Parser Exercise Notes

Daniel Beatty

March 19, 2006

$$B \rightarrow Bp_1|\cdots|Bp_m|p_{m+1}|\cdots|p_{m+n}$$

Replace the rule above with

$$B \rightarrow p_m B'|\cdots|p_{m+n}B'$$
$$B' \rightarrow p_1 B'|\cdots|p_m B'|\epsilon$$

$$S \rightarrow S;S|\{S\}\ when\ C\ else\ \{S\}|\{S\}\ when\ C|\{S\}\hat{}E| := I_D E$$

$$S \rightarrow ;S'$$
$$S' \rightarrow \{S\}\ when\ C\ else\ \{S\}|\{S\}\ when\ C|\{S\}\hat{}E| := I_D E$$

Left Factoring

$$S \rightarrow ;S'$$
$$S' \rightarrow \{S\}\ when\ C\ else\ \{S\}|\{S\}\ when\ C|\{S\}\hat{}E| := I_D E$$

$$S' \rightarrow \{S''| := I_D E$$
$$S'' \rightarrow S\}\ when\ C\ else\ \{S\}|S\}\ when\ C|S\}\hat{}E$$

$$S'' \rightarrow SS'''|\epsilon$$
$$S''' \rightarrow \}\ when\ C\ else\ \{S\}|\}\ when\ C|\}\hat{}E$$

$$S''' \rightarrow \}S^4$$
$$S^4 \rightarrow\ when\ C\ else\ \{S\}|\ when\ C|\hat{}E$$

$$Sublist\ \rightarrow E|E,\ Sublist$$

$$Left\ Factor$$
$$Sublist\ \rightarrow E\ Sublist\ '$$
$$Sublist\ ' \rightarrow\ Sublist\ |\epsilon$$

Selection Sets

Production	**First**
$P \to prog(I_L)D_1\{S\}\text{-}>(I_L)$	
$D_1 \to I_L : D$	id
$\to \epsilon$	
$D \to$ *array* [*cons* $D_2]; D_1$	
$\to$ *integer* $; D_1$	*integer*
$D_2 \to ,$ *cons* D_2	,
$\to \epsilon$	
$I_L \to I_D I_{L_1}$	id
$I_{L_1} \to , I_L \mid \epsilon$	,
$I_D \to id I'_D$	id
$I'_D \to$ [*Sublist*]	[
$\to$ *epsilon*	
$E \to +E\ E$	$+$
$\to -E\ E$	$-$
$\to *E\ E$	$*$
$\to /E\ E$	$/$
$\to$ *mod* $E\ E$	*mod*
$\to I_D$	id
$\to$ *cons*	*cons*
Sublist $\to E$ *Sublist'*	$*, +, -, /,$ *cons* , *mod* , *id*
Sublist' $\to ,$ *Sublist*	, *Sublist*
$\to \epsilon$	
$S \to$:=id$E; S$	:=
$\to \{S\}S^3$	{
$\to \epsilon$	
$S' \to$ *else* $\{S\}; S$	*else*
$\to ; S$	;
$S'' \to C; S$	$<, <=, >, >=,$ *and* , *or* , *not* , *eq* , *neq*
$\to E; S$	$*, +, -, /,$ *cons* , *mod* , *id*
$S^3 \to$ *when* CS'	*when*
ˆ when S''	ˆ
$C \to < E\ E$	$<$
$\to > E\ E$	$>$
$\to$ *eq* $E\ E$	*eq*
$\to <= E\ E$	$<=$
$\to >= E\ E$	$>=$
$\to$ *neq* $E\ E$	*neq*
$\to$ *and* $C\ C$	*and*
$\to$ *or* $C\ C$	*or*
$\to$ *not* E	*not*

The follow sets can be written as the following set equations:

$$\begin{aligned}
P &= \$ \\
D_1 &\underline{\cup} D \\
D_2 &=] \\
D &= \cup D_1 \\
I_L &= : \\
I_{L_1} &= I_L \\
I_D &= , \cup I_L \cup E \\
I'_D &= I_D \\
E &= \mathit{first}(E) \cup ; \cup \mathit{Sublist} \cup C \\
C &= ;\ \mathit{else}\ \cup\ \mathit{first}\ (C) \\
S &= \} \cup S' \cup S'' \\
S' &= S^3 \\
S'' &= S^3 \\
S^3 &\underline{\cup} S \\
\mathit{Sublist} &=]\} \cup \mathit{Sublist}' \\
\mathit{Sublist}' &= \mathit{Sublist}
\end{aligned}$$

Selection Sets

Production	**First**	**Follow**
$P \rightarrow$ *prog*$(I_L)D_1\{S\}$- $>$ (I_L)	*prog*	$
$D_1 \rightarrow I_L : D$	*id*	{
$\rightarrow \epsilon$		
$D \rightarrow$ *array* [*cons* , D_2]; D_1	*array*	{
$\rightarrow$ *integer* ; D_1	*integer*	
$D_2 \rightarrow$, *cons* D_2	,	]
$\rightarrow \epsilon$		
$I_L \rightarrow I_D I_{L_1}$	*id*	:,)
$I_{L_1} \rightarrow, I_L \mid \epsilon$	,	:,)
$I_D \rightarrow idI'_D$	*id*	,) :
$I'_D \rightarrow$ [*Sublist*]	[	,) :
$\rightarrow$ *epsilon*		
$E \rightarrow +E\ E$	$+$	; , $+, -, *, /$
$\rightarrow -E\ E$	$-$	*mod* , *id*
$\rightarrow *E\ E$	$*$	*else* , $<, >$
$\rightarrow /E\ E$	$/$	$<=, >=,$
$\rightarrow$ *mod* $E\ E$	*mod*	*eq* , *neq* ,
$\rightarrow I_D$	*id*	*and* , *or* , *not*
$\rightarrow$ *cons*	*cons*	]','
Sublist $\rightarrow E$ *Sublist*$'$	$*, +, -, /$, *cons* , *mod* , *id*	]
Sublist$'$ $\rightarrow$, *Sublist*	, *Sublist*	]
$\rightarrow \epsilon$		
$S \rightarrow$:=id$E; S$	:=	}
$\rightarrow \{S\}S^3$	{	
$\rightarrow \epsilon$		
$S' \rightarrow$ *else* $\{S\}; S$	*else*	}
$\rightarrow; S$	;	
$S'' \rightarrow C; S$	$<, <=, >, >=$, *and* , *or* , *not* , *eq* , *neq*	}
$\rightarrow E; S$	$*, +, -, /$, *cons* , *mod* , *id*	
$S^3 \rightarrow$ *when* CS'	*when*	}
ˆ *when* S''	ˆ	
$C \rightarrow< E\ E$	$<$	; *else*
$\rightarrow> E\ E$	$>$	$<, >$, *eq* , *neq*
$\rightarrow$ *eq* $E\ E$	*eq*	$<=, >=$, *and*
$\rightarrow<= E\ E$	$<=$	*not* , *or*
$\rightarrow>= E\ E$	$>=$	
$\rightarrow$ *neq* $E\ E$	*neq*	
$\rightarrow$ *and* $C\ C$	*and*	
$\rightarrow$ *or* $C\ C$	*or*	
$\rightarrow$ *not* E	*not*	

Problems I'_L, S^3, The constants and "var" have issues

Error Types:

```
1  + x
```

This line will be a syntax error for getting to the end of the program in the middle of a non-final production.

LL(1) compiler types are left to right input (scan), left most derivation and the one indicates a one symbol look ahead. In such a compiler, an error can leave the compiler in a dead state. Error recovery is an issue for all compilers, but is peculiar for LL(1).

Error recovery for multiple errors, one error does not prevent the compiler from discovering the rest of the errors. A scan set is important to bring the compiler to determine which symbol is the start to the next stable location in the program. The follow set yields values for the scan set. In the case of the scan set, the tokenizer is advanced to a member of the scan set. On the scan set element in the tokenizer, the syntax analyzer is reset to continue in the productions.

In the case of errors in syntax analysis, the processing does allow semantic analysis to commence.

Unfortunately, the scan set / error recover method can be costly. Usually compilers of today report the first error of a single line statement. Compound statements may allow for the error be found within the statement.