

Schema di traduzione per la grammatica di P:

$P \rightarrow \{SL.next = P.next\}SL\{emitLabel(P.next)\}EOF$
 $SL \rightarrow \{S.next = newlabel()\}S\{emitLabel(S.next); SLP.next = SL.next\}SLP$
 $SLP \rightarrow \{S.next = newlabel()\}S\{emitLabel(S.next)\}SLP$
 $S \rightarrow ID := \{emit(istore, id.address)\}$
 $S \rightarrow print(E\{emit(invokestatic, 1)\})$
 $S \rightarrow read(ID\{emit(invokestatic, 0); emit(istore, id.address)\})$
 $S \rightarrow case\{WL.next = S.next\}WL\{S1.next = S.next\}S1$
 $S \rightarrow while(\{BE.true = newLabel(); BE.false = S.next; S1.next = newLabel(); emitlabel(S.next)\}BE)\{emitlabel(BE.true)$
 $S \rightarrow \{SL.next = S.next\}\{SL\}$
 $WL \rightarrow \{WI.next = WL.next; WI.w_{next} = newLabel()\}WI\{emitlabel(WI.w_{next}); WLP.next = WL.next\}WLP$
 $WLP \rightarrow \{WI.next = WLP.next; WI.w_{next} = newLabel()\}WI\{emitlabel(WI.w_{next}); WLP1.next = WLP.next\}WLP1$
 $WI \rightarrow when(\{BE.true = newlabel(); BE.false = WI.w_{next}\}BE)\{emitlabel(BE.true); S.next = WI.next\}S\{emit(goto, W$