

### 1 Les listes

La *liste linéaire* est la forme la plus simple d'organisation de données que l'on puisse rencontrer. Celles-ci sont stockées les unes à la suite des autres dans des places et permettent divers traitements séquentiels. L'ordre des éléments dans une liste ne dépend pas des éléments eux-mêmes, mais de la **place** de ceux-ci dans la liste.

#### TYPES

liste, place

#### UTILISE

élément

Les opérations internes de **place** :

#### OPÉRATIONS

*contenu* : place  $\rightarrow$  élément

pour obtenir l'élément d'une place

*succ* : place  $\rightarrow$  place

pour passer à la place suivante

Il y a plusieurs façons de décrire une liste, soit **itérativement**, soit **récurivement** (cela reste la même donnée, seule la manière de s'en servir change).

### 2 Liste réursive

Une liste est

- soit vide (*listevide*),
- soit constituée d'un élément (la tête) suivi d'une liste (*cons*(e,  $\lambda$ )).

#### OPÉRATIONS

*listevide* :  $\rightarrow$  liste

crée une liste sans éléments

*cons* : liste  $\times$  élément  $\rightarrow$  liste

ajoute un élément en tête de liste

*fin* : liste  $\rightarrow$  liste

la liste privée de l'élément de tête

*tête* : liste  $\rightarrow$  place

la première place (celle de tête)

*premier* : liste  $\rightarrow$  élément

le premier élément d'une liste

#### PRÉCONDITIONS

*fin*( $\lambda$ ) **est-défini-ssi**  $\lambda \neq \text{listevide}$

*tête*( $\lambda$ ) **est-défini-ssi**  $\lambda \neq \text{listevide}$

*premier*( $\lambda$ ) **est-défini-ssi**  $\lambda \neq \text{listevide}$

#### AXIOMES

*premier*( $\lambda$ ) = *contenu*(*tête*( $\lambda$ ))

*fin*(*cons*(e, $\lambda$ )) =  $\lambda$

*premier*(*cons*(e, $\lambda$ )) = e

*succ*(*tête*( $\lambda$ )) = *tête*(*fin*( $\lambda$ ))

#### AVEC

liste  $\lambda$

élément e

Seul l'élément de tête est immédiatement accessible.

La liste  $\lambda = \{e_1, e_2, e_3\}$  s'écrit *cons*( $e_1$ , *cons*( $e_2$ , *cons*( $e_3$ , *listevide*))).

### 3 Liste itérative

Tous les éléments sont accessibles, par leur rang.

Le type **entier** est utilisé pour représenter les rangs.

#### UTILISE

entier

#### OPÉRATIONS

*listevide* :  $\rightarrow$  liste

*accès* : liste  $\times$  entier  $\rightarrow$  place

*accès*( $\lambda$ ,  $i$ ) : la place du  $i^{\text{ème}}$  élément de  $\lambda$

*ième* : liste  $\times$  entier  $\rightarrow$  élément

*longueur* : liste  $\rightarrow$  entier

*supprimer* : liste  $\times$  entier  $\rightarrow$  liste

*supprimer*( $\lambda$ ,  $k$ ) : supprime le  $k^{\text{ème}}$  élément de  $\lambda$

*insérer* : liste  $\times$  entier  $\times$  élément  $\rightarrow$  liste

*insérer*( $\lambda$ ,  $k$ ,  $e$ ) : insère  $e$  au rang  $k$  dans  $\lambda$

#### PRÉCONDITIONS

*accès*( $\lambda$ ,  $k$ ) **est-défini-ssi**  $\lambda \neq \text{listevide} \ \& \ 1 \leq k \leq \text{longueur}(\lambda)$

*supprimer*( $\lambda$ ,  $k$ ) **est-défini-ssi**  $\lambda \neq \text{listevide} \ \& \ 1 \leq k \leq \text{longueur}(\lambda)$

*insérer*( $\lambda$ ,  $k$ ,  $e$ ) **est-défini-ssi**  $1 \leq k \leq \text{longueur}(\lambda) + 1$

#### AXIOMES

$\text{longueur}(\text{listevide}) = 0$

$\text{longueur}(\text{supprimer}(\lambda, k)) = \text{longueur}(\lambda) - 1$

$\text{longueur}(\text{insérer}(\lambda, k, e)) = \text{longueur}(\lambda) + 1$

$1 \leq i < k \Rightarrow \text{ième}(\text{supprimer}(\lambda, k), i) = \text{ième}(\lambda, i)$

$k \leq i \leq \text{longueur}(\lambda) - 1 \Rightarrow \text{ième}(\text{supprimer}(\lambda, k), i) = \text{ième}(\lambda, i + 1)$

$1 \leq i < k \Rightarrow \text{ième}(\text{insérer}(\lambda, k, e), i) = \text{ième}(\lambda, i)$

$k = i \Rightarrow \text{ième}(\text{insérer}(\lambda, k, e), i) = e$

$k < i \leq \text{longueur}(\lambda) + 1 \Rightarrow \text{ième}(\text{insérer}(\lambda, k, e), i) = \text{ième}(\lambda, i - 1)$

$\text{contenu}(\text{accès}(\lambda, k)) = \text{ième}(\lambda, k)$

$\text{succ}(\text{accès}(\lambda, k)) = \text{accès}(\lambda, k + 1)$

#### AVEC

liste  $\lambda$

entier  $i, k$

élément  $e$

La liste  $\lambda = \{e_1, e_2, e_3\}$  sera définie par :

$\text{longueur}(\lambda) = 3$

$\text{ième}(\lambda, 1) = e_1$

$\text{ième}(\lambda, 2) = e_2$

$\text{ième}(\lambda, 3) = e_3$