://smlweb.cpsc.ucalgary.ca/start.html`

Practice

Construct the four tables (Nullable, First, Follow and Predict) for the following examples:

G_1

$$
\begin{array}{llll}
S' & \rightarrow & \vdash \ S \ \dashv & (0) \\
S & \rightarrow & Bb & (1) \\
S & \rightarrow & Cd & (2) \\
B & \rightarrow & aB & (3) \\
B & \rightarrow & \epsilon & (4) \\
C & \rightarrow & cC & (5) \\
C & \rightarrow & \epsilon & (6)
\end{array}
$$

G_2

$$
\begin{array}{llll}
S' & \rightarrow & \vdash \ S \ \dashv & (0) \\
S & \rightarrow & TZ' & (1) \\
Z' & \rightarrow & +TZ' \mid \epsilon & (2,3) \\
T & \rightarrow & FT' & (4) \\
T' & \rightarrow & *FT' \mid \epsilon & (5,6) \\
F & \rightarrow & a \mid b \mid c & (7,8,9)
\end{array}
$$

For G_1

	Nullable	First	Follow
S'	False	$\{\vdash\}$	$\{\}$
S	False	$\{a, b, c, d\}$	$\{\dashv\}$
B	True	$\{a\}$	$\{b\}$
C	True	$\{c\}$	$\{d\}$

Predict

	$\vdash$	a	b	c	d	$\dashv$
S'	0					
S		1	1	2	2	
B		3	4			
C				5	6	

For G_2

	Nullable	First	Follow
S'	False	$\{\vdash\}$	$\{\}$
S	False	$\{a, b, c\}$	$\{\dashv\}$
Z'	True	$\{+\}$	$\{\dashv\}$
T	False	$\{a, b, c\}$	$\{\dashv, +\}$
T'	True	$\{*\}$	$\{\dashv, +\}$
F	False	$\{a, b, c\}$	$\{\dashv, +, *\}$

Predict (Recall: Nullable(ϵ) is true).

	$\vdash$	a	b	c	$+$	$*$	$\dashv$
S'	0						
S		1	1	1			
Z'					2		3
T		4	4	4			
T'					6	5	6
F		7	8	9			