

Le langage MiniJaja

Nous définissons le langage MiniJaja sous la forme d'une grammaire non contextuelle de Backus-Naur avec les notations suivantes :

- les *symboles terminaux* sont soit des identificateurs soulignés (vide exprimant la chaîne vide), soit des caractères entre guillemets tels que " {", "(", "+"
- les *symboles non terminaux* sont tous les identificateurs non soulignés
- les symboles \rightarrow , $[$, $]$ et $|$ sont des symboles du méta-langage exprimant respectivement la dérivation, la présence facultative et le choix entre les règles.

classe \rightarrow	<u>class</u> ident " { " decls methmain " } "	<i>classe</i> (\$2,\$4,\$5)
ident \rightarrow	<u>identificateur</u>	<i>ident</i> (\$1)
decls \rightarrow	decl ";" decls <u>vide</u>	<i>decls</i> (\$1,\$3) <i>vnil</i>
decl \rightarrow	var methode	<i>\$1</i> <i>\$1</i>
vars \rightarrow	var ";" vars <u>vide</u>	<i>vars</i> (\$1,\$3) <i>vnil</i>
var \rightarrow	typemeth ident vexp typemeth ident "[" exp "]" <u>final</u> type ident vexp	<i>var</i> (\$1, \$2, \$3) <i>tableau</i> (\$1, \$2, \$4) <i>cst</i> (\$2, \$3, \$4)
vexp \rightarrow	"=" exp <u>vide</u>	<i>\$2</i> <i>omega</i>
methode \rightarrow	typemeth ident "(" entêtes ")" " { " vars instrs " } "	<i>méthode</i> (\$1, \$2, \$4, \$7, \$8)
methmain \rightarrow	<u>main</u> " { " vars instrs " } "	<i>main</i> (\$3,\$4)
entêtes \rightarrow	entête ", " entêtes entête <u>vide</u>	<i>entêtes</i> (\$1,\$3) <i>entêtes</i> (\$1, <i>enil</i>) <i>enil</i>
entête \rightarrow	type ident	<i>entête</i> (\$1,\$2)
instrs \rightarrow	instr ";" instrs <u>vide</u>	<i>instrs</i> (\$1, \$3) <i>inil</i>
instr \rightarrow	ident1 "=" exp ident1 "+=" exp ident1 "++" ident "(" listexp ")" <u>return</u> exp <u>if</u> exp " { " instrs " } " [<u>else</u> " { " instrs " } "] <u>while</u> "(" exp ")" " { " instrs " } "	<i>affectation</i> (\$1,\$3) <i>somme</i> (\$1,\$3) <i>incrément</i> (\$1) <i>appelI</i> (\$1,\$3) <i>retour</i> (\$2) <i>si</i> (\$2, \$4, \$8) <i>tantque</i> (\$3, \$6)
listexp \rightarrow	exp ", " listexp exp <u>vide</u>	<i>listexp</i> (\$1,\$3) <i>listexp</i> (\$1, <i>exnil</i>) <i>exnil</i>
exp \rightarrow	"!" exp1 exp "&&" exp1 exp " " exp1 exp1	<i>non</i> (\$2) <i>et</i> (\$1, \$3) <i>ou</i> (\$1, \$3) <i>\$1</i>
exp1 \rightarrow	exp1 "==" exp2 exp1 ">" exp2 exp2	<i>==</i> (\$1,\$3) <i>></i> (\$1,\$3) <i>\$1</i>
exp2 \rightarrow	exp2 "+" terme exp2 "-" terme "-" terme terme	<i>+</i> (\$1,\$3) <i>-</i> (\$1,\$3) <i>moins</i> (\$2) <i>\$1</i>
terme \rightarrow	terme "*" fact terme "/" fact fact	<i>*</i> (\$1,\$3) <i>/</i> (\$1,\$3) <i>\$1</i>
fact \rightarrow	ident1 ident "(" listexp ")" <u>true</u> <u>false</u> <u>nombre</u> "(" exp ")"	<i>\$1</i> <i>appelE</i> (\$1,\$3) <i>vrai</i> <i>faux</i> <i>nbre</i> (\$1) <i>\$2</i>
ident1 \rightarrow	ident ident "[" exp "]"	<i>\$1</i> <i>tab</i> (\$1,\$3)
typemeth \rightarrow	<u>void</u> type	<i>rien</i> <i>\$1</i>
type \rightarrow	<u>int</u> <u>boolean</u>	<i>entier</i> <i>booléen</i>