Prof. Burkhart Wolff
wolff@lri.fr

Hugo Mlodecki, Emmanuel Courtoux
hugo.mlodecki@u-psud.fr, e.courtoux@gmail.com

TD 4 - Reprise Spec et Test fonctionnel

Semaine du 5 octobre 2020

Exercice 1
L'opération `middle` prend en entrée trois entiers différents deux à deux et renvoie l'entier parmi les trois qui n'est ni le plus grand ni le plus petit.

1. Donnez une spécification formelle de cette opération.
2. Donnez par construction DNF un ensemble de tests pour l'opération `middle`.

Exercice 2 (Test fonctionnel (Revision formelle))
on reprend l'exercice TD3/2 :

Une société vend deux produits A et B au prix unitaire de 5 EUR pour A et de 10 EUR pour B. Une commande comprend une certaine quantité du produit A et une certaine quantité du produit B. Le coût d'une commande est la somme totale des prix unitaires des produits commandés, à laquelle on applique une réduction selon les règles suivantes :

— Si la somme totale est supérieure ou égale à 200 EUR, on applique une réduction de 5%, si elle est supérieure ou égale à 1000 EUR, la réduction est de 20%. Ces deux réductions ne sont pas cumulables et portent sur la somme totale.

— La société souhaitant encourager la vente de A, on applique, sur le prix obtenu grâce à la règle précédente, une réduction supplémentaire de 10% si la commande comprend au moins 45 produits A.

1. Donnez une specification formelle de la fonction `calc_prix(A,B:int):Rat`.
2. Donnez un ensemble de tests pour le calcul du coût total d'une commande. Utilisez la construction de la DNF pour deriver les cas de test formellement.
3. Complétez votre jeu de tests par une analyse aux limites.

Exercice 3 (Test boîte noire)
On veut tester une implementation un programme `magic_sort` on listes et specifie d'abord la notion de tri comme suivant :

$$\begin{aligned}
&\textsf{definition}\quad insert :: "int \Rightarrow int\ list \Rightarrow int\ list"\\
&\qquad \textsf{where} "insert\ a\ [] = [a]"\\
&\qquad\quad | "insert\ a\ (b\#S) = (\textsf{if}\ a \leq b\ \textsf{then}\ a\#(b\#S)\ \textsf{else}\ b\#(insert\ a\ S))\ "\\
&\textsf{definition}\quad sort :: "int\ list \Rightarrow int\ list"\\
&\qquad \textsf{where} "sort\ [] = []"\\
&\qquad\quad | "sort\ (a\#S) = insert\ a\ (sort\ S)"
\end{aligned}$$

which allows to define the contract of `magic_sort(S)` as follows :

$$\begin{aligned}
\textsf{definition}\quad & pre_{magic_sort}(S) \equiv length(S) = k\\
\textsf{definition}\quad & post_{magic_sort}(S, res) \equiv (sort\ S = res)
\end{aligned}$$

pour un k prédefini. On note que # et le constructeur d'une liste, et que la notation $[a, b, c]$ est donc une abbreviation de $a\#(b\#(c\#[]))$.

Questions

1. Dépliez, puis simplifiez jusqu'à la DNF l'application de *magic_sort* sur une liste contenant deux entiers : $pre_{magic\_sort}([a, b]) \wedge post_{magic\_sort}([a, b], res)$.
2. Construisez un jeu de tests concret couvrant la DNF. Ajouter les cas limites.
3. Combien y a-t-il de cas à couvrir dans le cas $k = 3$? (Cas concret et cas limites)