TD Programmation fonctionnelle CS-110

Chapitre 2 : Définitions récursives

Les séquences sont définies par analogie avec le type texte vu dans le chapitre 1. Nous présentons tout d'abord les constructeurs, testeur et sélecteurs des séquences. Puis nous étudions les modèles de découpage d'une séquence selon les extrémités, et selon une propriété.

1 Constructeurs, testeur, sélecteurs.

- Élément : un type
On travaille sur des séquences d'Éléments.

- [] désigne la séquence vide. Le testeur associé est **EstVide ?**
EstVide ? : une séquence d'Éléments → un booléen

- $S \bullet e$ désigne la séquence construite en ajoutant l'élément e à droite de la séquence S.
Les sélecteurs associés sont **début** et **dernier**.
$\bullet$: une séquence d'Éléments, un Élément → une séquence non vide d'Éléments
début : une séquence non vide d'Éléments → une séquence d'Éléments {début($S \bullet e$)= S}
dernier : une séquence non vide d'Éléments → un Élément {dernier($S \bullet e$)= e}

- $e_\circ S$ désigne la séquence construite en ajoutant l'élément e à gauche de la séquence S.
Les sélecteurs associés sont **premier** et **fin**.
$\circ$: un Élément, une séquence d'Éléments → une séquence non vide d'Éléments
premier : une séquence non vide d'Éléments → un Élément {premier($e_\circ S$)= e}
fin : une séquence non vide d'Éléments → une séquence d'Éléments {fin($e_\circ S$)= S}

- Une séquence peut être décrite en extension, en énumérant ses éléments séparés par des virgules, le tout étant entouré de crochets : $[e_1, e_2, ..., e_n]$. En particulier, $[e]$ désigne un **singleton** : c'est la séquence formée du seul élément e : $[e] = [] \bullet e = e_\circ[]$

- Le prédicat **EstSingleton ?** détermine si une séquence est un singleton.
EstSingleton ? : une séquence d'Éléments → un booléen
EstSingleton ?(X) ⇔ non EstVide ?(X) et puis EstVide ?(début(X))
⇔ non EstVide ?(X) et puis EstVide ?(fin(X))

- On peut concaténer deux séquences avec & :
& : deux séquences d'Éléments → une séquence d'Éléments

2 Modèles de découpage d'une séquence ...

2.1 ...selon les extrémités

Découpage isolant...	Forme des équations de récurrence	Forme des réalisations récursives de f(X)	Variante : séquence non vide
Le premier élément	f([]) = f($e_\circ S$) =	Si EstVide ?(X) alors ... sinon soit e = premier(X), S = fin(X) dans ...	f([e]) = f($e_\circ S$) = { *S non vide* }
Le dernier élément	f([]) = f($S \bullet e$) =	Si EstVide ?(X) alors ... sinon soit e = dernier(X), S = début(X) dans ...	f([e]) = f($S \bullet e$) = { *S non vide* }
Le premier et le dernier élément	f([]) = f([e]) = f($e1_\circ S \bullet e2$)	Si EstVide ?(X) alors ... sinon si EstVide ?(fin(X)) alors ... sinon soit e1 = premier(X), e2 = dernier(X), S = début(fin(X)) dans ...	f([e]) = f([$e1, e2$]) = f($e1_\circ S \bullet e2$) = { *S non vide* }
Les deux premiers éléments	f([]) = f([e]) = f($e1_\circ e2_\circ S$)	Si EstVide ?(X) alors ... sinon si EstVide ?(fin(X)) alors ... sinon soit e1 = premier(X), e2 = premier(fin(X)), S = fin(fin(X)) dans ...	f([e]) = f([$e1, e2$]) = f($e1_\circ e2_\circ S$) = { *S non vide* }
Les deux derniers éléments	f([]) = f([e]) = f($S \bullet e1 \bullet e2$)	Si EstVide ?(X) alors ... sinon si EstVide ?(début(X)) alors ... sinon soit e2 = dernier(X), e1 = dernier(début(X)), S = début(début(X)) dans ...	f([e]) = f([$e1, e2$]) = f($S \bullet e1 \bullet e2$) = { *S non vide* }

2.2 ...selon une propriété

P est une propriété sur les Éléments :

P : un Élément → un booléen

Découper une séquence donnée S selon la propriété P consiste à identifier le premier élément e de la séquence (de gauche à droite) vérifiant P (tel que P(e) soit vrai). Cet élément est appelé l'*élément de séparation* selon P. S est alors découpée en deux sous-séquences D et F (S = D&F), respectivement appelées le *début* de S et la *fin* de S selon P. Par convention, l'élément de séparation, s'il existe, est le premier élément de la séquence de fin F. Si aucun élément de la séquence donnée ne vérifie P, F est vide. Si le premier élément de S vérifie P, D est vide. Si S est vide, D et F le sont aussi.

On donne ici les équations de récurrence pour calculer le début et la fin selon P :

LeDébutSelonP : une séquence d'Éléments → une séquence d'Éléments
{séquence de début de la séquence donnée jusqu'au premier élément (exclu) vérifiant P }
(1) LeDébutSelonP([]) = []
(2) LeDébutSelonP($e_\circ S$) = si P(e) alors [] sinon $e_\circ$LeDébutSelonP(S)

LaFinSelonP : une séquence d'Éléments → une séquence d'Éléments
{séquence de fin de la séquence donnée à partir du premier élément (inclus) vérifiant P }
(1) LaFinSelonP([]) = []
(2) LaFinSelonP($e_\circ S$) = si P(e) alors $e_\circ S$ sinon LaFinSelonP(S)

DebEtFinSelonP : une séquence d'Éléments → <une séquence d'Éléments, une séquence d'Éléments>
{DebEtFinSelonP(S) = < LeDébutSelonP(S), LaFinSelonP(S) > }
(1) DebEtFinSelonP([]) = <[], []>
(2) DebEtFinSelonP($e_\circ S$) = si P(e) alors <[], $e_\circ S$>
sinon soit <D, F> = DebEtFinSelonP(S)
dans <$e_\circ D$, F>