

Examen Médian

Jeudi 16 Novembre 2006, de 8h à 10h, en salles A200 et B429, site de Belfort.

Coefficient : 30%

Document autorisé : une feuille simple recto-verso **manuscrite**.

Remarques :

- On suppose que les fonctions définies lors des TD ne pourront pas être utilisées.
- Pour une question donnée, les fonctions des questions précédentes pourront être utilisées.
- Toutes les fonctions à définir ne devront générer **aucune erreur**. On rappelle qu'en particulier, si $a \in \text{Atom}$, $tete\ a$ et $reste\ a$ génèrent une erreur.

1) Prédicat **objet** (2 points)

On définit le nouveau type de données “*objet*”, qui est décrit par un couple constitué de deux propriétés numériques : son poids et son utilité. Par exemple (5 10) et (2.5 5) sont des objets, alors que 2, a, (5 b) et (5 8 9) ne le sont pas.

1. Énoncez en français les 6 propriétés élémentaires que doit vérifier une expression symbolique E pour appartenir au type “objet”.
2. Déduisez-en la définition formelle du prédicat **objet o**.
3. Donnez le résultat de l'application de votre prédicat sur le jeu d'essais suivant :
 - objet ()
 - objet (1 2)
 - objet (1 u)
 - objet ((5 2))
 - objet (1 2 5)

2) Prédicat **sac** (3 points)

On définit le nouveau type de données “*sac*”. Un sac peut être vide ou contenir uniquement des objets ou d'autres sacs.

1. Donnez le profil et la définition formelle récursive du prédicat **sac s**, qui retourne *VRAI* si s est un sac et *FAUX* sinon.
2. Simuler l'application de votre fonction sur le jeu d'essais suivant :
 - sac ()
 - sac ((3 9) (5 1))
 - sac ((3 9) 1 a (5 2))
 - sac ((1 2) ((5 6) (10 2)))

3) Fonction **nettoieSac** (2 points)

Cette fonction retire tous les éléments d'un sac qui ne sont pas des objets ou des sacs.

1. Donnez le profil et la définition formelle récursive de la fonction **nettoieSac**.
2. Simuler l'application de votre fonction sur le jeu d'essais suivant :
 - nettoieSac ((3 9) (5 1))
 - nettoieSac ((3 9) 1 a (5 2))
 - nettoieSac ((1 2) ((5 6) (10 2) a ((1 5) b)))

4) Fonction **compteObjets** (3 points)

Cette fonction compte le nombre total d'objets contenus dans un sac *s* valide (c'est-à-dire nettoyé). **compteObjets** comptabilise également les objets contenus éventuellement dans d'autres sacs placés dans *s*.

1. Donnez le profil et la définition formelle récursive de la fonction **compteObjets**.
2. Simuler l'application de votre fonction sur un jeu d'essais exhaustif.

5) Prédicat **appObjet** (3 points)

Ce prédicat retourne *VERAI* si un objet est présent dans un sac, à n'importe quel niveau. Il retourne *FAUX* sinon.

1. Donnez le profil et la définition formelle du prédicat **appObjet**.
2. Simuler l'application de votre prédicat sur un jeu d'essais exhaustif.

6) Fonction **appObjet2** (2 points)

Cette fonction vérifie si un objet est présent dans un sac, à n'importe quel niveau, et retourne le nombre de sacs qu'il a fallu vider pour trouver l'objet. Elle retourne *FAUX* si l'objet n'a pas été trouvé.

1. Donnez le profil et la définition formelle de la fonction **appObjet2**.
2. Simuler l'application de votre fonction sur un jeu d'essais exhaustif.

7) Fonction **rangeObjets** (2 points)

Cette fonction retire tous les objets placés éventuellement dans des sous-sac du sac *s* et les range les uns à côté des autres dans le sac *s*.

Donnez le profil et la définition formelle de cette fonction **rangeObjets s**.

8) Prédicat **sacFragile** (1 point)

On définit le nouveau type de données "*sacFragile*". Un sac fragile est caractérisé par son contenu ainsi que par un poids maximum de contenance *P*, qui représente le poids total d'objets qu'il peut contenir sans s'endommager. Par exemple, le sac fragile contenant les objets (1 2) et (3 6) et tel que $P = 5$ n'est pas endommagé. Mais si un objet de poids supérieur à 1 y est ajouté, il le devient.

Donnez le profil et la définition formelle du prédicat **sacFragile sf**.

9) Prédicat **testePoids** (2 points)

Ce prédicat retourne *VERAI* si la somme des poids des objets contenus dans un sac fragile est inférieure ou égale au poids total du sac. Il retourne *FAUX* sinon.

1. Donnez le profil et la définition formelle de la fonction **poidsTotal s**, qui retourne la somme des poids des objets contenus dans le sac *s* valide.
2. Donnez le profil et la définition formelle du prédicat **testePoids sf**.

Questions bonus

3) 3. Donnez le profil et la définition formelle de la fonction **nettoieSac2 s**, qui retire tous les objets non valides dans le sac *s*, et également dans les sacs éventuellement placés dans *s*.

10) Donnez le profil et la définition formelle de la fonction **retireObjetsSacFragile**, qui retire les objets les moins lourds d'un sac fragile *sf* afin que la somme des poids des objets qu'il contient reste inférieure à son poids maximum de contenance.