

Le langage MiniJaja

Nous définissons, un 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, 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 métalangage exprimant 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	\$1
methode	\$1
vars \rightarrow var “;” vars <u>vide</u>	<i>vars</i> (\$1,\$3) <i>vnil</i>
var \rightarrow typemeth ident vexp	<i>var</i> (\$1, \$2, \$3)
typemeth ident “[“ exp “]”	<i>tableau</i> (\$1, \$2, \$4)
<u>final</u> type ident vexp	<i>cst</i> (\$2, \$3, \$4)
vexp \rightarrow “=” exp <u>vide</u>	\$2 <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	<i>entêtes</i> (\$1,\$3)
entête <u>vide</u>	<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	<i>affectation</i> (\$1,\$3)
ident1 “+=” exp	<i>somme</i> (\$1,\$3)
ident1 “++”	<i>incrément</i> (\$1)
ident (“listexp”)	<i>appel</i> (\$1,\$3)
<u>return</u> exp	<i>retour</i> (\$2)
if exp “{“ instrs “}” [else “{“ instrs “}”]	<i>si</i> (\$2, \$4, \$8)
<u>while</u> exp “{“ instrs “}”	<i>tantque</i> (\$2, \$4)
listexp \rightarrow exp “;” listexp	<i>listexp</i> (\$1,\$3)
exp <u>vide</u>	<i>listexp</i> (\$1, <i>exnil</i>) <i>exnil</i>
exp \rightarrow “!” exp1 “-” exp1	<i>non</i> (\$2) <i>moins</i> (\$2)
exp “&&” exp1	<i>et</i> (\$1, \$3)
exp “ ” exp1 exp1	<i>ou</i> (\$1, \$3) \$1
exp1 \rightarrow exp1 “==” exp2	= <i>(\$1,\$3)</i>
exp1 “>” exp2 exp2	<i>></i> (\$1,\$3) \$1
exp2 \rightarrow exp2 “+” terme	<i>+</i> (\$1,\$3)
exp2 “-” terme terme	<i>-</i> (\$1,\$3) \$1
terme \rightarrow terme “*” fact	<i>*</i> (\$1,\$3)
terme “/” fact fact	<i>/</i> (\$1,\$3) \$1
fact \rightarrow ident1	\$1
ident (“listexp”)	<i>appelE</i> (\$1,\$3)
<u>true</u> <u>false</u> <u>nombre</u>	<i>vrai</i> <i>faux</i> <i>nbre</i> (\$1)
“(“exp”)	\$2
ident1 \rightarrow ident ident “[“exp”]“	\$1 <i>tab</i> (\$1,\$3)
typemeth \rightarrow <u>void</u>	<i>rien</i>
type	\$1
type \rightarrow <u>int</u> <u>boolean</u>	<i>entier</i> <i>booléen</i>