Prog ::= Defs #

Defs ::= Def

| Def Defs

Def ::= DEF Lhs == Expr

Lhs ::= MAIN : Type

| id ( ) : Type

| id ( Params ) : Type

Params ::= Param

| Param, Params

Param ::= id : Type

Type ::= nat | bool

Expr ::= number | true | false

| id

| id ( Args )

| IF Expr THEN Expr FI

| IF Expr THEN Expr ELSE Expr FI

Args ::= Expr

| Expr, Args

Prog ::= Defs # (1)

Defs ::= Def Def1

Def1 ::= (2)

| Defs (3)

Def ::= DEF Lhs == Expr (4)

Lhs ::= MAIN : Type (5)

| id ( Lhs1

Lhs1 ::= ) : Type (6)

| Params ) : Type (7)

Params ::= Param Params1

Params1 ::= (8)

| , Params (9)

Param ::= id : Type (10)

Type ::= nat (11) | bool (12)

Expr ::= number (13) | true (14) | false (15)

| id Expr1

| IF Expr THEN Expr Expr2

Expr1 ::= (16)

| ( Args ) (17)

Expr2 ::= FI (18)

| ELSE Expr FI (19)

Args ::= Expr Args1

Args1 ::= (20)

| Expr, Args (21)

Nullable: NO: Prog, Defs, Def, Lhs, Lhs1, Params, Param, Type, Expr, Expr2, Args

YES: Def1, Params1, Expr1, Args1

Dir(2) = Fst (Def Def1) = Fst (Def) = {DEF}

Dir(3) = Flw(Def1) = Flw(Defs) = {#}

Dir(4) = Fst(Defs) = Fst(Def) = {DEF}

Dir(5) = {DEF}

Dir(6) = {MAIN}

Dir(7) = {id}

Dir(8) = {)}

Dir(9) = Fst(Params) = Fst (Param) = {id}

Dir(10) = Fst(Param) = {id}

Dir(11) = Flw(Params1) = {,}

Dir(12) = {,}

Dir(13) = {id}

Dir(14) = {nat; bool}

Dir(15) = {number; true; false}

Dir(16) = {id}

Dir(17) = {IF}

Dir(18) = Flw(Expr1) = Flw(Expr) = {THEN; FI; ELSE; number; true; false; id; IF; ,}

Dir(19) = {(}

Dir(20) = {FI}

Dir(21) = {ELSE}

Dir(22) = {number; true; false; id; IF}

Dir(23) = Flw(Args1) = Flw(Args) = {)} ∪ Flw(Args1) = {)}

Dir(24) = {number; true; false; id; IF}

Flw(Defs) = {#} ∪ Flw(Def1) = {#} ∪ Flw(Defs) = {#}

Flw(Params1) = Flw(Params) = {)} ∪ Flw(Params1) = {)}

Flw(Expr) = {THEN} ∪ Fst(Expr2) ∪ {FI} ∪ Fst(Arg1) ∪ {,} = {THEN; FI; ELSE; number; true; false; id; IF; ,}