FirstSet

- We now define the set of lookahead tokens that will cause the prediction of the production $A \rightarrow X_1 \ldots X_m$. Call the set Predict:

$$\text{Predict}(A \rightarrow X_1 \ldots X_m) =$$
$$\text{if } \lambda \in \text{First}(X_1 \ldots X_m)$$
$$(\text{First}(X_1 \ldots X_m) - \lambda) \cup \text{Follow}(A)$$
$$\text{else}$$
$$(\text{First}(X_1 \ldots X_m))$$

1	<program>	→	**begin** <statement list> **end**
2	<statement list>	→	<statement> <statement tail>
3	<statement tail>	→	<statement> <statement tail>
4	<statement tail>	→	λ
5	<statement>	→	ID:= <expression>;
6	<statement>	→	**read** (<id list>)
7	<statement>	→	**write** (<expr list>);
8	<id list>	→	ID <id tail>
9	<id tail>	→	, ID <id tail>
10	<id tail>	→	λ
11	<expr list>	→	<expression> <expr tail>

12	<expr tail>	→	,<expression> <expr tail>
13	<expr tail>	→	λ
14	<expression>	→	<primary> <primary tail>
15	<primary tail>	→	<add op> <primary> <primary tail>
16	<primary tail>	→	λ
17	<primary>	→	(<expression>)
18	<primary>	→	ID
19	<primary>	→	INTLIT
20	<add op>	→	+
21	<add op>	→	−
22	<system goal>	→	<program>$

Figure 1: A Micro Grammar in Standard Form

Non Terminal	First Set
<program>	{**begin**}
<statement list>	{ID, **read**, **write**}
<statement>	{ID, **read**, **write**}
<statement tail>	{ID, **read**, **write**, λ}
<expression>	{ID, INTLIT, (}
<id list>	{ID}
<expr list>	{ID, INTLIT, (}
<id tail>	{COMMA, λ}

<expr tail>	{COMMA, λ}
<primary>	{ID, INTLIT, (}
<primary tail>	{+, −, λ}
<add op>	{+, −}
<system goal>	{**begin**}

Figure 2: First Sets for Micro

Non Terminal	Follow Set
<program>	{**$**}
<statement list>	{**end**}
<statement>	{ID, **read**, **write**, **end**}
<statement tail>	{**end**}
<expression>	{COMMA, SEMICOLON,)}
<id list>	{)}
<expr list>	{)}
<id tail>	{)}

<expr tail>	{)}
<primary>	{COMMA, SEMICOLON, +, −,)}
<primary tail>	{COMMA, SEMICOLON,)}
<add op>	{ID, INTLIT, (}
<system goal>	{λ}

Figure 3: Follow Sets for Micro

Prod	Predict Set		
1	{**begin**}First(**begin**)= First(**begin** <statement list> **end**)=	First(**begin**)=	{**begin**}
2	First(<statement> <statement tail>)=	First(<statement>) =	{ID, **read**, **write**}
3	First(<statement> <statement tail>)=	First(<statement>) =	{ID, **read**, **write**}
4	(First(λ)− λ)∪Follow(<statement tail>) =	Follow(<statement tail>)=	{**end**}
5	First(ID <expression>;)=	First(ID)=	{ID}
6	First(**read** (<id list>);)=	First(**read**)=	{**read**}
7	First(**write** (<expr list>);)=	First(**write**)=	{**write**}
8	First(ID <id tail>)=	First(ID)=	{ID}
9	First(, ID <id tail>)=	First(,)=	{,}

Prod	Predict Set		
10	(First(λ) − λ) ∪ Follow(<id tail>) =	Follow(<id tail>) =	{)}
11	First(<expression> <expr tail>) =	First(<expression>) =	{ID, INTLIT, (}
12	First(, <expression> <expr tail>) =	First(,) =	{,}
13	(First(λ) − λ) ∪ Follow(<primary tail>) =	Follow(<expr tail>) =	{)}
14	First(<primary> <primary tail>) =	First(<primary>) =	{ID, INTLIT, (}
15	First(<add op> <primary> <primary tail>) =	First(<add op>) =	{+, −}
16	(First(λ) − λ) ∪ Follow(<primary tail>) =	Follow(<primary tail>) =	{COMMA, ; ,)}
17	First((<expression>)) =	First(() =	{(}
18	First(ID) =		{ID}

Prod	Predict Set		
19	First(INTLIT)=		{INTLIT}
20	First(+)=		{+}
21	First(−)=		{−}
22	First(<program>$)=	First(<program>) =	{**begin**}

Figure 4: Calculation of Predict Sets for Micro

NONTE R-MINAL	INPUT SYMBOL					
	id	+	*	(	)	$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow$ id			$F \rightarrow (E)$		

Figure 5: Parsing table ***M***

STACK	INPUT	OUTPUT
$E	id + id * id$	
$E'T	id + id * id$	$E \rightarrow TE'$
$E'T'F	id + id * id$	$T \rightarrow FT'$
$E'T'**id**	id + id * id$	$F \rightarrow$ id
$E'T'	+ id * id$	
$E'	+ id * id$	$T' \rightarrow \varepsilon$
$E'T+	+ id * id$	$E' \rightarrow +TE'$
$E'T	id * id$	
$E'T'F	id * id$	$T \rightarrow FT'$

$*E'T'***id**	id $*$ id$	$F \rightarrow$ id
$*E'T'*	$*$ id$	
$*E'T'F**	$*$ id$	$T' \rightarrow * FT'$
$*E'T'F*	id$	
$*E'T'***id**	id$	$F \rightarrow$ id
$*E'T'*	$	
$*E'*	$	$T' \rightarrow \varepsilon$
$	$	$E' \rightarrow \varepsilon$

Figure 6: Moves made by predictive parser on input id $+$ id $*$ id.