/SU/FSI/MASTER/INFO/MU4IN503
APS
Formulaire

P. Manoury [*]

Janvier 2022

4 APS2

4.1 Syntaxe

Lexique

Symboles réservés

`[ ] ( ) ; : , * ->`

Mots clef

```
CONST FUN REC VAR PROC
ECHO SET IF WHILE CALL
if and or
bool int vec
var adr
alloc len nth vset
```

Constantes numériques

num défini par *('-' ?)['0'-'9']+*

Identificateurs

ident défini par *(['a'-'z''A'-'Z'])(['a'-'z''A'-'Z''0'-'9'])**
dont on exclut les mots clef.

Remarque : les symboles d'opérateurs primitifs

`true false not eq lt add sub mul div`

sont des identificateurs.

Grammaire

Programme

Prog ::= Block

Bloc

Block ::= [Cmds]

Suite de commandes

Cmds ::= Stat
 | Def ; Cmds
 | Stat ; Cmds

*Avec la précieuse relecture de W.S. et V.M. Qu'ils en soient remerciés.

Définition

```
DEF     ::=  CONST ident TYPE EXPR
          |  FUN ident TYPE [ ARGS ] EXPR
          |  FUN REC ident TYPE [ ARGS ] EXPR
          |  VAR ident STYPE
          |  PROC ident [ ARGSP ] BLOCK
          |  PROC REC ident [ ARGSP ] BLOCK
```

Type

```
TYPE    ::=  STYPE
          |  ( TYPES -> TYPE )
TYPES   ::=  TYPE
          |  TYPE * TYPES
```

SType

```
STYPE   ::=  bool | int
          |  ( vec STYPE )
```

Paramètre formel (fonctions)

```
ARGS    ::=  ARG
          |  ARG , ARGS
ARG     ::=  ident : TYPE
```

Paramètre formel (procédures)

```
ARGSP ::=   ::=  ARGP
              |  ARGP , ARGSP
ARGP        ::=  ident : TYPE
              |  var ident : TYPE
```

Instruction

```
STAT    ::=  ECHO EXPR
          |  SET LVALUE EXPR
          |  IF EXPR BLOCK BLOCK
          |  WHILE EXPR BLOCK
          |  CALL ident EXPRSP
```

lvalue

```
LVALUE  ::=  ident
          |  ( nth LVALUE EXPR )
```

Paramètres d'appel

```
EXPRSP  ::=  EXPRP
          |  EXPRP EXPRSP
EXPRP   ::=  EXPR
          |  (adr ident)
```

Expression

```
EXPR    ::=  num
          |  ident
          |  (if EXPR EXPR EXPR )
          |  ( and EXPR EXPR )
          |  ( or EXPR EXPR )
          |  ( EXPR EXPRS )
          |  [ ARGS ] EXPR
          |  (alloc EXPR )
          |  (len EXPR )
          |  (nth EXPR EXPR )
          |  (vset EXPR EXPR EXPR )
```

Suite d'expressions

EXPRS	::=	EXPR
	|	EXPR EXPRS

4.2 Typage

Soit $p1, \ldots, p_n \in$ EXPRP.
Posons $A([p_1 : t_1, \ldots, x_n : t_n]) = [x_1 : t'_1, \ldots, x_n : t'_n]$ avec

$$t'_i = \begin{cases} t_i & \text{si } p_i = x_i \\ (\texttt{ref}\ \ t_i) & \text{si } p_i = \texttt{var}\ \ x_i \end{cases}$$

Programmes

(PROG) si $\Gamma_0 \vdash_{\text{Block}} bk : \texttt{void}$
alors $\vdash bk : \texttt{void}$

Blocs

(BLOC) si $\Gamma \vdash_{\text{Cmds}} (cs\texttt{;}\varepsilon) : \texttt{void}$
alors $\Gamma \vdash_{\text{Block}} \texttt{[}cs\texttt{]} : \texttt{void}$

Suite de commandes

(DECS) si $d \in$ DEC, si $\Gamma \vdash_{\text{Def}} d : \Gamma'$, si $\Gamma' \vdash_{\text{Cmds}} cs : \texttt{void}$
alors $\Gamma \vdash_{\text{Cmds}} (d\texttt{;}cs) : \texttt{void}$.

(STATS) si $s \in$ STAT, si $\Gamma \vdash_{\text{Stat}} s : \texttt{void}$, si $\Gamma \vdash_{\text{Cmds}} cs : \texttt{void}$
alors $\Gamma \vdash_{\text{Cmds}} (s\texttt{;}cs) : \texttt{void}$.

(END) $\Gamma \vdash_{\text{Cmds}} \varepsilon : \texttt{void}$.

Définitions

(CONST) si $\Gamma \vdash_{\text{Expr}} e : t$
alors $\Gamma \vdash_{\text{Def}} (\texttt{CONST}\ x\ t\ e) : \Gamma[x : t]$

(FUN) si $\Gamma[x_1 : t_1; \ldots; x_n : t_n] \vdash_{\text{Expr}} e : t$
alors $\Gamma \vdash_{\text{Def}} (\texttt{FUN}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,}\ldots\texttt{,}x_n\texttt{:}t_n\texttt{]}\ e) : \Gamma[x : (t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t)]$

(FUNREC) si $\Gamma[x_1 : t_1; \ldots; x_n : t_n; x : t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t] \vdash_{\text{Expr}} e : t$
alors $\Gamma \vdash_{\text{Def}} (\texttt{FUN REC}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,}\ldots\texttt{,}x_n\texttt{:}t_n\texttt{]}\ e) : \Gamma[x : t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t]$

(VAR) si $t \in \{\texttt{int}, \texttt{bool}\}$
alors $\Gamma \vdash_{\text{Def}} (\texttt{VAR}\ x\ t) : \Gamma[x : (\texttt{ref}\ \ t)]$

(PROC) si $A([p_1 : t_1, \ldots, p_n : t_n]) = [x_1 : t'_1, \ldots, x_n : t'_n]$
si $\Gamma[x_1 : t'_1; \ldots; x_n : t'_n] \vdash_{\text{Block}} bk : \texttt{void}$
alors $\Gamma \vdash_{\text{Def}} (\texttt{PROC}\ x\ \texttt{[}p_1\texttt{:}t_1, \ldots, p_n\texttt{:}t_n\texttt{]}bk) : \Gamma[x : t'_1\ \texttt{*}\ \ldots\ \texttt{*}\ t'_n\ \texttt{->}\ \texttt{void}]$

(PROCREC)
si $A([p_1 : t_1, \ldots, x_n : t_n]) = [x_1 : t'_1, \ldots, x_n : t'_n]$
si $\Gamma[x_1 : t'_1; \ldots; x_n : t'_n; x : t'_1\ \texttt{*}\ \ldots\ \texttt{*}\ t'_n\ \texttt{->}\ \texttt{void}] \vdash_{\text{Block}} bk : \texttt{void}$
alors $\Gamma \vdash_{\text{Def}} (\texttt{PROC REC}\ x\ \texttt{[}p_1\texttt{:}t_1, \ldots, p_n\texttt{:}t_n\texttt{]}bk) : \Gamma[x : t'_1\ \texttt{*}\ \ldots\ \texttt{*}\ t'_n\ \texttt{->}\ \texttt{void}]$

Instructions

(ECHO) si $\Gamma \vdash_{\text{Expr}} e : \texttt{int}$
alors $\Gamma \vdash_{\text{Stat}} (\texttt{ECHO}\ e) : \texttt{void}$

(SET) si $\Gamma \vdash_{\text{Lval}} e_1 : t$ et si $\Gamma \vdash_{\text{Expr}} e_2 : t$
alors $\Gamma \vdash_{\text{Stat}} (\texttt{SET}\ e_1\ e_2) : \texttt{void}$

(IF) si $\Gamma \vdash_{\text{Expr}} e : \texttt{bool}$, si $\Gamma \vdash_{\text{Block}} bk_1 : \texttt{void}$ et si $\Gamma \vdash_{\text{Block}} bk_2 : \texttt{void}$
alors $\Gamma \vdash_{\text{Stat}} (\texttt{IF}\ e\ bk_1\ bk_2) : \texttt{void}$

(WHILE) si $\Gamma \vdash_{\text{Expr}} e : \texttt{bool}$, si $\Gamma \vdash_{\text{Block}} bk : \texttt{void}$
alors $\Gamma \vdash_{\text{Stat}} (\texttt{WHILE}\ e\ bk) : \texttt{void}$

(CALL) si $\Gamma(x) = t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ \texttt{void}$, si $\Gamma \vdash_{\text{Expar}} e_1 : t_1$, …, si $\Gamma \vdash_{\text{Expar}} e_n : t_n$
alors $\Gamma \vdash_{\text{Stat}} (\texttt{CALL}\ x\ e_1 \ldots e_n) : \texttt{void}$

lvalue

(LVAR) si $\Gamma(x) = (\texttt{ref}\ t)$
alors $\Gamma \vdash_{\text{Lval}} x : t$

(LNTH) si $\Gamma \vdash_{\text{Expr}} e_1 : (\texttt{vec}\ t)$ et $\Gamma \vdash_{\text{Expr}} e_2 : \texttt{int}$
alors $\Gamma \vdash_{\text{Lval}} (\texttt{nth}\ e_1\ e_2) : t$

Paramètres d'appel

(REF) si $\Gamma(x) = (\texttt{ref}\ t)$
alors $\Gamma \vdash_{\text{Expar}} (\texttt{adr}\ x) : (\texttt{ref}\ t)$

(VAL) si $e \in \text{Expr}$, si $\Gamma \vdash_{\text{Expr}} e : t$
alors $\Gamma \vdash_{\text{Expar}} e : t$

Expressions

(NUM) si $n \in \texttt{num}$
alors $\Gamma \vdash_{\text{Expr}} n : \texttt{int}$

(IDV) si $x \in \mathsf{ident}$, si $\Gamma(x) = t$ avec $t \neq (\texttt{ref}\ t')$
alors $\Gamma \vdash_{\text{Expr}} x : t$

(IDR) si $x \in \mathsf{ident}$, si $\Gamma(x) = (\texttt{ref}\ t)$
alors $\Gamma \vdash_{\text{Expr}} x : t$

(IF) si $\Gamma \vdash_{\text{Expr}} e_1 : \texttt{bool}$, si $\Gamma \vdash_{\text{Expr}} e_2 : t$, si $\Gamma \vdash_{\text{Expr}} e_3 : t$
alors $\Gamma \vdash_{\text{Expr}} (\texttt{if}\ e_1\ e_2\ e_3) : t$

(APP) si $\Gamma \vdash_{\text{Expr}} e : (t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t)$,
si $\Gamma \vdash_{\text{Expr}} e_1 : t_1$, …, si $\Gamma \vdash_{\text{Expr}} e_n : t_n$
alors $\Gamma \vdash_{\text{Expr}} (e\ e_1 \ldots e_n) : t$

(ABS) si $\Gamma[x_1 : t_1; \ldots; x_n : t_n] \vdash_{\text{Expr}} e : t$
alors $\Gamma \vdash_{\text{Expr}} [x_1 : t_1, \ldots, x_n : t_n] e : (t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t)$

(ALLOC) si $\Gamma \vdash_{\text{Expr}} e : \texttt{int}$
alors $\Gamma \vdash_{\text{Expr}} (\texttt{alloc}\ e) : (\texttt{vec}\ t)$

(LEN) si $\Gamma \vdash_{\text{Expr}} e : (\texttt{vec}\ t)$
alors $\Gamma \vdash_{\text{Expr}} (\texttt{len}\ e) : \texttt{int}$

(NTH) si $\Gamma \vdash_{\text{Expr}} e_1 : (\texttt{vec}\ t)$ et si $\Gamma \vdash_{\text{Expr}} e_2 : \texttt{int}$
alors $\Gamma \vdash_{\text{Expr}} (\texttt{nth}\ e_1\ e_2) : t$

(VSET) si $\Gamma \vdash_{\text{Expr}} e_1 : (\text{vec}\ t)$, si $\Gamma \vdash_{\text{Expr}} e_1 : \text{int}$ et si $\Gamma \vdash_{\text{Expr}} e_3 : t$ alors $\Gamma \vdash_{\text{Expr}} (\text{vset}\ e_1\ e_2\ e_3) : (\text{vec}\ t)$

4.3 Sémantique

Programmes

(PROG) si $\varepsilon, \varepsilon, \varepsilon \vdash_{\text{Block}} bk \rightsquigarrow \omega$
alors $\vdash bk \rightsquigarrow (\sigma, \omega)$

Blocs

BLOCK si $\rho, \sigma, \omega \vdash_{\text{CMDS}} \texttt{cs} \rightsquigarrow (\sigma', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{BLOCK}} \texttt{[}cs\texttt{]} \rightsquigarrow (\sigma', \omega')$.

Suites de commandes

(DECS) si $\rho, \sigma \vdash_{\text{DEF}} d \rightsquigarrow (\rho', \sigma')$
et si $\rho', \sigma', \omega \vdash_{\text{CMDS}} cs \rightsquigarrow (\sigma'', \omega')$
alors $\rho, \omega \vdash_{\text{CMDS}} (d\texttt{;}\ cs) \rightsquigarrow (\sigma'', \omega')$

(STATS) si $\rho, \sigma, \omega \vdash_{\text{STAT}} s \rightsquigarrow (\sigma', \omega')$
et si $\rho, \sigma', \omega' \vdash_{\text{CMDS}} cs \rightsquigarrow (\sigma'', \omega'')$
alors $\rho, \sigma, \omega \vdash_{\text{CMDS}} (s\texttt{;}\ cs) \rightsquigarrow (\sigma'', \omega'')$

(END) si $\rho, \sigma, \omega \vdash_{\text{STAT}} s \rightsquigarrow (\sigma', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{CMDS}} (s) \rightsquigarrow (\sigma', \omega')$

Définitions Soit $p1, \ldots, p_n \in$ ARGSP.
Posons $X([p_1 : t_1, \ldots, p_n : t_n]) = [x_1, \ldots, x_n]$ avec
$$x_i = \begin{cases} x_i & \text{si } p_i = x_i \\ x_i & \text{si } p_i = \texttt{var}\ x_i \end{cases}$$

(CONST) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow (v, \sigma')$
alors $\rho, \sigma \vdash_{\text{DEF}} (\texttt{CONST}\ x\ t\ e) \rightsquigarrow (\rho[x = v], \sigma')$

(FUN) $\rho, \sigma \vdash_{\text{DEF}} (\texttt{FUN}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,} \ldots \texttt{,}x_n\texttt{:}t_n\texttt{]}\ e) \rightsquigarrow (\rho[x = inF(e, (x_1; \ldots; x_n), \rho), \sigma)$

(FUNREC) $\rho, \sigma \vdash_{\text{DEF}} (\texttt{FUN REC}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,} \ldots \texttt{,}x_n\texttt{:}t_n\texttt{]}\ e)$
$\rightsquigarrow (\rho[x = inFR(e, x, (x_1; \ldots; x_n)\rho), \sigma)$

(VAR) si $alloc(\sigma) = (a, \sigma')$, avec $\sigma' = \sigma[a = any]$ et $a \notin \mathsf{dom}(\sigma)$
alors $\rho, \sigma \vdash_{\text{DEF}} (\texttt{VAR}\ x\ t) \rightsquigarrow (\rho[x = inA(a)], \sigma')$

(PROC) $\rho, \sigma \vdash_{\text{DEF}} (\texttt{PROC}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,} \ldots x_n\texttt{:}t_n\texttt{]}\ bk) \rightsquigarrow (\rho[x = inP(bk, (x_1; \ldots; x_n), \rho)], \sigma)$

(PROCREC) $\rho, \sigma \vdash_{\text{DEF}} (\texttt{PROC REC}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,} \ldots \texttt{,}x_n\texttt{:}t_n\texttt{]}bk) \rightsquigarrow (\rho[x = inPR(inP(bk, x, (x_1; \ldots; x_n), \rho), \sigma)$

Instructions

(SET) si $\rho, \sigma \vdash_{\text{LVAL}} e_1 \rightsquigarrow a$ et si $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow (v, \sigma')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{SET}\ e_1\ e_2) \rightsquigarrow (\sigma'[a := v], \omega)$

(IF1) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow (inZ(1), \sigma')$ et si $\rho, \sigma', \omega \vdash_{\text{BLOCK}} bk_1 \rightsquigarrow (\sigma'', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{IF}\ e\ bk_1\ bk_2) \rightsquigarrow (\sigma'', \omega')$

(IF0) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow (inZ(0), \sigma')$ et si $\rho, \sigma', \omega \vdash_{\text{BLOCK}} bk_2 \rightsquigarrow (\sigma'', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{IF}\ e\ bk_1\ bk_2) \rightsquigarrow (\sigma'', \omega')$

(LOOP0) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow (inZ(0), \sigma')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{WHILE}\ e\ bk) \rightsquigarrow (\sigma', \omega)$

(LOOP1) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow (inZ(1), \sigma')$, si $\rho, \sigma', \omega \vdash_{\text{BLOCK}} bk \rightsquigarrow (\sigma'', \omega')$ et si $\rho, \sigma'', \omega' \vdash_{\text{STAT}} (\texttt{WHILE}\ e\ bk) \rightsquigarrow (\sigma''', \omega'')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{WHILE}\ e\ bk) \rightsquigarrow (\sigma''', \omega'')$

(CALL) si $\rho(x) = inP(bk, (x_1; \ldots; x_n), \rho')$,
si $\rho, \sigma \vdash_{\text{EXPAR}} e_1 \rightsquigarrow v_1$, ..., si $\rho, \sigma \vdash_{\text{EXPAR}} e_n \rightsquigarrow v_n$
si $\rho'[x_1 = v_1; \ldots; x_n = v_n), \sigma, \omega \vdash_{\text{BLOCK}} bk \rightsquigarrow (\sigma', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{CALL}\ x\ e_1 \ldots e_n) \rightsquigarrow (\sigma', \omega')$

(CALLR) si $\rho(x) = inPR(bk, x, (x_1; \ldots; \rho')$,
si $\rho, \sigma \vdash_{\text{EXPAR}} e_1 \rightsquigarrow v_1$, ..., si $\rho, \sigma \vdash_{\text{EXPAR}} e_n \rightsquigarrow v_n$
et si $\rho'[x_1 = v_1; \ldots; x_n = v_n][x = inPR(bk, x, (x_1; \ldots; x_n), \rho')], \sigma, \omega \vdash_{\text{BLOCK}} bk \rightsquigarrow (\sigma', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\texttt{CALL}\ x\ e_1 \ldots e_n) \rightsquigarrow (\sigma', \omega')$

(ECHO) si $\rho, \sigma, \omega \vdash_{\text{Expr}} e \rightsquigarrow (inZ(n), \sigma')$
alors $\rho, \sigma, \omega \vdash_{\text{Stat}} (\texttt{ECHO}\ e) \rightsquigarrow (\sigma', n \cdot \omega)$

lvalue $\rho, \sigma \vdash_{\text{Lval}} lv \rightsquigarrow (a, \sigma')$

(LID) si $x \in ident$, si $\rho(x) = inA(a)$ alors $\rho, \sigma \vdash_{\text{Lval}} x \rightsquigarrow (a, \sigma)$

(LNTH1) si $x \in \textsf{ident}$, si $\rho(x) = inB(a, n)$ et si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow (inZ(i), \sigma')$
alors $\rho, \sigma \vdash_{\text{Lval}} (\text{nth}\ x\ e) \rightsquigarrow (a + i, \sigma')$

(LNTH2) si $\rho, \sigma \vdash_{\text{Lval}} e_1 \rightsquigarrow (a_1, \sigma')$ avec $\sigma'(a_1) = inB(a_2, \_)$ et si $\rho, \sigma' \vdash_{\text{Expr}} e \rightsquigarrow (inZ(i), \sigma'')$
alors $\rho, \sigma \vdash_{\text{Lval}} (\text{nth}\ lv\ e) \rightsquigarrow (a_2 + i, \sigma'')$

Paramètres d'appel

(REF) si $\rho(x) = inA(a)$
alors $\rho, \sigma \vdash_{\text{Expar}} (\texttt{adr}\ x) \rightsquigarrow (inA(a), \sigma)$

(VAL) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow (v, \sigma')$
alors $\rho, \sigma \vdash_{\text{Expar}} e \rightsquigarrow (v, \sigma')$

Expressions

(TRUE) $\rho, \sigma \vdash_{\text{Expr}} \texttt{true} \rightsquigarrow (inZ(1), \sigma)$

(FALSE) $\rho, \sigma \vdash_{\text{Expr}} \texttt{false} \rightsquigarrow (inZ(0), \sigma)$

(NUM) si $n \in \texttt{num}$ alors $\rho, \sigma \vdash_{\text{Expr}} n \rightsquigarrow (inZ(\nu(n)), \sigma)$

(ID1) si $x \in \textsf{ident}$ et $\rho(x) = inA(a)$
alors $\rho, \sigma \vdash_{\text{Expr}} x \rightsquigarrow (inZ(\sigma(a)), \sigma)$

(ID2) si $x \in \textsf{ident}$ et si $\rho(x) = v$ et $v \neq inA(a)$
alors $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow (v, \sigma)$

(PRIM1) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow inZ(n)$, et si $\pi_1(not)(n) = n'$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{not}\ e) \rightsquigarrow (inZ(n'), \sigma')$

(PRIM2) si $x \in \{\texttt{eq lt add sub mul div}\}$,
si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(n_1), \sigma')$, si $\rho, \sigma' \vdash_{\text{Expr}} e_2 \rightsquigarrow (inZ(n_2); \sigma'')$ et si $\pi_2(x)(n_1, n_2) = n$
alors $\rho, \sigma \vdash_{\text{Expr}} (x\ e_1 e_2) \rightsquigarrow (inZ(n), \sigma'')$

(AND0) si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(0), \sigma')$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{and}\ e_1\ e_2) \rightsquigarrow (inZ(0), \sigma')$.

(AND1) si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(1), \sigma')$ et si $\rho, \sigma' \vdash_{\text{Expr}} e_2 \rightsquigarrow (v, \sigma'')$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{and}\ e_1\ e_2) \rightsquigarrow (v, \sigma'')$.

(OR1) si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(1), \sigma')$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{or}\ e_1\ e_2) \rightsquigarrow (inZ(1), \sigma')$.

(OR0) si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(0)\sigma')$ et si $\rho, \sigma' \vdash_{\text{Expr}} e_2 \rightsquigarrow (v, \sigma'')$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{or}\ e_1\ e_2) \rightsquigarrow (v, \sigma'')$.

(IF1) si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(1), \sigma')$ et si $\rho, \sigma' \vdash_{\text{Expr}} e_2 \rightsquigarrow (v, \sigma'')$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{if}\ e_1\ e_2\ e_3) \rightsquigarrow (v, \sigma'')$

(IF0) si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow (inZ(0), \sigma')$ et si $\rho, \sigma' \vdash_{\text{Expr}} e_3 \rightsquigarrow (v, \sigma'')$
alors $\rho, \sigma \vdash_{\text{Expr}} (\texttt{if}\ e_1\ e_2\ e_3) \rightsquigarrow (v, \sigma'')$

(ABS) $\rho, \sigma \vdash_{\text{Expr}} \texttt{[}x_1 \texttt{:} t_1 \texttt{,} \ldots \texttt{,} x_n \texttt{:} t_n\texttt{]} e \rightsquigarrow (inF(e, (x_1, \ldots, x_n), \rho), \sigma)$

(APP) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow inF(e', (x_1; \ldots; x_n), \rho')$, si $\rho, \sigma \vdash_{\text{Expr}} e_1 \rightsquigarrow v_1$, ..., si $\rho, \sigma \vdash_{\text{Expr}} e_n \rightsquigarrow v_n$,
si $\rho'[x_1 = v_1; \ldots; x_n = v_n], \sigma \vdash_{\text{Expr}} e' \rightsquigarrow v$
alors $\rho, \sigma \vdash \texttt{(}e\ e_1 \ldots e_n\texttt{)} \rightsquigarrow v$

(APPR) si $\rho, \sigma \vdash_{\text{EXPR}} e \leadsto inFR(e', x, (x_1; \ldots; x_n), \rho')$,
si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \leadsto v_1$, ..., si $\rho, \sigma \vdash_{\text{EXPR}} e_n \leadsto v_n$,
si $\rho'[x_1 = v_1; \ldots; x_n = v_n][x = inFR(e', x, (x_1; \ldots; x_n), \rho')], \sigma \vdash_{\text{EXPR}} e' \leadsto v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (e\ e_1 \ldots e_n) \leadsto v$

(ALLOC) si $\rho, \sigma \vdash_{\text{EXPR}} e \leadsto (inZ(n), \sigma')$, avec $n > 0$, et si $allocb(\sigma', n) = (a, \sigma'')$, avec $\sigma'' = \sigma'[a = inZ(n)]$,
alors $\rho, \sigma \vdash_{\text{EXPR}}$ `(alloc` e`)` $\leadsto (inB(a), \sigma'')$

(LEN) si $\rho, \sigma \vdash_{\text{EXPR}} e \leadsto (inB(a), \sigma')$
alors $\rho, \sigma \vdash_{\text{EXPR}}$ `(len` e`)` $\leadsto (\sigma'(a), \sigma')$

(NTH) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \leadsto (inB(a), \sigma')$ et si $\rho, \sigma' \vdash_{\text{EXPR}} e_2 \leadsto (inZ(i), \sigma'')$
alors $\rho, \sigma \vdash_{\text{EXPR}}$ `(nth` $e_1\ e_2$`)` $\leadsto (\sigma''(a + i + 1), \sigma'')$

(VSET) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \leadsto (inB(a, n), \sigma')$, si $\rho, \sigma' \vdash_{\text{EXPR}} e_2 \leadsto (inZ(i), \sigma'')$ et si $\rho, \sigma'' \vdash_{\text{EXPR}} e_3 \leadsto (v, \sigma''')$
alors $\rho, \sigma \vdash_{\text{EXPR}}$ (vset $lv\ e_1\ e_2$) $\leadsto (inB(a, n), \sigma'''[a + i := v])$