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

P. Manoury*

Janvier 2022

3 APS1a

3.1 Syntaxe

Lexique

Symboles réservés

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

Mots clef

`CONST FUN REC VAR PROC   ECHO SET IF WHILE CALL`
`if and or`
`bool int`
`var adr`

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 [ARGSP] BLOCK
	|	PROC REC ident [ARGSP] BLOCK

Type

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

Paramètre formel (fonctions)

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

Paramètre formel (procédure)

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

Instruction

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

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

Suite d'expressions

EXPRS	::=	EXPR
	|	EXPR EXPRS

3.2 Typage

Soit $p1, \ldots, p_n \in$ ARGSP.
Posons $A([p_1 : t_1, \ldots, p_n : t_n]) = [x_1 : t'_1, \ldots, x_n : t'_n]$ 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 $d \in \text{DEC}$, 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 $s \in \text{STAT}$, 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 $\Gamma \vdash_{\text{DEF}} (\texttt{FUN}\ x\ t\ \texttt{[}x_1\texttt{:}t_1\texttt{,}\ldots\texttt{,}x_n\texttt{:}t_n\texttt{]}\ e) : \Gamma[x : \texttt{(}t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t\texttt{)}]$

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

Intructions

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

(SET) si $\Gamma(x) = (\texttt{ref}\ t)$ et si $\Gamma \vdash_{\text{EXPR}} e : t$
alors $\Gamma \vdash_{\text{STAT}} (\texttt{SET}\ x\ e) : \texttt{void}$

(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 $\Gamma \vdash_{\text{STAT}} (\texttt{CALL}\ x\ e_1 \ldots e_n) : \texttt{void}$

Paramètres d'appel

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

(VAL) si $e \in$ 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 $\Gamma \vdash_{\text{EXPR}} [x_1 : t_1, \ldots, x_n : t_n] e : (t_1\ \texttt{*}\ \ldots\ \texttt{*}\ t_n\ \texttt{->}\ t)$

3.3 Sémantique

Programmes

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

Blocs

BLOCK 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

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

(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 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 $\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow v$
alors $\rho, \sigma \vdash_{\text{DEF}} (\texttt{CONST}\ x\ t\ e) \rightsquigarrow (\rho[x = v], \sigma)$

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

(FUNREC) $\rho, \sigma \vdash_{\text{Def}}$ (FUN REC x t [x_1:t_1, ... ,x_n:t_n] 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 $\rho, \sigma \vdash_{\text{Def}}$ (VAR x t) $\rightsquigarrow (\rho[x = inA(a)], \sigma')$

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

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

Instructions

(SET) si $\rho(x) = inA(a)$ et si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow v$
alors $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (SET x e) $\rightsquigarrow (\sigma[a := v], \omega)$

(IF1) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow inZ(1)$ et si $\rho, \sigma, \omega \vdash_{\text{Block}} bk_1 \rightsquigarrow (\sigma', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (IF e bk_1 bk_2) $\rightsquigarrow (\sigma', \omega')$

(IF0) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow inZ(0)$ et si $\rho, \sigma, \omega \vdash_{\text{Block}} bk_2 \rightsquigarrow (\sigma', \omega')$
alors $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (IF e bk_1 bk_2) $\rightsquigarrow (\sigma', \omega')$

(LOOP0) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow inZ(0)$
alors $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (WHILE e bk) $\rightsquigarrow (\sigma, \omega)$

(LOOP1) si $\rho, \sigma \vdash_{\text{Expr}} e \rightsquigarrow inZ(1)$, si $\rho, \sigma, \omega \vdash_{\text{Block}} bk \rightsquigarrow (\sigma', \omega')$ et si $\rho, \sigma', \omega' \vdash_{\text{Stat}}$ (WHILE e bk) $\rightsquigarrow (\sigma'', \omega'')$
alors $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (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 $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (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 $\rho, \sigma, \omega \vdash_{\text{Stat}}$ (CALL x $e_1 \ldots e_n$) $\rightsquigarrow (\sigma', \omega')$

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

Paramètres d'appel

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

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

Expressions

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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