Chukwuemeka Ajaero, 8593168

Chukwuemeka Ajaero, 8593168

Exercise 1

<program> --> begin <statement_list> end
<statement_list> --> <statement> ; <statement_list>
<statement_list> --> <statement>
<statement> --> id = <expression>
<expression> --> <factor> + <factor>
<expression> --> <factor> - <factor>
<expression> --> <factor>
<factor> --> id | num

① Eliminating Left recursion

The grammar does not contain left recursion as none of the grammar fragments are in the form $A \rightarrow A\alpha \mid \beta$.

② Left factoring

After left factoring, you get the following grammar

<program> --> begin <statement_list> end
<statement_list> --> <statement> <statement_list'>
<statement_list'> --> ; <statement_list> | ε
<statement> --> id = <expression>
<expression> --> <factor><expression'>
<expression'> --> + <factor> | - <factor> | ε
<factor> --> id | num

Chukwuemeka Ajaero, 8593168

Chukwuemeka Ajaero, 8593168

Exercise 3

1. <program> → begin <statement_list> end
<statement_list> → <statement> <statement_list'>
<statement_list'> → ; <statement_list> | ε
<statement> → id = <expression>
<expression> → <factor><expression'>
<expression'> → +<factor> | -<factor> | ε
<factor> → id | num

- FIRST(program) = { begin }
- FIRST(statement_list) = { FIRST (statement) } = { id,= }
- FIRST(statement_list') = { ; , ε } Need to ask whether this should be one terminal or two
- FIRST(statement) = { id,= }
- FIRST(expression) = { FIRST (factor) } = { id , num }
- FIRST(expression') = { + , - , ε }
- FIRST (factor) = { id , num }

- FOLLOW(program) = { \$ }
- FOLLOW(statement_list) = { end , FOLLOW (statement_list') }
  = { end } Asked the prof about what to do in this situation. Waiting for a response.

- FOLLOW(statement_list') = { FOLLOW (statement_list) }
  = { end }
- FOLLOW(statement) = { (FIRST (statement_list') − ε) ∪ FOLLOW(statement_list) }
  = { ; , end }

Chukwuemeka Ajaero, 8593168

Chukwuemeka Ajaero, 8593168

- FOLLOW(expression) = { FOLLOW(statement) }
  = { ; , end } ← Remember to redo if wrong

- FOLLOW(expression') = { FOLLOW(expression) }
  = { ; , end }

- FOLLOW(factor) = { (FIRST(expression') − ε) ∪ FOLLOW(expression)), FOLLOW(expression') }
  = { +, −, ;, end } ← Redo

Chukwuemeka Ajaero, 8593168

Exercise 3.2

	begin	end	;	id	num	=	+	-	$
program	<program> ➜ begin <statement_list> end	-	-	-	-	-	-	-	-
statement_list	-	-	-	<statement_list> ➜ <statement> <statement_list’>	-	<statement_list> ➜ <statement> <statement_list’>	-	-	-
statement_list’	-	<statement_list’> ➜ ε	<statement_list’> ➜ ; <statement_list>	-	-	-	-	-	-
statement	-	-	-	<statement> id = <expression>	-	<statement> id = <expression>	-	-	-
expression	-	-	-	<expression> ➜ <factor> <expression’>	<expression> ➜ <factor> <expression’>	-	-	-	-
expression’	-	<expression> ➜ ε	<expression> ➜ ε	-	-	-	<expression> ➜	<expression> ➜	-

Chukwuemeka Ajaero, 8593168

							+<factor>	-<factor>	
factor	-	-	-	<factor> ➜ id	<factor> ➜ num	-	-	-	-