Points fixes et sémantique

Le langage IML_{rep}

Le langage IML_{rep} est une variante du langage IML. Les classes syntaxiques *Aexpr* et *Bexpr* sont les mêmes du langage IML. La classe *Comm* des commandes est définie par la grammaire

$$c = \; n \mid X{:=}a \mid c_0 \; ; \; c_1 \mid \texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1 \mid \texttt{repeat}\ c\ \texttt{until}\ b$$

où $a \in Aexpr$, $b \in Bexpr$, et $c, c_0, c_1 \in Comm$.

Exercice 1. Utiliser votre intuitions pour donner
– une sémantique opérationnelle,
– et une sémantique denotationelle de ce langage.

Exercice 2. Dans le langage IML_{rep} peut-on définir une commande qui se comporte comme la commande `skip` ? Si oui, en donner une preuve formelle à l'aide de la sémantique.

Le plus petit point préfixe

Exercice 3. Soient $f : L \longrightarrow M$ et $g : M \longrightarrow L$ deux fonctions monotones, où L et M sont deux treillis complets. On connaît, donc, qu'un plus petit point fixe de $g \circ f$, noté $fix(g \circ f)$, et un plus petit point fixe de $f \circ g$, noté $fix(f \circ g)$, existent. Montrer que :

$$f(\ fix(g \circ f)\ ) = fix(f \circ g)\ .$$

On peut démontrer à l'aide des axiomes suivants :

$$h(fix(h)) \leq fix(h)$$
$$h(y) \leq y \quad \Rightarrow \; fix(h) \leq y\ .$$

Exercice 4. On considère maintenant un langage IML^+_{rep} : les classes syntaxiques *Aexpr* et *Bexpr* sont toujours les mêmes, la classe *Comm* des commandes inclue des boucles `while` ainsi que des boucles `repeat`. Sa grammaire est :

$$c := n \mid X := a \mid c_0 \; ; \; c_1 \mid \texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1 \mid \texttt{repeat}\ c\ \texttt{until}\ b \mid \texttt{while}\ b\ \texttt{do}\ c$$

où $a \in Aexpr$, $b \in Bexpr$, et $c, c_0, c_1 \in Comm$.
On suppose que vous avez proposé la « bonne » sémantique dénotationnelle à la boucle `repeat`. Démontrer alors l'équivalence des commandes

$$\texttt{repeat}\ c\ \texttt{until}\ b \qquad \texttt{c;}\ \texttt{while not}\ b\ \texttt{do}\ c\ .$$

Suggestions :

1. Si $P \subseteq \Sigma$, on définit la relation binaire
$$T_P = \{\ (\sigma, \sigma) \mid \sigma \in P\ \}\ .$$
Montrer alors que
$$R \circ T_P = \{\ (\sigma, \sigma') \mid \sigma \in P \text{ et } (\sigma, \sigma') \in R\ \}\ .$$
Déterminer aussi $T_P \circ R$.
2. Pour $b \in Bexpr$, soit $T_b = \{\,(\sigma, \sigma) \mid \beta\|b\|(\sigma) = true\,\}$. Exprimer l'opérateur $\Gamma(Z)$ (dont le point fixe interprète la commande `while` b `do` c) à l'aide des relations T_b, $\gamma = \mathcal{C}\|c\|$ et des opérations d'union et de composition.
Faire de même pour l'opérateur $\Delta(Z)$ dont le point fixe interprète la commande `repeat` c `until` b.
3. Se convaincre que la composition des relations est distributive par rapport à la union :
$$R \circ (S_1 \cup S_2) = (R \circ S_1) \cup (R \circ S_2)$$
$$(R_1 \cup R_2) \circ S = (R_1 \circ S) \cup (R_2 \circ S)$$
4. On pourra maintenant se servir de l'Exercice 3.