

/SU/FSI/MASTER/INFO/MU4IN503

APS

Formulaire

P. MANOURY *

Janvier 2022

2 APS1 : noyau impératif

2.1 Syntaxe

Lexique

Symboles réservés

[] () ; : , * ->

Mots clef

CONST FUN REC **VAR** **PROC**

ECHO **SET** IF **WHILE** **CALL**

if and or

bool int

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 TYPE
| PROC ident [ARGS] BLOCK
| PROC REC ident [ARGS] BLOCK

Type

TYPE ::= bool | int
| (TYPES -> TYPE)
TYPES ::= TYPE
| TYPE * TYPES

Paramètre formel

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

Instruction

STAT ::= ECHO EXPR
| SET ident EXPR
| IF EXPR BLOCK BLOCK
| WHILE EXPR BLOCK
| CALL ident EXPRS

Expression

EXPR ::= num
| ident
| (if EXPR EXPR EXPR)
| (and EXPR EXPR)
| (or EXPR EXPR)
| (EXPR EXPRS)
| [ARGS] EXPR

Suite d'expressions

EXPRS ::= EXPR
| EXPR EXPRS

2.2 Typage

Programmes

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

Blocs

(BLOCK) si $\Gamma \vdash_{\text{CMDs}} (cs; \varepsilon) : \text{void}$
alors $\Gamma \vdash_{\text{BLOCK}} [cs] : \text{void}$

Suite de commandes

(DECS) si $d \in \text{DEC}$, si $\Gamma \vdash_{\text{DEF}} d : \Gamma'$, si $\Gamma' \vdash_{\text{CMDs}} cs : \text{void}$
alors $\Gamma \vdash_{\text{CMDs}} (d; cs) : \text{void}$.
(STATS) si $s \in \text{STAT}$, si $\Gamma \vdash_{\text{STAT}} s : \text{void}$, si $\Gamma \vdash_{\text{CMDs}} cs : \text{void}$
alors $\Gamma \vdash_{\text{CMDs}} (s; cs) : \text{void}$.
(END) $\Gamma \vdash_{\text{CMDs}} \varepsilon : \text{void}$.

Définitions

- (CONST) si $\Gamma \vdash_{\text{EXPR}} e : t$
alors $\Gamma \vdash_{\text{DEF}} (\text{CONST } x \ t \ e) : \Gamma[x : t]$
- (FUN) si $\Gamma[x_1 : t_1; \dots; x_n : t_n] \vdash_{\text{EXPR}} e : t$
alors $\Gamma \vdash_{\text{DEF}} (\text{FUN } x \ t \ [x_1 : t_1, \dots, x_n : t_n] \ e) : \Gamma[x : (t_1 * \dots * t_n \rightarrow t)]$
- (FUNREC) si $\Gamma[x_1 : t_1; \dots; x_n : t_n; x : t_1 * \dots * t_n \rightarrow t] \vdash_{\text{EXPR}} e : t$
alors $\Gamma \vdash_{\text{DEF}} (\text{FUN REC } x \ t \ [x_1 : t_1, \dots, x_n : t_n] \ e) : \Gamma[x : t_1 * \dots * t_n \rightarrow t]$
- (VAR) si $t \in \{\text{int}, \text{bool}\}$
alors $\Gamma \vdash_{\text{DEF}} (\text{VAR } x \ t) : \Gamma[x : t]$
- (PROC) si $\Gamma[x_1 : t_1; \dots; x_n : t_n] \vdash_{\text{BLOCK}} bk : \text{void}$
alors $\Gamma \vdash_{\text{DEF}} (\text{PROC } x \ [x_1 : t_1, \dots, x_n : t_n] \ bk) : \Gamma[x : t_1 * \dots * t_n \rightarrow \text{void}]$
- (PROCREC)
si $\Gamma[x_1 : t_1; \dots; x_n : t_n; x : t_1 * \dots * t_n \rightarrow \text{void}] \vdash_{\text{BLOCK}} bk : \text{void}$
alors $\Gamma \vdash_{\text{DEF}} (\text{PROC REC } x \ [x_1 : t_1, \dots, x_n : t_n] \ bk) : \Gamma[x : t_1 * \dots * t_n \rightarrow \text{void}]$

Intructions

- (ECHO) si $\Gamma \vdash_{\text{EXPR}} e : \text{int}$
alors $\Gamma \vdash_{\text{STAT}} (\text{ECHO } e) : \text{void}$
- (SET) si $\Gamma(x) = t$ et si $\Gamma \vdash_{\text{EXPR}} e : t$
alors $\Gamma \vdash_{\text{STAT}} (\text{SET } x \ e) : \text{void}$
- (IF) si $\Gamma \vdash_{\text{EXPR}} e : \text{bool}$, si $\Gamma \vdash_{\text{BLOCK}} bk_1 : \text{void}$ et si $\Gamma \vdash_{\text{BLOCK}} bk_2 : \text{void}$
alors $\Gamma \vdash_{\text{STAT}} (\text{IF } e \ bk_1 \ bk_2) : \text{void}$
- (WHILE) si $\Gamma \vdash_{\text{EXPR}} e : \text{bool}$, si $\Gamma \vdash_{\text{BLOCK}} bk : \text{void}$
alors $\Gamma \vdash_{\text{STAT}} (\text{WHILE } e \ bk) : \text{void}$
- (CALL) si $\Gamma(x) = t_1 * \dots * t_n \rightarrow \text{void}$, si $\Gamma \vdash_{\text{EXPR}} e_1 : t_1, \dots$, si $\Gamma \vdash_{\text{EXPR}} e_n : t_n$
alors $\Gamma \vdash_{\text{STAT}} (\text{CALL } x \ e_1 \dots e_n) : \text{void}$

Expressions

- (NUM) si $n \in \text{num}$
alors $\Gamma \vdash_{\text{EXPR}} n : \text{int}$
- (ID) si $x \in \text{ident}$, si $\Gamma(x) = t$
alors $\Gamma \vdash_{\text{EXPR}} x : t$
- (IF) si $\Gamma \vdash_{\text{EXPR}} e_1 : \text{bool}$, si $\Gamma \vdash_{\text{EXPR}} e_2 : t$, si $\Gamma \vdash_{\text{EXPR}} e_3 : t$
alors $\Gamma \vdash_{\text{EXPR}} (\text{if } e_1 \ e_2 \ e_3) : t$
- (APP) si $\Gamma \vdash_{\text{EXPR}} e : (t_1 * \dots * t_n \rightarrow t)$,
si $\Gamma \vdash_{\text{EXPR}} e_1 : t_1, \dots$, si $\Gamma \vdash_{\text{EXPR}} e_n : t_n$
alors $\Gamma \vdash_{\text{EXPR}} (e \ e_1 \dots e_n) : t$
- (ABS) si $\Gamma[x_1 : t_1; \dots; x_n : t_n] \vdash_{\text{EXPR}} e : t$
alors $\Gamma \vdash_{\text{EXPR}} [x_1 : t_1, \dots, x_n : t_n] e : (t_1 * \dots * t_n \rightarrow t)$

2.3 Sémantique

Programmes $\vdash p \rightsquigarrow (\sigma, \omega)$

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

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

Suites de commandes $\rho, \sigma, \omega \vdash_{\text{CMDS}} cs \rightsquigarrow (\sigma', \omega')$
 (DECS) si $\rho, \sigma \vdash_{\text{DEF}} d \rightsquigarrow (\rho', \sigma')$ et si $\rho', \sigma', \omega \vdash_{\text{CMDS}} cs \rightsquigarrow (\sigma'', \omega')$
 alors $\rho, \sigma, \omega \vdash_{\text{CMDS}} (d; cs) \rightsquigarrow (\sigma'', \omega')$
 (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 $\rho, \sigma, \omega \vdash_{\text{CMDS}} (s; cs) \rightsquigarrow (\sigma'', \omega'')$
 (END) si $s \in \text{STAT}$, si $\rho, \sigma, \omega \vdash_{\text{STAT}} s \rightsquigarrow \omega'$
 alors $\rho, \sigma, \omega \vdash_{\text{CMDS}} (s) \rightsquigarrow \omega'$

Définitions $\rho, \sigma \vdash_{\text{DEF}} d \rightsquigarrow (\rho', \sigma')$
 (CONST) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow v$
 alors $\rho, \sigma \vdash_{\text{DEF}} (\text{CONST } x \ t \ e) \rightsquigarrow (\rho[x = v], \sigma)$
 (FUN) $\rho \vdash_{\text{DEF}} (\text{FUN } x \ t \ [x_1:t_1, \dots, x_n:t_n] \ e) \rightsquigarrow (\rho[x = \text{inF}(e, (x_1; \dots; x_n), \rho)], \sigma)$
 (FUNREC) $\rho \vdash_{\text{DEF}} (\text{FUN REC } x \ t \ [x_1:t_1, \dots, x_n:t_n] \ e) \rightsquigarrow \rho[x = \text{inFR}(e, x, (x_1; \dots; x_n), \rho)]$
 (VAR) si $\text{alloc}(\sigma) = (a, \sigma')$, avec $\sigma' = \sigma[a = \text{any}]$ et $a \notin \text{dom}(\sigma)$
 alors $\rho, \sigma \vdash_{\text{DEF}} (\text{VAR } x \ t) \rightsquigarrow (\rho[x = \text{inA}(a)], \sigma')$
 (PROC) $\rho, \sigma \vdash_{\text{DEF}} (\text{PROC } x \ t \ [x_1:t_1, \dots, x_n:t_n] \ bk) \rightsquigarrow (\rho[x = \text{inP}(bk, (x_1; \dots; x_n), \rho)], \sigma)$
 (PROCREC) $\rho, \sigma \vdash_{\text{DEF}} (\text{PROC REC } x \ t \ [x_1:t_1, \dots, x_n:t_n] \ bk) \rightsquigarrow (\rho[x = \text{inPR}(\text{inP}(bk, x, (x_1; \dots; x_n), \rho), \sigma)]$

Instructions $\rho, \sigma, \omega \vdash_{\text{STAT}} s \rightsquigarrow (\sigma', \omega')$
 (SET) si $\rho(x) = \text{inA}(a)$ et si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow v$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{SET } x \ e) \rightsquigarrow (\sigma[a := v], \omega)$
 (IF1) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inZ}(1)$ et si $\rho, \sigma, \omega \vdash_{\text{BLOCK}} bk_1 \rightsquigarrow (\sigma', \omega')$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{IF } e \ bk_1 \ bk_2) \rightsquigarrow (\sigma', \omega')$
 (IF0) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inZ}(0)$ et si $\rho, \sigma, \omega \vdash_{\text{BLOCK}} bk_2 \rightsquigarrow (\sigma', \omega')$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{IF } e \ bk_1 \ bk_2) \rightsquigarrow (\sigma', \omega')$
 (LOOP0) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inZ}(0)$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{WHILE } e \ bk) \rightsquigarrow (\sigma, \omega)$
 (LOOP1) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inZ}(1)$,
 si $\rho, \sigma, \omega \vdash_{\text{BLOCK}} bk \rightsquigarrow (\sigma', \omega')$ et $\rho, \sigma', \omega' \vdash_{\text{STAT}} (\text{WHILE } e \ bk) \rightsquigarrow (\sigma'', \omega'')$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{WHILE } e \ bk) \rightsquigarrow (\sigma'', \omega'')$
 (CALL) si $\rho(x) = \text{inP}(bk, (x_1; \dots; x_n), \rho')$,
 si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow v_1, \dots, \text{si } \rho, \sigma \vdash_{\text{EXPR}} e_n \rightsquigarrow v_n$
 si $\rho'[x_1 = v_1; \dots; x_n = v_n], \sigma, \omega \vdash_{\text{BLOCK}} bk \rightsquigarrow (\sigma', \omega')$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{CALL } x \ e_1 \dots e_n) \rightsquigarrow (\sigma', \omega')$
 (CALLR) si $\rho(x) = \text{inPR}(bk, x, (x_1; \dots; \rho'))$,
 si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow v_1, \dots, \text{si } \rho, \sigma \vdash_{\text{EXPR}} e_n \rightsquigarrow v_n$
 et si $\rho'[x_1 = v_1; \dots; x_n = v_n][x = \text{inPR}(bk, x, (x_1; \dots; x_n), \rho')], \sigma, \omega \vdash_{\text{BLOCK}} bk \rightsquigarrow (\sigma', \omega')$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{CALL } x \ e_1 \dots e_n) \rightsquigarrow (\sigma', \omega')$
 (ECHO) si $\rho, \sigma, \omega \vdash_{\text{EXPR}} e \rightsquigarrow \text{inZ}(n)$
 alors $\rho, \sigma, \omega \vdash_{\text{STAT}} (\text{ECHO } e) \rightsquigarrow (\sigma, n \cdot \omega)$

Expressions $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow v$

- (TRUE) $\rho, \sigma \vdash_{\text{EXPR}} \mathbf{true} \rightsquigarrow \text{inZ}(1)$
- (FALSE) $\rho, \sigma \vdash_{\text{EXPR}} \mathbf{false} \rightsquigarrow \text{inZ}(0)$
- (NUM) si $n \in \mathbf{num}$ alors $\rho, \sigma \vdash_{\text{EXPR}} n \rightsquigarrow \text{inZ}(\nu(n))$
- (ID1) si $\rho(x) = \text{inA}(a)$
alors $\rho, \sigma \vdash_{\text{EXPR}} x \rightsquigarrow \text{inZ}(\sigma(a))$
- (ID2) si $\rho(x) = v$ et $v \neq \text{inA}(a)$
alors $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow v$
- (PRIM1) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inZ}(n)$, et si $\pi_1(\mathbf{not})(n) = n'$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{not} \ e) \rightsquigarrow \text{inZ}(n')$
- (PRIM2) si $x \in \{\mathbf{eq} \ \mathbf{lt} \ \mathbf{add} \ \mathbf{sub} \ \mathbf{mul} \ \mathbf{div}\}$,
si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(n_1)$, si $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow \text{inZ}(n_2)$ et si $\pi_2(x)(n_1, n_2) = n$
alors $\rho, \sigma \vdash_{\text{EXPR}} (x \ e_1 e_2) \rightsquigarrow \text{inZ}(n)$
- (AND1) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(1)$ et si $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{and} \ e_1 \ e_2) \rightsquigarrow v$.
- (AND0) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(0)$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{and} \ e_1 \ e_2) \rightsquigarrow \text{inZ}(0)$.
- (OR1) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(1)$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{or} \ e_1 \ e_2) \rightsquigarrow \text{inZ}(1)$.
- (OR0) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(0)$ et si $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{or} \ e_1 \ e_2) \rightsquigarrow v$.
- (IF1) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(1)$ et si $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{if} \ e_1 \ e_2 \ e_3) \rightsquigarrow v$
- (IF0) si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow \text{inZ}(0)$ et si $\rho, \sigma \vdash_{\text{EXPR}} e_3 \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (\mathbf{if} \ e_1 \ e_2 \ e_3) \rightsquigarrow v$
- (ABS) $\rho, \sigma \vdash_{\text{EXPR}} [x_1:t_1, \dots, x_n:t_n]e \rightsquigarrow \text{inF}(e, (x_1; \dots; x_n), \rho)$
- (APP) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inF}(e', (x_1; \dots; x_n), \rho')$,
si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow v_1, \dots, \text{si } \rho, \sigma \vdash_{\text{EXPR}} e_n \rightsquigarrow v_n$,
si $\rho'[x_1 = v_1; \dots; x_n = v_n], \sigma \vdash_{\text{EXPR}} e' \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (e \ e_1 \dots e_n) \rightsquigarrow v$
- (APPR) si $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow \text{inFR}(e', x, (x_1; \dots; x_n), \rho')$,
si $\rho, \sigma \vdash_{\text{EXPR}} e_1 \rightsquigarrow v_1, \dots, \text{si } \rho, \sigma \vdash_{\text{EXPR}} e_n \rightsquigarrow v_n$,
si $\rho'[x_n = v_1, \dots, x_n = v_n, x = \text{inFR}(e', x, (x_1; \dots; x_n), \rho')], \sigma \vdash_{\text{EXPR}} e' \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{EXPR}} (e \ e_1 \dots e_n) \rightsquigarrow v$