R. Hinze, S. Franck Summer Semester 2005

Software Engineering

http://proglang.informatik.uni-freiburg.de/teaching/swt/2005/

Exercise Sheet 7

Deadline: June 7th, 2005

Exercise 1 – Trees in Z: (2 points)
The following type-definition models a Binary Tree.

$$\mathit{BINTREE} ::= \mathit{Leaf} \mid \mathit{Node}\langle\!\langle \mathbb{Z} \times \mathit{BINTREE} \times \mathit{BINTREE} \rangle\!\rangle$$

Give a schema for Heap, e.g. a Binary tree where the content of every node is greater than or equal to the content of all his descendants.

Solution:

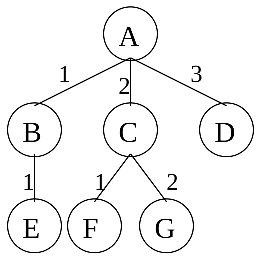
$$\begin{array}{|l}
\mathit{greaterequal} : \mathit{BINTREE} \rightarrow \mathbb{P}\,\mathbb{Z} \\
\hline
\forall\, i : \mathbb{Z}, t_l, t_r, : \mathit{BINTREE} \bullet \\
\quad \mathit{greaterequal}\ \mathit{Leaf} = \mathbb{Z} \\
\quad \mathit{greaterequal}\ \mathit{Node}(i, t_l, t_r) = \{n : \mathbb{Z} \mid n \geq i\}
\end{array}$$

$$\begin{array}{|l}
\text{— } \mathit{Heap} \text{ —} \\
t : \mathit{BINTREE} \\
\hline
t = \mathit{Leaf} \vee \\
(\exists\, i : \mathbb{Z}, h_l, h_r : \mathit{Heap} \bullet \\
\quad t = \mathit{Node}(i, h_l.t, h_r.t) \wedge \\
\quad i \in \mathit{greaterequal}\ h_l.t \wedge \\
\quad i \in \mathit{greaterequal}\ h_r.t) \\
\hline
\end{array}$$

Exercise 2 – Trees in Z (part II): (6 points)
Another way to define trees is analogous to the definition of sequences (which map numbers to values): Trees may be seen as functions from sequences to nodes, where the sequence describes the path from the root to the node.
Hence, trees can be defined as:

$$\begin{aligned}
\text{tree}\ X = \{t : \text{seq}\,\mathbb{N} \mathrel{+\!\!\!\!\twoheadrightarrow} X \mid\ & \langle\rangle \in \mathsf{dom}\ t \wedge \\
& \forall \pi \frown \sigma \in \mathsf{dom}\ t \Rightarrow \pi \in \mathsf{dom}\ t \wedge \\
& \forall \pi \frown \langle n \rangle \in \mathsf{dom}\ t \Rightarrow \forall\, i < n \bullet \pi \frown \langle i \rangle \in \mathsf{dom}\ t\}
\end{aligned}$$

So t maps a sequence of numbers to nodes (elements of X). The numbers represent the path from the root to the respective node by indicating which son of the current node has to be taken. A sequence $\langle 1, 2 \rangle$ describes for example the node which is the second son of the first son of root. With every number of the sequence you descend one level in the tree.
The empty sequence maps to the root of the tree.

If we have a tree ex 

then

$$\begin{array}{rcl} ex(\langle\rangle) & = & A \\ ex(\langle 1\rangle) & = & B \\ ex(\langle 2,1\rangle) & = & F \\ ex(\langle 3,1\rangle) & = & \text{undefined} \end{array}$$

Define the following functions:

1. content, which returns a set of all nodes in a tree.
   Example:
   $$content(ex) = \{A, B, C, D, E, F, G\}$$

2. subtree, which returns the subtree at a position specified by a sequence.
   Example:
   $$subtree(ex, \langle 1\rangle) = \{(\langle\rangle, B), (\langle 1\rangle, E)\}$$

3. appendtree, which appends a tree under a given position.
   Example:
   $$appendtree(ex, \{(\langle\rangle, H)\}, \langle 2\rangle) = ex \cup \{(\langle 2, 3\rangle, H)\}$$
   (a tree which only contained H was inserted under C).

4. prune, which removes a subtree from a specified position. The node reached by the sequence shall be removed as well.
   Example:
   $$prune(ex, \langle 2\rangle) = \{(\langle\rangle, A), (\langle 1\rangle, B), (\langle 2\rangle, D), (\langle 1, 1\rangle, E)\}$$
   prune cannot be called with the empty sequence, since any tree has to have a root (see definition).

Solution:

$[X]$

$content(\_) : \text{tree } X \rightarrow \mathbb{P}\, X$

$\forall\, t : \text{tree } X \bullet content(t) = \mathsf{ran}\, t$

$[X]$

$subtree(\_, \_) : \text{tree } X \times \text{seq}\, \mathbb{N} \rightarrow \text{tree } X$

$\forall\, t \in \text{tree } X, s \in \mathsf{dom}\, t :$
$\quad subtree(t, s) = (\lambda\, x : \{n : \text{seq}\, \mathbb{N} \mid s \frown n \in \mathsf{dom}\, t\} \bullet t(s \frown x))$

$$\begin{array}{l}
[X] \\
\hline
\mathit{freeSon}(\_,\ \_) : \text{tree } X \times \text{seq}\,\mathbb{N} \to \mathbb{N} \\
\hline
\forall\, t\text{: tree } X, s : \mathsf{dom}\ t \bullet \\
\quad \mathit{freeSon}(t, s) = \mu\, n : \mathbb{N} \bullet s \frown \langle n \rangle \notin \mathsf{dom}\ t\ \land \\
\qquad\qquad\qquad\qquad\qquad (n = 1 \lor s \frown \langle (n-1) \rangle \in \mathsf{dom}\ t)
\end{array}$$

$$\begin{array}{l}
[X] \\
\hline
\mathit{appendtree}(\_,\ \_,\ \_) : \text{tree } X \times \text{tree } X \times \text{seq}\,\mathbb{N} \to \text{tree } X \\
\hline
\forall\, t_1, t_2\text{: tree } X, s : \mathsf{dom}\ t \\
\qquad \mathit{appendtree}(t_1, t_2, s) = t_1 \cup \\
\qquad\qquad\qquad\qquad \{p : \text{seq}\,\mathbb{N} \mid p \in \mathsf{dom}\ t_2 \bullet s \frown \langle \mathit{freeSon}(t_1,\ s) \rangle \frown p \mapsto t_2(p)\}
\end{array}$$

$$\begin{array}{l}
[X] \\
\hline
\mathit{dropLast}(\_) : \text{seq}_1\ X \to \text{seq}\ X \\
\hline
\forall\, s\text{: seq}_1\ X \bullet \\
\quad \mathit{dropLast}(s) = (\lambda\, x : 1..(\#s - 1) \bullet s(x))
\end{array}$$

$$\begin{array}{l}
[X] \\
\hline
\mathit{dependentNodes}(\_,\ \_) : \text{tree } X \times \text{seq}_1\,\mathbb{N} \to \mathbb{P}\,\text{seq}\,\mathbb{N} \\
\hline
\forall\, t\text{: tree } X, s : \mathsf{dom}\ t \bullet \\
\quad \mathit{dependentNodes}(t, s) = \{n : \text{seq}\,\mathbb{N}, i : \mathbb{N} \mid i \geq s(\#s) \land \mathit{dropLast}(s) \frown \langle i \rangle \frown n \in \mathsf{dom}\ t \\
\qquad\qquad\qquad\qquad\qquad\qquad \bullet (\mathit{dropLast}(s) \frown \langle i \rangle \frown n\}
\end{array}$$

$$\begin{array}{l}
[X] \\
\hline
\mathit{prune}(\_,\ \_) : \text{tree } X, \text{seq}_1\,\mathbb{N} \to \text{tree } X \\
\hline
\forall\, t\text{: tree } X, s : \mathsf{dom}\ t \bullet \\
\quad \mathit{prune}(t, s) = \mathit{dependentNodes}(t, s) \lhd t \cup \\
\quad \{n : \text{seq}\,\mathbb{N}, i : \mathbb{N} \mid i > s(\#s) \land \mathit{dropLast}(s) \frown \langle i \rangle \frown n \in \mathsf{dom}\ t \\
\qquad\qquad \bullet \mathit{dropLast}(s) \frown \langle (i-1) \rangle \frown n \mapsto t(\mathit{dropLast}(s) \frown \langle (i) \rangle \frown n)\}
\end{array}$$

Exercise 3 – Refinement Relation: (2 points)
The following two schemata are given:

$$\begin{array}{l}
\text{— } \mathit{Abigail} \text{ —} \\
x : \mathbb{N} \\
x' : \mathbb{N} \\
\hline
x' \geq x
\end{array}
\qquad\qquad
\begin{array}{l}
\text{— } \mathit{Baldwin} \text{ —} \\
x : \mathbb{N} \\
x' : \mathbb{N} \\
\hline
x' = 2 \cdot x
\end{array}$$

Show that $Baldwin \sqsupseteq Abigail$.

Solution:

$$
\begin{aligned}
&\mathsf{pre}\, Abigail \\
&\quad \equiv [x : \mathbb{N} \mid \exists\, x' : \mathbb{N} \bullet x' \geq x] \\
&\quad \equiv [x : \mathbb{N}] \\
&\quad \equiv [x : \mathbb{N} \mid \exists\, x' : \mathbb{N} \bullet x' = 2 \cdot x] \\
&\quad \equiv \mathsf{pre}\, Baldwin \\
&(\mathsf{pre}\, Abigai \land Baldwin) \Rightarrow Abigail \\
&\quad \equiv ([x : \mathbb{N}] \land [x, x' : \mathbb{N} \mid x' = 2 \cdot x]) \Rightarrow [x, x' : \mathbb{N} \mid x' \geq x] \\
&\quad \equiv [x, x' : \mathbb{N} \mid x' = 2 \cdot x \Rightarrow x' \geq x]
\end{aligned}
$$

holds for all x, x'.