

/SU/FSI/MASTER/INFO/MU4IN503 (APS)
Analyse des Programmes et
Sémantique

Janvier 2021

Pascal MANOURY – Romain DEMANGEON
pascal.manoury@lip6.fr

6 : *APS2* : tableaux

Tableaux

Structure séquentielle statique

Zone mémoire contigüe

- ▶ taille fixe
- ▶ lecture/écriture indicée

(bloc mémoire)

(longueur)

(position)

APS2

Type

Syntaxe

Un nouveau type *paramétré* : `vec`

`(vec bool) (vec int) (vec (vec int))` etc.

Restriction au types représentables en mémoire

Storable types

$$\text{STYPE} ::= \text{bool} \mid \text{int} \\ \mid (\text{vec STYPE})$$
$$\text{TYPE} ::= \text{STYPE} \\ \mid (\text{TYPES} \rightarrow \text{TYPE})$$

Restriction justifiée par la sémantique

Déclaration de variable

Syntaxe

On n'admet que des types représentables en mémoire

$$\begin{array}{l} \text{DEF} \quad ::= \quad \dots \\ \quad \quad | \quad \text{VAR ident STYPE} \\ \quad \quad | \quad \dots \end{array}$$

Remarque : on aurait pu aussi le repérer au typage (cf. règle de typage (VAR) de *APS1a*)

Opérateurs

Syntaxe

Trois opérateurs primitifs : `alloc len nth` (EXPR)

Affectation

- ▶ une nouvelle expression *assignable*
- ▶ un nouveau non terminal LVALUE

```
STAT ::= ...  
      | SET LVALUE EXPR  
      | ...
```

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

Opérateurs

Typage

Opérateurs *polymorphes*

Pour tout type t :

- (ALLOC) si $\Gamma \vdash_{\text{EXPR}} e : \text{int}$
alors $\Gamma \vdash_{\text{EXPR}} (\text{alloc } e) : (\text{vec } t)$
- (LEN) si $\Gamma \vdash_{\text{EXPR}} e : (\text{vec } t)$
alors $\Gamma \vdash_{\text{EXPR}} (\text{len } e) : \text{int}$
- (NTH) si $\Gamma \vdash_{\text{EXPR}} e_1 : (\text{vec } t)$ et si $\Gamma \vdash_{\text{EXPR}} e_2 : \text{int}$
alors $\Gamma \vdash_{\text{EXPR}} (\text{nth } e_1 e_2) : t$

Affectation

Typage

Valeurs modifiables : *lvalue*

- ▶ variables de type $(\text{ref } t)$ (APS1a)
- ▶ expression $(\text{nth } e_1 \ e_2)$ lorsque e_1 est de type $(\text{vec } t)$

Nouvelle relation de typage : \vdash_{LVAL}

(SET) si $\Gamma \vdash_{\text{LVAL}} e_1 : t$ et si $\Gamma \vdash_{\text{EXPR}} e_2 : t$
alors $\Gamma \vdash_{\text{STAT}} (\text{SET } e_1 \ e_2) : \text{void}$

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

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

Domaines sémantiques

Nouvelle valeur pour les tableaux : *bloc mémoire*

- ▶ une adresse pour le début de bloc (*adresse de base*)
- ▶ *calcul d'adresse* pour les éléments du bloc :
adresse de base + décalage¹

Domaine des adresses : $A = N$ (entiers naturels)

Identifier les adresses des blocs mémoires : $B = A$

Nouveau domaine de valeurs :

$$V = Z \oplus A \oplus B \oplus F \oplus FR \oplus P \oplus PR$$

Environnement : $E = \text{ident} \rightarrow V$

Mémoire : $S = A \rightarrow (Z \oplus B)$

1. *arithmétique pointeur*

Représentation d'un bloc

Intervalle d'adresses $[a, a + 1, \dots, a + n]$

En mémoire :

- ▶ adresse a : taille du bloc ($\text{lnZ}(n)$)
- ▶ adresses $a + 1 \dots a + n$: éléments du bloc ($\text{inZ}(x)$ ou $\text{inB}(a)$)

Fonction d'allocation : $\text{allocb} : S \times N \rightarrow (A \times S)$

$$\text{allocb}(\sigma, n) = (a, \sigma')$$

si et seulement si

$$n > 0$$

et pour $i \in [0, n]$, $a + i \notin \text{dom}(\sigma)^2$

$$\text{et } \sigma' = \sigma[a = \text{lnZ}(n)]$$

Expression : alloc

Sémantique

L'expression (`alloc e`) a un effet sur la mémoire

Nouveau domaine de \vdash_{EXPR}

$$E \times S \times \text{EXPR} \times (V \times S)$$

$$\rho, \sigma \vdash_{\text{EXPR}} e \rightsquigarrow (v, \sigma')$$

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

Expression : len et nth

Sémantique

La longueur est donnée par la 1ère valeur du bloc,
à l'adresse de base du bloc :

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

Le i -ème élément est donné par décalage à partir de
l'adresse de base du bloc :

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

Défensif : $0 \leq i < \sigma(a)$

Intruction : affectation

Sémantique

Nouvelle relation pour l'adresse assignable :

\vdash_{LVAL} de domaine $E \times S \times \text{LVALUE} \times A$

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

Distinguera 4 cas :

1. identificateur
 - 1.1 de variable (déclaration VAR)
 - 1.2 de bloc (de type (vec t))
2. n-ième d'un bloc
 - 2.1 valeur immédiate ($\text{inZ}(x)$)
 - 2.2 bloc ($\text{inB}(a)$)

lvalue

Sémantique

- (LID0) si $x \in \text{ident}$, si $\rho(x) = \text{inA}(a)$
alors $\rho, \sigma \vdash_{\text{LVAL}} x \rightsquigarrow a$
- (LID1) si $x \in \text{ident}$, si $\rho(x) = \text{inB}(a)$
alors $\rho, \sigma \vdash_{\text{LVAL}} x \rightsquigarrow a + 1$
- (LNTH1) si $\rho, \sigma \vdash_{\text{LVAL}} e_1 \rightsquigarrow a$, si
 $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow (\text{inZ}(i), \sigma')$
et si $\sigma'(a + i) = \text{InZ}(n)$
alors $\rho, \sigma \vdash_{\text{LVAL}} (\text{nth } e_1 \ e_2) \rightsquigarrow (a + i)$
- (LNTH2) si $\rho, \sigma \vdash_{\text{LVAL}} e_1 \rightsquigarrow a$, si
 $\rho, \sigma \vdash_{\text{EXPR}} e_2 \rightsquigarrow (\text{inZ}(i), \sigma')$
et si $\sigma'(a + i) = \text{InB}(a')$
alors $\rho, \sigma \vdash_{\text{LVAL}} (\text{nth } e_1 \ e_2) \rightsquigarrow (a' + 1)$

Discussion

Des tableaux (de tableaux)*

Les tableaux sont passés par référence

Les fonctions ne calculent pas de tableau