$r_1 :< fichier > \rightarrow$ with Ada.Text_IO; use Ada.Text_IO; procedure $< ident >$ is $< fichier >_2$

$r_2 :< fichier >_2 \rightarrow$ begin $< instr >^+$ end $< fichier >_3$
$r_3 : \rightarrow < decl >^+$ begin $< instr >^+$ end $< fichier >_3$

$r_4 :< fichier >_3 \rightarrow$ ; EOF
$r_5 : \rightarrow < ident >$; EOF

$r_6 :< decl > \rightarrow$ type $< ident > < decl >_{11}$
$r_7 : \rightarrow$ procedure $< ident > < decl >_{21}$
$r_8 : \rightarrow$ function $< ident > < decl >_{31}$
$r_9 : \rightarrow < ident >_,^+ : < ident > < decl >_{41}$

$r_{10} :< decl >_{11} \rightarrow$ ;
$r_{11} : \rightarrow$ is record $< champs >^+$ end record;

$r_{12} :< decl >_{21} \rightarrow$ is $< decl >_{22}$
$r_{13} : \rightarrow < params >$ is $< decl >_{22}$

$r_{14} :< decl >_{22} \rightarrow$ begin $< instr >^+$ end $< decl >_{23}$
$r_{15} : \rightarrow < decl >^+$ begin $< instr >^+$ end $< decl >_{23}$

$r_{16} :< decl >_{23} \rightarrow$ ;
$r_{17} : \rightarrow < ident >$ ;

$r_{18} :< decl >_{31} \rightarrow$ return $< ident >$ is $< decl >_{22}$
$r_{19} : \rightarrow < params >$ return $< ident >$ is $< decl >_{22}$

$r_{20} :< decl >_{41} \rightarrow$ ;
$r_{21} : \rightarrow :=< expr >$;

$r_{22} :< champs > \rightarrow < ident >_,^+ : < ident >$;

$r_{23} :< params > \rightarrow (< param >_;^+)$

$r_{24}$ :$< param >$ $\rightarrow < ident >^{+}_{,} : < param >_2$

$r_{25}$ :$< param >_2$ $\rightarrow < ident >$

$r_{26}$ : $\rightarrow < mode > < ident >$

$r_{27}$ :$< mode >$ $\rightarrow$ in $< mode >_1$

$r_{28}$ :$< mode >_1$ $\rightarrow$ out

$r_{29}$ : $\rightarrow \wedge$

$r_{30}$ :$< expr >$ $\rightarrow N_1 < disjonction >$

$r_{31}$ :$< disjonction >$ $\rightarrow$ or $< expr >$

$r_{32}$ : $\rightarrow$ or else $< expr >$

$r_{33}$ : $\rightarrow \wedge$

$r_{34}$ : $N_1$ $\rightarrow N_2 < conjonction >$

$r_{35}$ :$< conjonction >$ $\rightarrow$ and $N_1$

$r_{36}$ : $\rightarrow$ and then $N_1$

$r_{37}$ : $\rightarrow \wedge$

$r_{38}$ : $N_2$ $\rightarrow N_3 < comparaison_1 >$

$r_{39}$ :$< comparaison_1 >$ $\rightarrow = N_2$

$r_{40}$ : $\rightarrow /= N_2$

$r_{41}$ : $\rightarrow \wedge$

$r_{42}$ : $N_3$ $\rightarrow N_4 < comparaison_2 >$

$r_{43}$ :$< comparaison_2 >$ $\rightarrow > N_3$

$r_{44}$ : $\rightarrow >= N_3$

$r_{45}$ : $\rightarrow < N_3$

$r_{46}$ : $\rightarrow <= N_3$

$r_{47}$ : $\rightarrow \wedge$

$r_{48}$ : $N_4$ $\rightarrow N_5 < operation_1 >$

$r_{49}$ :$< operation_1 >$ $\rightarrow +N_4$

$r_{50}$ : $\rightarrow -N_4$

$r_{51}$ : $\rightarrow \wedge$

$r_{52}$ : $N_5$ $\rightarrow < negation > < operation_2 >$

$r_{53}$ :$< operation_2 >$ $\rightarrow *N_5$

$r_{54}$ : $\rightarrow /N_5$

$r_{55}$ : $\rightarrow remN_5$

$r_{56}$ : $\rightarrow \wedge$

$r_{57}$ :$< negation >$ $\rightarrow < axiome >$

$r_{58}$ : $\rightarrow - < axiome >$

$r_{59}$ : $\rightarrow not\ < axiome >$

$r_{60}$ :$< axiome >$ $\rightarrow < entier > < axiome >_{recur}$

$r_{61}$ : $\rightarrow < caractere > < axiome >_{recur}$

$r_{62}$ : $\rightarrow true < axiome >_{recur}$

$r_{63}$ : $\rightarrow false < axiome >_{recur}$

$r_{64}$ : $\rightarrow null < axiome >_{recur}$

$r_{65}$ : $\rightarrow$ character ' val $(< expr >) < axiome >_{recur}$

$r_{66}$ : $\rightarrow (< expr >) < axiome >_{recur}$

$r_{67}$ : $\rightarrow < ident > < axiome >_1$

$r_{68}$ :$< axiome >_{recur}$ $\rightarrow . < ident > < axiome >_{recur}$

$r_{69}$ : $\rightarrow \wedge$

$r_{70}$ :$< axiome >_1$ $\rightarrow < axiome >_{recur}$

$r_{71}$ : $\rightarrow (< expr >_{,}^{+}) < axiome >_{recur}$

$r_{72}$ :$< instr >$ $\rightarrow < ident > < instr >_1$

$r_{73}$ : $\rightarrow - < axiome >< suite > . < ident > := < expr >;$

$r_{74}$ : $\rightarrow not \ < axiome >< suite > . < ident > := < expr >;$

$r_{75}$ : $\rightarrow < entier >< axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{76}$ : $\rightarrow < caractere >< axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{77}$ : $\rightarrow true \ < axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{78}$ : $\rightarrow false \ < axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{79}$ : $\rightarrow null \ < axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{80}$ : $\rightarrow character'val(< expr >) < axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{81}$ : $\rightarrow (< expr >) < axiome >_{recur}< suite > . < ident > := < expr >;$

$r_{82}$ : $\rightarrow$ return $< instr >_2$

$r_{83}$ : $\rightarrow$ begin $< instr >^{+}$ end;

$r_{84}$ : $\rightarrow$ if $< expr >$ then $< instr >^{+} < instr >_3$

$r_{85}$ : $\rightarrow$ for $< ident >$ in $< instr >_4$

$r_{86}$ : $\rightarrow$ while $< expr >$ loop $< instr >^{+}$ end loop;

$r_{87}$ : $\rightarrow$ put$(< expr >)$;

$r_{88}$ :$< instr >_1$ $\rightarrow := < expr >;$

$r_{89}$ : $\rightarrow ;$

$r_{90}$ : $\rightarrow (< expr >_{,}^{+}) < instr >_{11}$

$r_{91}$ : $\rightarrow . < ident > < suite > . < ident > := < expr > ;$

$r_{92}$ : $\rightarrow < suite > . < ident > := < expr > ;$

$r_{93}$ :$< instr >_{11}$ $\rightarrow ;$

$r_{94}$ : $\rightarrow < axiome >_{recur} < suite > . < ident > := < expr > ;$

Pour abréger on note $< suite > = < operation_2 >< operation_1 >< comparaison_2 >< comparaison_1 >< conjonction >< disjonction >$

$r_{95}$ :$< instr >_2$ $\rightarrow < expr >$;

$r_{96}$ : $\rightarrow$ ;

$r_{97}$ :$< instr >_3$ $\rightarrow$ end if;

$r_{98}$ : $\rightarrow$ else $< instr >^+$ end if;

$r_{99}$ : $\rightarrow < elsif >^+ < instr >_3$

$r_{100}$ :$< instr >_4$ $\rightarrow < expr >$ .. $< expr >$ loop $< instr >^+$ end loop;

$r_{101}$ : $\rightarrow$ reverse $< expr >$ .. $< expr >$ loop $< instr >^+$ end loop;

$r_{102}$ :$< instr >^+$ $\rightarrow < instr > < instr >^+_1$

$r_{103}$ :$< instr >^+_1$ $\rightarrow < instr >^+$

$r_{104}$ : $\rightarrow \wedge$

$r_{105}$ :$< decl >^+$ $\rightarrow < decl > < decl >^+_1$

$r_{106}$ :$< decl >^+_1$ $\rightarrow < decl >^+$

$r_{107}$ : $\rightarrow \wedge$

$r_{108}$ :$< champs >^+$ $\rightarrow < champs > < champs >^+_1$

$r_{109}$ :$< champs >^+_1$ $\rightarrow < champs >^+$

$r_{110}$ : $\rightarrow \wedge$

$r_{111}$ :$< ident >^+_,$ $\rightarrow < ident > < ident >^+_{,1}$

$r_{112}$ :$< ident >^+_{,1}$ $\rightarrow$ , $< ident >^+_,$

$r_{113}$ : $\rightarrow \wedge$

$r_{114}$ :$< param >^+_;$ $\rightarrow < param > < param >^+_{;1}$

$r_{115}$ :$< param >^+_{;1}$ $\rightarrow$ ; $< param >^+_;$

$r_{116}$ : $\rightarrow \wedge$

$r_{117}$ :$< expr >^+_,$ $\rightarrow < expr > < expr >^+_{,1}$

$r_{118}$ :$< expr >^+_{,1}$ $\rightarrow$ , $< expr >^+_,$

$r_{119}$ : $\rightarrow \wedge$

$r_{120}$ :$< elsif >^+$ $\rightarrow$ elsif $< expr >$ then $< instr >^+ < elsif >^+_1$

$r_{121}$ :$< elsif >^+_1$ $\rightarrow < elsif >^+$

$r_{122}$ : $\rightarrow \wedge$