Genial Imperative Language for Learning and the Enlightenment of Students
Introduction to language theory and compiling Project – Solution of Part 2

Gilles GEERAERTS   Mrudula BALACHANDER   Mathieu SASSOLAS

Info-F403 – 2024-2025

The grammar is transformed as follows:

- Eliminate rule <Instruction> → <Call> as <Call> is an unproductive nonterminal.
- Separate rules for <ExprArith> according to operators precedence. The <Op> nonterminal has been removed in the process. Then transform these to remove left-recursion.
- Separate rules for <Cond> according to operators precedence (in this case there is only grouping between |s and ->), then remove left-recursion. We also ensured that -> is right-associative with rule 26: in the produced parsing tree, the right hand-side of an implication will be in a <Cond> subtree.

The resulting grammar is given in Figure 1. We can show that this grammar is $LL1$ by showing it is *Strong* $LL1$. To do this we compute for each rule $\alpha \to \beta$ the set $\text{First}^1(\beta \cdot \text{Follow}^1(\alpha))$ (see Table 2). Then for each pair of rules with the same left handside $\alpha \to \beta$ and $\alpha \to \beta'$ we check that

$$\text{First}^1(\beta \cdot \text{Follow}^1(\alpha)) \cap \text{First}^1(\beta' \cdot \text{Follow}^1(\alpha)) = \emptyset.$$

As we see, there is no conflict. Indeed, we can use these to fill up the $LL1$ action table of Table 3.

[1]	<Program>	→ LET [ProgName] BE <Code> END
[2]	<Code>	→ <Instruction> : <Code>
[3]		→ ε
[4]	<Instruction>	→ <Assign>
[5]		→ <If>
[6]		→ <While>
[7]		→ <Output>
[8]		→ <Input>
[9]	<Assign>	→ [VarName] = <ExprArith>
[10]	<ExprArith>	→ <Prod> <ExprArith'>
[11]	<ExprArith'>	→ + <Prod> <ExprArith'>
[12]		→ - <Prod> <ExprArith'>
[13]		→ ε
[14]	<Prod>	→ <Atom> <Prod'>
[15]	<Prod'>	→ ⋆ <Atom> <Prod'>
[16]		→ / <Atom> <Prod'>
[17]		→ ε
[18]	<Atom>	→ [VarName]
[19]		→ [Number]
[20]		→ (<ExprArith>)
[21]		→ - <Atom>
[22]	<If>	→ IF { <Cond> } THEN <Code> <IfTail>
[23]	<IfTail>	→ END
[24]		→ ELSE <Code> END
[25]	<Cond>	→ <SimpleCond> <Cond'>
[26]	<Cond'>	→ -> <Cond>
[27]		→ ε
[28]	<SimpleCond>	→ |<Cond>|
[29]		→ <ExprArith> <Comp> <ExprArith>
[30]	<Comp>	→ ==
[31]		→ <=
[32]		→ <
[33]	<While>	→ WHILE {<Cond>} REPEAT <Code> END
[34]	<Output>	→ OUT([VarName])
[35]	<Input>	→ IN([VarName])

Table 1: The GILLES modified grammar.

$$\mathrm{First}^1(\texttt{LET}\ \text{[ProgName]}\ \texttt{BE}\text{<Code>}\texttt{END}\ \mathrm{Follow}^1(\text{<Program>})) = \{\texttt{LET}\} \tag{1}$$

$$\mathrm{First}^1(\text{<Instruction>:<Code>}\ \mathrm{Follow}^1(\text{<Code>})) = \{\text{[VarName]}, \texttt{IF}, \texttt{WHILE}, \texttt{OUT}, \texttt{IN}\} \tag{2}$$

$$\mathrm{First}^1(\varepsilon\ \mathrm{Follow}^1(\text{<Code>})) = \{\texttt{END}, \texttt{ELSE}\} \tag{3}$$

$$\mathrm{First}^1(\text{<Assign>}\ \mathrm{Follow}^1(\text{<Instruction>})) = \{\text{[VarName]}\} \tag{4}$$

$$\mathrm{First}^1(\text{<If>}\ \mathrm{Follow}^1(\text{<Instruction>})) = \{\texttt{IF}\} \tag{5}$$

$$\mathrm{First}^1(\text{<While>}\ \mathrm{Follow}^1(\text{<Instruction>})) = \{\texttt{WHILE}\} \tag{6}$$

$$\mathrm{First}^1(\text{<Output>}\ \mathrm{Follow}^1(\text{<Instruction>})) = \{\texttt{OUT}\} \tag{7}$$

$$\mathrm{First}^1(\text{<Input>}\ \mathrm{Follow}^1(\text{<Instruction>})) = \{\texttt{IN}\} \tag{8}$$

$$\mathrm{First}^1(\text{[VarName]=<ExprArith>}\ \mathrm{Follow}^1(\text{<Assign>})) = \{\text{[VarName]}\} \tag{9}$$

$$\mathrm{First}^1(\text{<Prod><ExprArith'>}\ \mathrm{Follow}^1(\text{<ExprArith>})) = \{\text{[VarName]}, \text{[Number]}, \texttt{(}, \texttt{-}\} \tag{10}$$

$$\mathrm{First}^1(\text{+<Prod><ExprArith'>}\ \mathrm{Follow}^1(\text{<ExprArith'>})) = \{+\} \tag{11}$$

$$\mathrm{First}^1(-\text{<Prod><ExprArith'>}\ \mathrm{Follow}^1(\text{<ExprArith'>})) = \{-\} \tag{12}$$

$$\mathrm{First}^1(\varepsilon\ \mathrm{Follow}^1(\text{<ExprArith'>})) = \{\texttt{:}, \texttt{)}, ==, <=, <, \texttt{|}, \texttt{->}, \texttt{\}}\} \tag{13}$$

$$\mathrm{First}^1(\text{<Atom><Prod'>}\ \mathrm{Follow}^1(\text{<Prod>})) = \{\text{[VarName]}, \text{[Number]}, \texttt{(}, \texttt{-}\} \tag{14}$$

$$\mathrm{First}^1(\star\text{<Atom><Prod'>}\ \mathrm{Follow}^1(\text{<Prod'>})) = \{\star\} \tag{15}$$

$$\mathrm{First}^1(/\text{<Atom><Prod'>}\ \mathrm{Follow}^1(\text{<Prod'>})) = \{/\} \tag{16}$$

$$\mathrm{First}^1(\varepsilon\ \mathrm{Follow}^1(\text{<Prod'>})) = \{+, -, \texttt{:}, \texttt{)}, ==, <=, <, \texttt{|}, \texttt{->}, \texttt{\}}\} \tag{17}$$

$$\mathrm{First}^1(\text{[VarName]}\ \mathrm{Follow}^1(\text{<Atom>})) = \{\text{[VarName]}\} \tag{18}$$

$$\mathrm{First}^1(\text{[Number]}\ \mathrm{Follow}^1(\text{<Atom>})) = \{\text{[Number]}\} \tag{19}$$

$$\mathrm{First}^1(\texttt{(}\text{<ExprArith>}\texttt{)}\ \mathrm{Follow}^1(\text{<Atom>})) = \{\texttt{(}\} \tag{20}$$

$$\mathrm{First}^1(-\text{<Atom>}\ \mathrm{Follow}^1(\text{<Atom>})) = \{-\} \tag{21}$$

$$\mathrm{First}^1(\texttt{IF\{}\text{<Cond>}\texttt{\}THEN}\text{<Code><IfTail>}\ \mathrm{Follow}^1(\text{<If>})) = \{\texttt{IF}\} \tag{22}$$

$$\mathrm{First}^1(\texttt{END}\ \mathrm{Follow}^1(\text{<IfTail>})) = \{\texttt{END}\} \tag{23}$$

$$\mathrm{First}^1(\texttt{ELSE}\text{<Code>}\texttt{END}\ \mathrm{Follow}^1(\text{<IfTail>})) = \{\texttt{ELSE}\} \tag{24}$$

$$\mathrm{First}^1(\text{<SimpleCond><Cond'>}\ \mathrm{Follow}^1(\text{<Cond>})) = \{\texttt{|}, \text{[VarName]}, \text{[Number]}, \texttt{(}, \texttt{-}\} \tag{25}$$

$$\mathrm{First}^1(\texttt{->}\text{<Cond>}\ \mathrm{Follow}^1(\text{<Cond'>})) = \{\texttt{->}\} \tag{26}$$

$$\mathrm{First}^1(\varepsilon\ \mathrm{Follow}^1(\text{<Cond'>})) = \{\texttt{\}}, \texttt{|}\} \tag{27}$$

$$\mathrm{First}^1(\texttt{|}\text{<Cond>}\texttt{|}\ \mathrm{Follow}^1(\text{<SimpleCond>})) = \{\texttt{|}\} \tag{28}$$

$$\mathrm{First}^1(\text{<ExprArith><Comp><ExprArith>}\ \mathrm{Follow}^1(\text{<SimpleCond>})) = \{\text{[VarName]}, \text{[Number]}, \texttt{(}, \texttt{-}\} \tag{29}$$

$$\mathrm{First}^1(==\ \mathrm{Follow}^1(\text{<Comp>})) = \{==\} \tag{30}$$

$$\mathrm{First}^1(<=\ \mathrm{Follow}^1(\text{<Comp>})) = \{<=\} \tag{31}$$

$$\mathrm{First}^1(<\ \mathrm{Follow}^1(\text{<Comp>})) = \{<\} \tag{32}$$

$$\mathrm{First}^1(\texttt{WHILE\{}\text{<Cond>}\texttt{\}REPEAT}\text{<Code>}\texttt{END}\ \mathrm{Follow}^1(\text{<While>})) = \{\texttt{WHILE}\} \tag{33}$$

$$\mathrm{First}^1(\texttt{OUT(}\text{[VarName]}\texttt{)}\ \mathrm{Follow}^1(\text{<Output>})) = \{\texttt{OUT}\} \tag{34}$$

$$\mathrm{First}^1(\texttt{IN(}\text{[VarName]}\texttt{)}\ \mathrm{Follow}^1(\text{<Output>})) = \{\texttt{IN}\} \tag{35}$$

Table 2: Sets $\mathrm{First}^1(\beta \cdot \mathrm{Follow}^1(\alpha))$ for all rules $\alpha \to \beta$.

	LET	BE	END	IF	THEN	ELSE	WHILE	REPEAT	OUT	IN	:	=	+	−	*	/	(	)	{	}	==	<=	<	−>	|	[ProgName]	[VarName]	[Number]
<Program>	1																											
<Code>			3	2		3	2		2	2																	2	
<Instruction>				5			6		7	8																	4	
<Assign>																											9	
<ExprArith>														10			10										10	10
<ExprArith’>											13		11	12				13		13	13	13	13	13	13			
<Prod>														14			14										14	14
<Prod’>											17		17	17	15	16		17		17	17	17	17	17	17			
<Atom>														21			20										18	19
<If>				22																								
<IfTail>			23			24																						
<Cond>														25			25								25		25	25
<Cond’>																				27				26	27			
<SimpleCond>														29			29								28		29	29
<Comp>																					30	31	32					
<While>							33																					
<Output>									34																			
<Input>									35																			

Table 3: Action table