## INFO-F-403 : Language theory and compiling Rapport projet partie 2 - Grammaire

## Simon Picard Arnaud Rosette

## 17 décembre 2014

## 1 Grammaire

[1]	<program></program>	$\rightarrow$	<InstructionList $>$
[2]	<InstructionList $>$	$\rightarrow$	<identifierinstruction></identifierinstruction>
[3]		$\rightarrow$	END_OF_INSTRUCTION <instructionlist> <constdefinition> END_OF_INSTRUCTION <instructionlist></instructionlist></constdefinition></instructionlist>
[4]		$\rightarrow$	<block> END_OF_INSTRUCTION</block>
[5]		$\rightarrow$	<instructionlist> <loop> END_OF_INSTRUCTION</loop></instructionlist>
[6]		$\rightarrow$	<instructionlist> <builtinfunctioncall> END_OF_INSTRUCTION</builtinfunctioncall></instructionlist>
[7]		$\rightarrow$	<instructionlist> <functiondefinition> END_OF_INSTRUCTION</functiondefinition></instructionlist>
[8]		$\rightarrow$	<pre><instructionlist> END_OF_INSTRUCTION <instructionlist></instructionlist></instructionlist></pre>
[9]		$\rightarrow$	EPSILON_VALUE
[10]	<identifierinstruction></identifierinstruction>	$\rightarrow$	IDENTIFIER <identifierinstructiontail></identifierinstructiontail>
[11] [12]	<IdentifierInstructionTail $>$	$\rightarrow$ $\rightarrow$	<assignationtail> TYPE DEFINITION <type></type></assignationtail>
[13]		$\stackrel{'}{\rightarrow}$	<pre><functioncalltail></functioncalltail></pre>
[14]	<AssignationTail $>$	$\rightarrow$	ASSIGNATION < Expression>
[15]		$\rightarrow$	COMMA IDENTIFIER <assignationtail> COMMA</assignationtail>
			<expression></expression>
[16]	<ConstDefinition $>$	$\rightarrow$	CONST IDENTIFIER <assignationtail></assignationtail>
[17]	<block></block>	$\rightarrow$	${\rm LET\ IDENTIFIER} < {\rm Assignation Tail} >$
			END_OF_INSTRUCTION <instructionlist> END</instructionlist>

[18]	<loop></loop>	$\rightarrow$	<if></if>
[19]		$\rightarrow$	WHILE <expression> END_OF_INSTRUCTION</expression>
[20]		$\rightarrow$	<pre><instructionlist> END FOR IDENTIFIER ASSIGNATION <expression></expression></instructionlist></pre>
[20]		$\rightarrow$	TERNARY ELSE < Expression > < ForTail >
[21]	<ForTail $>$	$\rightarrow$	END OF INSTRUCTION <instructionlist> END</instructionlist>
[22]		$\rightarrow$	TERNARY ELSE < Expression >
			END_OF_INSTRUCTION <instructionlist> END</instructionlist>
[23]	<Type $>$	$\rightarrow$	BOOLEAN_TYPE
[24]		$\rightarrow$	REAL_TYPE
[25]		$\rightarrow$	INTEGER_TYPE
[26]	<expression></expression>	$\rightarrow$	<binaryexpression> <ternaryifexpression></ternaryifexpression></binaryexpression>
[27]	<TernaryIfExpression $>$	$\rightarrow$	TERNARY_IF <expression></expression>
[0.0]			<ternaryelseexpression></ternaryelseexpression>
[28]	an plp :	$\rightarrow$	EPSILON_VALUE
[29]	<ternaryelseexpression></ternaryelseexpression>	$\rightarrow$	TERNARY_ELSE <expression></expression>
[30]	<AtomicExpression $>$	$\rightarrow$	<atomicidentifierexpression> INTEGER</atomicidentifierexpression>
[31] [32]		$\stackrel{ o}{ o}$	REAL
$\begin{bmatrix} 32 \end{bmatrix}$		$\rightarrow$	BOOLEAN
[34]		$\stackrel{'}{\rightarrow}$	<builtinfunctioncall></builtinfunctioncall>
[35]	<AtomicIdentifierExpression $>$	$\stackrel{'}{ ightarrow}$	IDENTIFIER
[00]	(1100mioraniana)	,	<atomicidentifierexpressiontail></atomicidentifierexpressiontail>
[36]	<AtomicIdentifierExpressionTail $>$	$\rightarrow$	<functioncalltail></functioncalltail>
[37]	•	$\rightarrow$	EPSILON VALUE
[38]	<UnaryExpression $>$	$\rightarrow$	NEGATION < Unary Expression >
[39]		$\rightarrow$	$<\!$
[40]	$<\!\! \text{UnaryBitwiseNotExpression} \!\!>$	$\rightarrow$	${\bf BITWISE\_NOT} < {\bf UnaryBitwiseNotExpression} >$
[41]		$\rightarrow$	<UnaryMinusPlusExpression $>$
[42]	<UnaryMinusPlusExpression $>$	$\rightarrow$	MINUS < Unary Minus Plus Expression >
[43]		$\rightarrow$	PLUS < UnaryMinusPlusExpression>
[44]		$\rightarrow$	<unaryatomicexpression></unaryatomicexpression>
[45]	<Unary $AtomicExpression>$	$\rightarrow$	<atomicexpression></atomicexpression>
[46]		$\rightarrow$	LEFT_PARENTHESIS < Expression >
[47]	<dinom-europagaion></dinom-europagaion>	,	RIGHT_PARENTHESIS
[47]	<binaryexpression></binaryexpression>	$\rightarrow$	<binarylazyorexpression> <binaryexpression'></binaryexpression'></binarylazyorexpression>
[48]	<binaryexpression'></binaryexpression'>	$\rightarrow$	LAZY_OR <binarylazyorexpression></binarylazyorexpression>
[40]	\DinaryExpression >	_	<pre> </pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pr< td=""></pr<></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
[49]		$\rightarrow$	EPSILON VALUE
[50]	<binarylazyorexpression></binarylazyorexpression>	$\stackrel{'}{ ightarrow}$	<pre><binarylazyandexpression></binarylazyandexpression></pre>
r -1	v v r		<binarylazyorexpression'></binarylazyorexpression'>

[51]	$<\!$	$\rightarrow$	LAZY_AND <binarylazyandexpression></binarylazyandexpression>
[59]			<binarylazyorexpression'> EPSILON VALUE</binarylazyorexpression'>
[52] [53]	<binarylazyandexpression></binarylazyandexpression>	$\overset{\rightarrow}{\rightarrow}$	<pre><binarynumericexpression></binarynumericexpression></pre>
[99]	\Dinary Lazy And Expression >	/	<pre><binarytvuniericexpression'></binarytvuniericexpression'></pre>
[54]	<binarylazyandexpression'></binarylazyandexpression'>	$\rightarrow$	GREATER THAN <binarynumericexpression></binarynumericexpression>
	(Billory Eddy Fillad Empression )	,	<binarylazyandexpression'></binarylazyandexpression'>
[55]		$\rightarrow$	LESS THAN <binarynumericexpression></binarynumericexpression>
[]			<binarylazyandexpression'></binarylazyandexpression'>
[56]		$\rightarrow$	GREATER_OR_EQUALS_THAN
. ,			<binarynumericexpression></binarynumericexpression>
			<binarylazyandexpression'></binarylazyandexpression'>
[57]		$\rightarrow$	LESS OR EQUALS THAN
			<binarynumericexpression></binarynumericexpression>
			<binarylazyandexpression'></binarylazyandexpression'>
[58]		$\rightarrow$	${\bf EQUALITY} < {\bf Binary Numeric Expression} >$
			<BinaryLazyAndExpression'>
[59]		$\rightarrow$	INEQUALITY < Binary Numeric Expression >
			<BinaryLazyAndExpression' $>$
[60]		$\rightarrow$	EPSILON_VALUE
[61]	<BinaryNumericExpression $>$	$\rightarrow$	<binarytermexpression></binarytermexpression>
			<binarynumericexpression'></binarynumericexpression'>
[62]	<binarynumericexpression'></binarynumericexpression'>	$\rightarrow$	PLUS <binarytermexpression></binarytermexpression>
			<binarynumericexpression'></binarynumericexpression'>
[63]		$\rightarrow$	MINUS <binarytermexpression></binarytermexpression>
Fo. 43			<binarynumericexpression'></binarynumericexpression'>
[64]		$\rightarrow$	BITWISE_OR <binarytermexpression></binarytermexpression>
[a=1			<binarynumericexpression'></binarynumericexpression'>
[65]		$\rightarrow$	BITWISE_XOR <binarytermexpression></binarytermexpression>
[0.0]			<binarynumericexpression'></binarynumericexpression'>
[66]	an: III III ii	$\rightarrow$	EPSILON_VALUE
[67]	<binarytermexpression></binarytermexpression>	$\rightarrow$	<binaryshiftedexpression></binaryshiftedexpression>
[60]	DinameTonnaEvennagaian?	,	<pre><binarytermexpression'></binarytermexpression'></pre>
[68]	<BinaryTermExpression' $>$	$\rightarrow$	ARITHMETIC_SHIFT_LEFT <binaryshiftedexpression></binaryshiftedexpression>
			<pre><binarytermexpression'></binarytermexpression'></pre>
[69]		$\rightarrow$	ARITHMETIC SHIFT RIGHT
[09]		7	<pre><binaryshiftedexpression></binaryshiftedexpression></pre>
			<pre><binarytermexpression'></binarytermexpression'></pre>
[70]		$\rightarrow$	EPSILON VALUE
[71]	<binaryshiftedexpression></binaryshiftedexpression>	$\stackrel{}{\rightarrow}$	<pre><binaryfactorexpression></binaryfactorexpression></pre>
[•+]	Differ y offitted Dapi coston /	,	<pre><binaryfactorexpression'></binaryfactorexpression'></pre>

[72]	$<\!$	$\rightarrow$	TIMES <binaryfactorexpression></binaryfactorexpression>
[73]		$\rightarrow$	<binaryshiftedexpression'> DIVIDE <binaryfactorexpression></binaryfactorexpression></binaryshiftedexpression'>
[10]			<pre> </pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre< td=""></pre<></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
[74]		$\rightarrow$	REMAINDER <binaryfactorexpression></binaryfactorexpression>
r. 1			<binaryshiftedexpression'></binaryshiftedexpression'>
[75]		$\rightarrow$	BITWISE AND <binaryfactorexpression></binaryfactorexpression>
			<binaryshiftedexpression'></binaryshiftedexpression'>
[76]		$\rightarrow$	$INVERSE\_DIVIDE < BinaryFactorExpression>$
			<BinaryShiftedExpression' $>$
[77]		$\rightarrow$	EPSILON_VALUE
[78]	<BinaryFactorExpression $>$	$\rightarrow$	<unaryexpression></unaryexpression>
			<binaryfactorexpression'></binaryfactorexpression'>
[79]	<BinaryFactorExpression' $>$	$\rightarrow$	POWER < Unary Expression >
f 1			<binaryfactorexpression'></binaryfactorexpression'>
[80]	7.0	$\rightarrow$	EPSILON_VALUE
[81]	<If $>$	$\rightarrow$	IF <expression> END_OF_INSTRUCTION</expression>
[00]	.I(T) 1.		<instructionlist> <ifend></ifend></instructionlist>
[82]	<ifend></ifend>	$\rightarrow$	ELSE_IF <expression> END_OF_INSTRUCTION</expression>
[60]		,	<instructionlist> <ifend></ifend></instructionlist>
[83] [84]		$\rightarrow$	ELSE <instructionlist> END END</instructionlist>
[84]	<builtinfunctioncall></builtinfunctioncall>	$\rightarrow$ $\rightarrow$	READ REAL LEFT PARENTHESIS
[00]	\Duntim unction\dan\	7	RIGHT PARENTHESIS
[86]		$\rightarrow$	READ INTEGER LEFT PARENTHESIS
[OO]		,	RIGHT PARENTHESIS
[87]		$\rightarrow$	INTEGER CAST LEFT PARENTHESIS < Expression>
[0.]			RIGHT PARENTHESIS
[88]		$\rightarrow$	REAL_CAST LEFT_PARENTHESIS < Expression>
. ,			RIGHT PARENTHESIS
[89]		$\rightarrow$	BOOLEAN CAST LEFT PARENTHESIS < Expression>
			RIGHT_PARENTHESIS
[90]		$\rightarrow$	PRINTLN LEFT_PARENTHESIS < Expression>
			RIGHT_PARENTHESIS
[91]	<FunctionCallTail $>$	$\rightarrow$	LEFT_PARENTHESIS < Parameter>
			RIGHT_PARENTHESIS
[92]	<parameter></parameter>	$\rightarrow$	<expression> <parametertail></parametertail></expression>
[93]		$\rightarrow$	EPSILON_VALUE
[94]	<parametertail></parametertail>	$\rightarrow$	COMMA < Expression > < Parameter Tail >
[95]		$\rightarrow$	EPSILON_VALUE

[96]	<functiondefinition></functiondefinition>	_	FUNCTION IDENTIFIER LEFT PARENTHESIS
[90]	\T unctionDemittion		<pre><argument> RIGHT PARENTHESIS</argument></pre>
			<u> </u>
			<InstructionList $>$ $<$ FunctionDefinitionEnd $>$
[97]	<FunctionDefinitionEnd $>$	$\rightarrow$	RETURN < Expression > END
[98]		$\rightarrow$	END
[99]	<argument></argument>	$\rightarrow$	IDENTIFIER TYPE DEFINITION < Type>
			<argumenttail></argumenttail>
[100]		$\rightarrow$	EPSILON VALUE
[101]	<argumenttail></argumenttail>	$\rightarrow$	COMMA IDENTIFIER TYPE DEFINITION < Type>
			<argumenttail></argumenttail>
[102]		$\rightarrow$	EPSILON VALUE
			_