Introduction to Language Theory and Compilation Exercises

Session 6: First sets, Follow sets and LL(1) parsing

Reminders

$First^k$ sets construction algorithm

```
\begin{array}{|c|c|c|} \textbf{begin} \\ & \textbf{foreach } a \in T \textbf{ do } \operatorname{First}^k(a) \leftarrow \{a\} \; ; \\ & \textbf{foreach } A \in V \textbf{ do } \operatorname{First}^k(A) \leftarrow \emptyset \; ; \\ & \textbf{repeat} \\ & \textbf{ foreach } A \in V \textbf{ do} \\ & & \text{ } \operatorname{First}^k(A) \leftarrow \operatorname{First}^k(A) \cup \{x \in T^* \mid A \rightarrow Y_1 Y_2 \dots Y_n \land x \in \\ & & \text{ } \operatorname{First}^k(Y_1) \oplus^k \operatorname{First}^k(Y_2) \oplus^k \dots \oplus^k \operatorname{First}^k(Y_n)\} \; ; \\ & \textbf{ until } \textit{stability} ; \end{array}
```

$Follow^k$ sets construction algorithm

```
\begin{array}{c|c} \textbf{begin} \\ & \textbf{foreach}\ A \in V \setminus \{S\}\ \textbf{do}\ \mathrm{Follow}^k(A) \leftarrow \emptyset\ ; \\ & \mathrm{Follow}^k(S) \leftarrow \{\varepsilon\}\ ; \\ & \textbf{repeat} \\ & \textbf{if}\ B \rightarrow \alpha A \beta \in P\ (with\ B \in V\ and\ \alpha, \beta \in (V \cup T)^*)\ \textbf{then} \\ & & L \ \mathrm{Follow}^k(A) \leftarrow \mathrm{Follow}^k(A) \cup \{\mathrm{First}^k(\beta) \oplus^k \mathrm{Follow}^k(B)\}\ ; \\ & \textbf{until}\ stability; \end{array}
```

Action table construction algorithm

```
\begin{array}{c|c} \mathbf{begin} \\ \hline M \leftarrow \times ; \\ \mathbf{foreach} \ A \rightarrow \alpha \ \mathbf{do} \\ \hline & \mathbf{foreach} \ a \in First^1(\alpha) \ \mathbf{do} \\ \hline & M[A,a] \leftarrow M[A,a] \cup \mathrm{Produce}(A \rightarrow \alpha) \ ; \\ \mathbf{if} \ \varepsilon \in First^1(\alpha) \ \mathbf{then} \\ \hline & \mathbf{foreach} \ a \in Follow^1(A) \ \mathbf{do} \\ \hline & M[A,a] \leftarrow M[A,a] \cup \mathrm{Produce}(A \rightarrow \alpha) \ ; \\ \hline & \mathbf{foreach} \ a \in T \ \mathbf{do} \ M[a,a] \leftarrow \mathrm{Match} \ ; \\ \hline & M[\$, \varepsilon] \leftarrow \mathrm{Accept} \ ; \\ \hline \end{array}
```

Exercises

- Ex. 1. With regards to the grammar given by Figure 1:
 - 1. Give the First $^{1}(A)$ and the Follow $^{1}(A)$ sets for each $A \in V$.
 - 2. Give the First²(<expression>) and the Follow²(<expression>) sets.

```
(1)
       <S>
                                 program> $
(2)
       cprogram>
                                begin <statement list> end
       <statement list>
(3)
                                <statement> <statement tail>
(4)
       <statement tail>
                                <statement> <statement tail>
(5)
       <statement tail>
(6)
       <statement>
                                ID := <expression> ;
                                read ( <id list> );
(7)
       <statement>
(8)
       <statement>
                                 write ( <expr list> );
(9)
                                ID <id tail>
       <id list>
(10)
       <id tail>
                                , ID <id tail>
(11)
       <id tail>
       <expr list>
(12)
                                <expression> <expr tail>
(13)
       <expr tail>
                                , <expression> <expr tail>
(14)
       <expr tail>
(15)
       <expression>
                                <primary> <primary tail>
(16)
       cprimary tail>
                                 <add op> <primary> <primary tail>
(17)
       primary tail>
                                ( <expression> )
(18)
       primary>
(19)
       cprimary>
                                ID
(20)
                                INTLIT
       primary>
(21)
       <add op>
(22)
       <add op>
```

Figure 1: Grammar for exercises 1 and 4

Ex. 2. Which grammars in Figure 2 are LL(1)?

S	\rightarrow	ABBA	$S \rightarrow$	aSe	S	\rightarrow	ABc	S	\rightarrow	Ab
\boldsymbol{A}	\rightarrow	a		B	\boldsymbol{A}	\rightarrow	a	\boldsymbol{A}	\rightarrow	a
		ε	$B \rightarrow$	bBe			ε			B
B	\rightarrow	b		C	B	\rightarrow	b			ε
		ε	$C \rightarrow$	cCe			ε	B	\rightarrow	b
				d						ϵ
	(a)	(b)	l		(c)			(d)	

Figure 2: Grammars for exercise 2

Ex. 3. Give the action table for the following grammar:

```
<S>
                           <expr> $
(2)
      <expr>
                           - <expr>
(3)
      <expr>
                          ( <expr> )
(4)
      <expr>
                           <var> <expr-tail>
(5)
      <expr-tail>
                           - <expr>
(6)
      <expr-tail>
(7)
      <var>
                          ID <var-tail>
      <var-tail>
(8)
                          ( <expr> )
(9)
      <var-tail>
```

Ex. 4. Program a recursive descent parser (in Java, C, C++, ...) for rules (15) through (22) of the grammar given by Figure 1.