
Programmation – Spécifications formelles

II. Spécification ensemblistes – Z

P. MANOURY

Nov. - Déc. *2001*

Le langage des ensembles

Rapide rappels

Appartenance (abstraite) : $x \in y$

Égalité (axiome) : $x = y \Leftrightarrow (\forall z.z \in x \Leftrightarrow z \in y)$

Inclusion (définie) : $x \subseteq y \hat{=} (\forall z.z \in x \Rightarrow z \in y)$

Constructions de base et axiomes :

	notation	axiome
Ensemble vide :	\emptyset	$\forall x.x \notin \emptyset$
Couples	(x, y)	$(x, y) = (x', y') \Leftrightarrow x = x' \wedge y = y'$
Produit	$X \times Y$	$z \in X \times Y \Leftrightarrow \exists x \in X. \exists y \in Y. z = (x, y)$
Union	$X \cup Y$	$z \in X \cup Y \Leftrightarrow z \in X \vee z \in Y$
Ensemble des parties	$\mathcal{P}(X)$	$z \in \mathcal{P}(X) \Leftrightarrow z \subseteq X$
Schéma de compréhension	$\{x \in X \mid \varphi\}$	$y \in \{x \in X \mid \varphi\} \Leftrightarrow y \in X \wedge \varphi[y/x]$

Construtions définies :

	notation	définition
Intersection	$X \cap Y$	$\{z \in X \mid z \in Y\}$
Différence	$X \setminus Y$	$\{z \in X \mid z \notin Y\}$

etc.

Notions fondamentales

Relations – fonctions

Ensemble des relations binaires entre X et Y (définition) :

$$X \leftrightarrow Y \hat{=} \mathcal{P}(X \times Y)$$

Relations entre éléments (notation) :

$$\text{si } R \in X \leftrightarrow Y, x \underline{R} y \hat{=} (x, y) \in R$$

Domaine et codomaine de $R \in X \leftrightarrow Y$ (définitions) :

$$\begin{aligned} \text{dom}(R) &\hat{=} \{x \in X \mid \exists y \in Y. x \underline{R} y\} \\ \text{ran}(R) &\hat{=} \{y \in Y \mid \exists x \in X. x \underline{R} y\} \end{aligned}$$

Ensemble des fonctions partielles de X vers Y (définition) :

$$X \not\rightarrow Y \hat{=} \{f \in X \leftrightarrow Y \mid \forall x \in \text{dom}(f). \exists! y \in Y. x \underline{f} y\}$$

Ensemble des fonctions totales de X vers Y (définition) :

$$X \rightarrow Y \hat{=} \{f \in X \not\rightarrow Y \mid \text{dom } f = X\}$$

Application d'une fonction $f \in X \not\rightarrow Y$ (axiome) :

$$f(x) = y \Leftrightarrow (x, y) \in f$$

Opération de «mise à jour» :

$$R \oplus S \hat{=} \{ (x, y) \in R \cup S \mid \begin{array}{l} (x \in \text{dom}(R) \setminus \text{dom}(S) \Rightarrow (x, y) \in R) \wedge \\ (x \in \text{dom}(S) \Rightarrow (x, y) \in S) \end{array} \}$$

Exemple : en notant $x \mapsto y$ le couple (x, y)

$$\begin{cases} f \oplus \{x \mapsto y\}(x) &= y \\ f \oplus \{x \mapsto y\}(z) &= f(z) \quad \text{si } z \neq x \end{cases}$$

Notions fondamentales (suite)

Arithmétique

L'ensemble des entiers naturels (définition) :

$$IN \hat{=} \text{ un truc un peu compliqué ...}$$

Constantes, opérations, relations :

$$0 \ 1 \ \dots \ + \ \times \ \leq \ \dots$$

Entiers non nuls (définition) :

$$IN_1 \hat{=} \{n \in IN \mid n \neq 0\}$$

Intervales d'entiers (définition) :

$$n..m \hat{=} \{k \in IN \mid n \leq k \wedge k \leq m\}$$

etc.

Structure linéaire générique

Les suites

Ensemble des suites d'éléments de X (définition) :

$$\boxed{\text{seq } X \hat{=} \{s \in IN_1 \rightarrow X \mid \exists n \in IN. \text{dom}(s) = 1..n\}}$$

Accès aux éléments de $s \in \text{seq } X$:

$$\boxed{s(i)}$$

avec $i \in \text{dom}(s)$

Suite vide :

$$\boxed{\langle \rangle \hat{=} \emptyset}$$

remarque : $\emptyset \in 1..0 \rightarrow X \in \text{seq } X$

Longueur d'une suite :

$$\boxed{\#s}$$

cardinal de l'ensemble s

Suites non vides (définition) :

$$\boxed{\text{seq}_1 X \hat{=} \{s \in \text{seq } X \mid s \neq \langle \rangle\}}$$

Fonctions sur les suites

Définitions axiomatiques

Format des définitions : $\frac{nom : type}{formule}$

Concaténation :

$$\frac{\begin{array}{c} \cap : \text{seq } X \times \text{seq } X \rightarrow \text{seq } X \\ \forall s_1, s_2 \in \text{seq } X. \forall i \in IN_1. \\ (i \leq \#s_1 \Rightarrow s_1 \cap s_2(i) = s_1(i)) \wedge \\ (i > \#s_1 \Rightarrow s_1 \cap s_2(i) = s_2(i - \#s_1)) \end{array}}{\quad}$$

Sous suite :

$$\frac{\begin{array}{c} sub : \text{seq } X \times IN \times IN \rightarrow \text{seq } X \\ \forall s \in \text{seq } X. \forall i, j \in IN. \\ \text{dom}(sub(s, i, j)) = 1..j - i + 1 \\ \forall k \in \text{dom}(sub(s, i, j)). sub(s, i, j)(k) = s(k + i - 1) \end{array}}{\quad}$$

Comme des listes :

$$Constructeurs \quad \frac{\begin{array}{c} nil : \text{seq } X \\ \hline nil = \emptyset \end{array}}{\quad} \quad \frac{\begin{array}{c} cons : X \times \text{seq } X \rightarrow \text{seq}_1 X \\ \forall s \in \text{seq } X. \\ cons(x, s) = \{1 \mapsto x\} \cap s \end{array}}{\quad}$$

$$Accesseurs \quad \frac{\begin{array}{c} car : \text{seq}_1 X \rightarrow X \\ \forall s \in \text{seq}_1 X. \\ car(s) = s(1) \end{array}}{\quad} \quad \frac{\begin{array}{c} cdr : \text{seq}_1 X \rightarrow \text{seq } X \\ \forall s \in \text{seq}_1 X. \\ cdr(s) = sub(s, 2, \#s) \end{array}}{\quad}$$

Comme des files d'attentes :

$$Constructeurs \quad \frac{\begin{array}{c} new : \text{seq } X \\ \hline new = \emptyset \end{array}}{\quad} \quad \frac{\begin{array}{c} add : \text{seq } X \times X \rightarrow \text{seq}_1 X \\ \forall s \in \text{seq } X. \\ add(s, x) = s \cap \{1 \mapsto x\} \end{array}}{\quad}$$

$$Accesseurs \quad \frac{\begin{array}{c} last : \text{seq}_1 X \rightarrow X \\ \forall s \in \text{seq}_1 X. \\ last(s) = s(\#s) \end{array}}{\quad} \quad \frac{\begin{array}{c} front : \text{seq}_1 X \rightarrow \text{seq } X \\ \forall s \in \text{seq}_1 X. \\ front(s) = sub(s, 1, \#s - 1) \end{array}}{\quad}$$

Retournement d'une suite

Schéma d'opérations

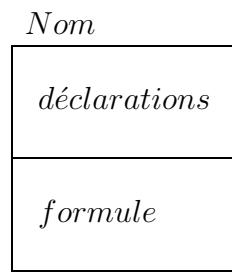
Relation (prédicat) *avant-après* :

$$Rev \in \text{seq } X \leftrightarrow \text{seq } X$$

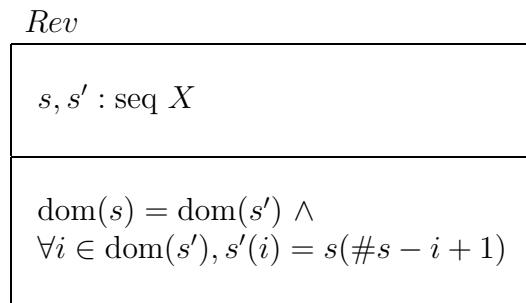
Intuitivement, procédure *vs* fonction :

$$s \underline{Rev} s' \hat{=} \text{«} s' \text{ est l'état de } s \text{ après l'opération »}$$

Format des schémas d'opération :



L'opération de retournement de suite :



Schémas d'opérations Quelque manipulations

Apostrophe :

$$\begin{array}{ccc}
 \boxed{\begin{array}{c} S \\ \hline x : T \\ \hline \varphi \end{array}} & \rightsquigarrow & \boxed{\begin{array}{c} S' \\ \hline x' : T \\ \hline \varphi[x'/x] \end{array}}
 \end{array}$$

Conjonction :

$$\begin{array}{ccc}
 \boxed{\begin{array}{c} S_1 \\ \hline x_1 : T_1 \\ \hline \varphi_1 \end{array}} & \boxed{\begin{array}{c} S_2 \\ \hline x_2 : T_2 \\ \hline \varphi_2 \end{array}} & \rightsquigarrow \boxed{\begin{array}{c} S_1 \wedge S_2 \\ \hline x_1 : T_1 \\ x_2 : T_2 \\ \hline \varphi_1 \wedge \varphi_2 \end{array}}
 \end{array}$$

Inclusion :

$$\begin{array}{ccc}
 \boxed{\begin{array}{c} S_2 \\ \hline S_1 \\ x_2 : T_2 \\ \hline \varphi_2 \end{array}} & \stackrel{\cong}{=} & \boxed{\begin{array}{c} S_2 \\ \hline x_1 : T_1 \\ x_2 : T_2 \\ \hline \varphi_1 \wedge \varphi_2 \end{array}}
 \end{array}$$

Identificateurs :

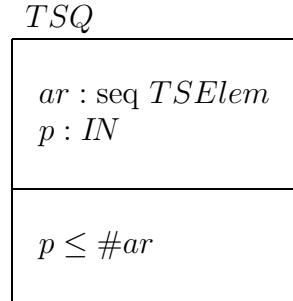
$$\begin{array}{ccc}
 \boxed{\begin{array}{c} S \\ \hline x_1 : T_1 \\ \cdots \\ x_n : T_n \\ \hline \varphi \end{array}} & \rightsquigarrow & \theta S \hat{=} (x_1, \dots, x_n)
 \end{array}$$

IBM CICS API Temporary Storage Queue

Ensembles de base :

$$\begin{array}{lcl} \textit{BYTE} & \hat{=} & 0..255 \\ \textit{TSElem} & \hat{=} & \text{seq } \textit{BYTE} \end{array}$$

Une suite et un pointeur (sur le dernier objet modifié) :



Initialisation :

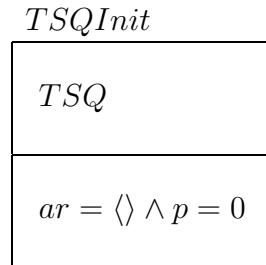
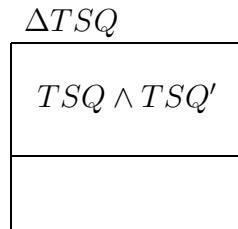


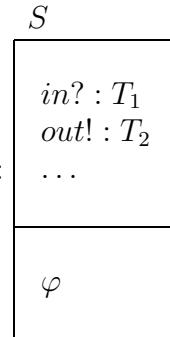
Schéma «avant-après» :



Temporary Storage Queue

Opérations fournies

Entrées-sortie, notations conventionnelles:



Ajout et retrait:

Append0

ΔTSQ
 $from? : TSElem$
 $item! : IN$

$ar' = ar \setminus \langle from? \rangle \wedge$
 $item! = \#ar' \wedge$
 $p' = p$

Remove0

ΔTSQ
 $item! : TSElem$

$p < \#ar \wedge$
 $p' = p + 1 \wedge$
 $into! = ar(p') \wedge$
 $ar' = ar$

Lecture et écriture:

Write0

ΔTSQ
 $item? : IN$
 $from? : TSElem$

$item? \in 1..\#ar \wedge$
 $ar' = ar \oplus \{item? \mapsto from?\} \wedge$
 $p' = p$

Read0

ΔTSQ
 $item? : IN$
 $into! : TSElem$

$item? \in 1..\#ar \wedge$
 $into! = ar(item?) \wedge$
 $p' = item? \wedge$
 $ar' = ar$

Temporary Storage Queue

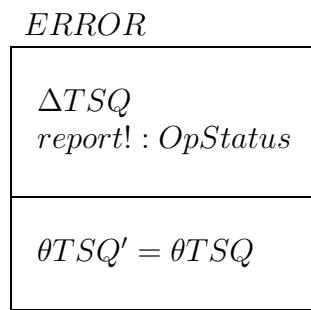
Gestion d'erreurs

Statuts d'exécution :

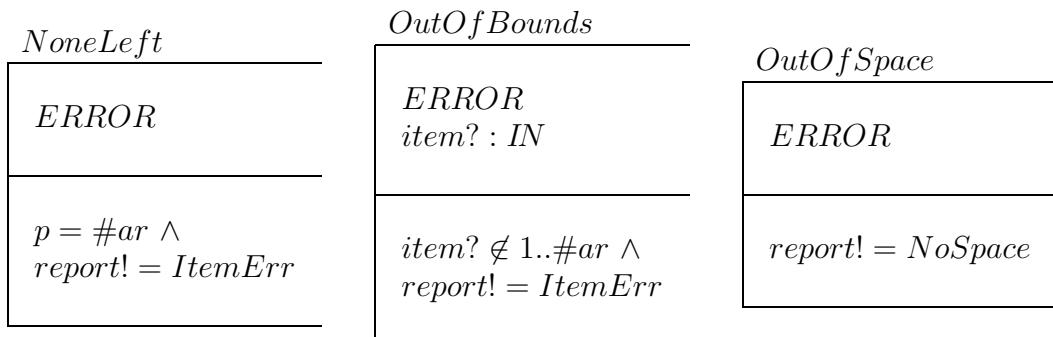
$$OpStatus ::= Success \mid ItemErr \mid NoSpace$$

(macro syntaxique pour $OpStatus = \{Success, ItemErr, NoSpace\}$)

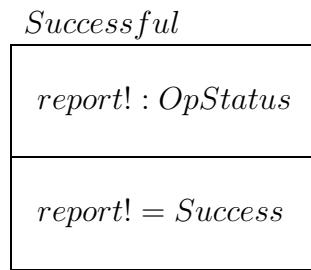
Schéma générique d'erreur :



Rapports d'erreurs :



Tout va bien :



Temporary Storage Queue Intégration

$$\begin{aligned} \text{Append} &\equiv (\text{Append0} \wedge \text{Successful}) \vee \text{OutOfSpace} \\ \text{Remove} &\equiv (\text{Remove0} \wedge \text{Successful}) \vee \text{NoneLeft} \\ \text{Write} &\equiv (\text{Write0} \wedge \text{Successful}) \vee \text{OutOfBounds} \vee \text{OutOfSpace} \\ \text{Read} &\equiv (\text{Read0} \wedge \text{Successful}) \vee \text{OutOfBounds} \end{aligned}$$