

UTBM

Automne 2002

IA41

INTELLIGENCE ARTIFICIELLE

Représentation des connaissances et résolution de problèmes

Responsable : Jean César

Intervenants : Yassine Ruichek, Olivier Grunder

Le 13 novembre 2003 de 8h à 9h, salles A200 et B429

EXAMEN MEDIAN

SUJET : SUBSTITUTIONS (Corrigé)

1. Etudiez la fonction **subst** qui remplace dans une liste L la première occurrence (au premier niveau seulement) de Exp1 par Exp2.

Profil : $List \times Expr^2 \rightarrow List$

Spécification : - La liste vide donne la liste vide,

- S'il n'y a pas d'occurrence de Exp1, la liste est rendue inchangée,

- Cas général : la première occurrence de Exp1, au premier niveau, est changée en Exp2.

Jeu d'essais (fourni) : $() \rightarrow ()$

$(a \ b \ c \ (c \ (a \ b)) \ (a \ b) \ (a \ b)), (a \ b), * \rightarrow (a \ b \ c \ (c \ (a \ b)) \ * \ (a \ b))$

Définition formelle : $\forall (L, Exp1, Exp2) \in List \times Expr^2$

$L = () \Rightarrow subst \ L, Exp1, Exp2 = ()$

$L \neq ()$ et tête $L = Exp1 \Rightarrow subst \ L, Exp1, Exp2 = ajout \ Exp2, reste \ L$

et tête $L \neq Exp1 \Rightarrow subst \ L, Exp1, Exp2 = ajout \ têteL, subst \ reste \ L, Exp1, Exp2$

2. Etudiez la fonction **substall** qui remplace dans une liste L toutes les occurrences (au premier niveau seulement) de Exp1 par Exp2.

Donnez AU CHOIX une version récursive OU une version applicative.

Profil : idem.

Spécification : idem sauf

-Cas général : toutes les occurrences de Exp1, au premier niveau, sont changées en Exp2.

Jeu d'essais (fourni) : résultat de l'exemple non vide : $(a \ b \ c \ (c \ (a \ b)) \ * \ *)$

Définition formelle récursive:

$L = () \Rightarrow substall \ L, Exp1, Exp2 = ()$

$L \neq ()$ et tête $L = Exp1 \Rightarrow substall \ L, Exp1, Exp2 = ajout \ Exp2, substall \ reste \ L, Exp1, Exp2$

et tête $L \neq Exp1 \Rightarrow substall \ L, Exp1, Exp2 = ajout \ têteL, substall \ reste \ L, Exp1, Exp2$

Schéma applicatif : $(a \ b \ c \ (c \ (a \ b)) \ (a \ b) \ (a \ b))$

$\begin{array}{ccccccc} | & | & | & & | & & | \\ (a \ b \ c \ (c \ (a \ b)) \ * \ *) \end{array}$

Définition formelle applicative:

$substall \ L, Exp1, Exp2 = terme\text{-}après\text{-}terme \quad \lambda(x) \quad x = Exp1 \rightarrow Exp2$

$x \neq Exp1 \rightarrow x,$

L

3. Etudiez la fonction **substdeep** qui remplace dans une liste L toutes les occurrences (à tous les niveaux) de Exp1 par Exp2.

Donnez AU CHOIX une version doublement récursive OU une version applicative en largeur et récursive en profondeur.

Profil : idem.

Spécification : idem sauf

Cas général : toutes les occurrences de Exp1, à tous les niveaux, sont changées en Exp2.

Jeu d'essais (fourni) : résultat de l'exemple non vide : (a b c (c *) **)

Définition formelle doublement récursive:

$L = () \Rightarrow \text{substdeep } L, \text{Exp1}, \text{Exp2} = ()$

$L \neq ()$ et tête $L = \text{Exp1} \Rightarrow \text{substdeep } \dots = \text{ajout } \text{Exp2}, \text{substdeep } \text{reste } L, \text{Exp1}, \text{Exp2}$

et tête $L \neq \text{Exp1}$ et tête $L \in \text{Atom} \Rightarrow \text{substdeep } L, \text{Exp1}, \text{Exp2} =$

ajout têteL, substdeep reste L, Exp1, Exp2

et tête $L \notin \text{Atom} \Rightarrow \text{substdeep } L, \text{Exp1}, \text{Exp2} =$

ajout substdeep têteL Exp1, Exp2, substdeep reste L, Exp1, Exp2

Schéma applicatif :

(a b c (c (a b)) (a b) (a b))

||| | | |
(a b c (c *) * *)

Définition formelle applicative: $\text{substdeep } L, \text{Exp1}, \text{Exp2} =$

terme-après-terme lambda (x) $x = \text{Exp1} \rightarrow \text{Exp2}$

$x \neq \text{Exp1}$ et $x \in \text{Atom} \rightarrow x$

et $x \notin \text{Atom} \rightarrow \text{substdeep } x, \text{Exp1}, \text{Exp2},$

L

4. Etudiez la fonction **substlist** qui remplace dans une liste L toutes les occurrences (à tous les niveaux) des clés par leurs valeurs selon une a-liste AL donnée. On précise que nil n'est pas utilisé comme valeur dans AL.

4-a) Tirez de votre réponse au n°3 une version parcourant une fois la liste L,

4-b) Donnez une autre version parcourant une fois la liste AL.

Profil : List X AList \rightarrow List

Spécification : - La liste vide donne la liste vide,

- S'il n'y a pas d'occurrence de clé, la liste est rendue inchangée,

- Cas général : toutes les occurrences de toutes les clés, à tous les niveaux, sont changées en leurs valeurs associées.

Jeu d'essais (fourni) : $() \rightarrow ()$

(a b c (c (a b)) (a b) (a b)), (((a b), *) (c -)) \rightarrow (a b - (- *) **)

Schéma applicatif :

(a b c (c (a b)) (a b) (a b))

||| | | |
(a b - (- *) * *)

Définition formelle en parcourant une fois L : $\forall(L,AL) \in \text{List X Alist}$

La définition applicative permet facilement de n'appeler qu'une fois **accès** :

substlist L,AL = terme-après-terme lambda (x) = accès x,AL ou
 $x \in \text{atom} \rightarrow x$
 $x \notin \text{atom} \rightarrow \text{substlist } x,AL$

Définition formelle en parcourant une fois AL :

AL = () => L

AL ≠ () => substlist substdeep L,tête tête AL,tête reste tête AL, reste AL

ATTENTION : les deux définitions donnent des résultats différents si les substitutions peuvent se composer, comme par exemple avec ((a d) (d e))!!!