VDM-SL Sorting Algorithms

IFAD

February 17, 1994

1 Introduction

This document is part of the examples released with the *IFAD VDM-SL Toolbox* and it is located in the `vdmhome/examples` directory. The document illustrates a number of specifications of sorting algorithms and it is used in the *Getting Started* section in the *Users Manual* to introduce the basic functionalities of the Toolbox.

2 Specifications

The first example shows the standard merge sort algorithm well known from standard text books.

functions

$$
\begin{array}{l}
\mathit{MergeSort} : \mathbb{R}^* \to \mathbb{R}^* \\
\mathit{MergeSort}\,(l) \triangleq \\
\quad \mathsf{cases}\ l : \\
\qquad [] \to l, \\
\qquad [e] \to l, \\
\qquad \mathsf{others} \to \mathsf{let}\ l1 \frown l2 \in \{l\}\ \mathsf{be\ st\ abs}\ (\mathsf{len}\ l1 - \mathsf{len}\ l2) < 2\ \mathsf{in} \\
\qquad\qquad\qquad \mathsf{let}\ l\text{-}l = \mathit{MergeSort}\,(l1), \\
\qquad\qquad\qquad\quad\ \ l\text{-}r = \mathit{MergeSort}\,(l2)\ \mathsf{in} \\
\qquad\qquad\qquad \mathit{Merge}\,(l\text{-}l, l\text{-}r) \\
\quad \mathsf{end};
\end{array}
$$

$$
\begin{array}{l}
\mathit{Merge} : \mathbb{Z}^* \times \mathbb{Z}^* \to \mathbb{Z}^* \\
\mathit{Merge}\,(l1, l2) \triangleq \\
\quad \mathsf{cases\ mk\text{-}}\,(l1, l2) : \\
\qquad \mathsf{mk\text{-}}\,([], l), \mathsf{mk\text{-}}\,(l, []) \to l, \\
\qquad \mathsf{others} \to \mathsf{if\ hd}\ l1 \le \mathsf{hd}\ l2 \\
\qquad\qquad\qquad \mathsf{then}\ [\mathsf{hd}\ l1] \frown \mathit{Merge}\,(\mathsf{tl}\ l1, l2) \\
\qquad\qquad\qquad \mathsf{else}\ [\mathsf{hd}\ l2] \frown \mathit{Merge}\,(l1, \mathsf{tl}\ l2) \\
\quad \mathsf{end}
\end{array}
$$

The next example shows an implicit specification of a sorting algoritm. The `ImplSort` function cannot be interpreted as it is described here, but the other VDM-SL tools like the latex generator, the type checker and the syntax checker can process the full VDM-SL language and therefore also this specification.

types

3.0 $PosReal = \mathbb{R}$

.1 inv $r \triangleq r \geq 0$

functions

4.0 $ImplSort\,(l : PosReal^*)\; r : PosReal^*$
.1 post $IsPermutation\,(r, l) \wedge IsOrdered\,(r)$;

5.0 $IsPermutation : \mathbb{R}^* \times \mathbb{R}^* \to \mathbb{B}$
.1 $IsPermutation\,(l1, l2) \triangleq$
.2 $\quad \forall\, e \in (\text{elems } l1 \cup \text{elems } l2) \cdot$
.3 $\qquad \text{card } \{i \mid i \in \text{inds } l1 \cdot l1\,(i) = e\} =$
.4 $\qquad \text{card } \{i \mid i \in \text{inds } l2 \cdot l2\,(i) = e\};$

6.0 $IsOrdered : \mathbb{R}^* \to \mathbb{B}$
.1 $IsOrdered\,(l) \triangleq$
.2 $\quad \forall\, i, j \in \text{inds } l \cdot i > j \;\Rightarrow\; l\,(i) \geq l\,(j);$

In the following example we have changed the implicit function `ImplSort` to an explicit version `ExplSort`. This is done by changing the IsPermutation test to a generator function.

7.0 $ExplSort : PosReal^* \to PosReal^*$
.1 $ExplSort\,(l) \triangleq$
.2 $\quad \text{let } r \in Permutations\,(l) \text{ be st } IsOrdered\,(r) \text{ in}$
.3 $\quad r;$

8.0 $Permutations : \mathbb{R}^* \to \mathbb{R}^*\text{-set}$
.1 $Permutations\,(l) \triangleq$
.2 $\quad \text{cases } l :$
.3 $\qquad [\,], [\text{-}] \to \{l\},$
.4 $\qquad \text{others} \to \bigcup \{\{[l\,(i)] \frown j \mid$
.5 $\qquad\qquad j \in Permutations\,(RestSeq\,(l, i))\} \mid$
.6 $\qquad\qquad i \in \text{inds } l\}$
.7 $\quad \text{end};$

9.0 $RestSeq : \mathbb{R}^* \times \mathbb{N} \to \mathbb{R}^*$
.1 $RestSeq\,(l, i) \triangleq$
.2 $\quad [l\,(j) \mid j \in (\text{inds } l \setminus \{i\})];$

The last example is also a standard algorithm based on the principle of sorting by insertion.

10.0 $DoSort : \mathbb{R}^* \to \mathbb{R}^*$

.1 $DoSort\,(l) \triangleq$

.2 $\quad$ if $l = []$

.3 $\quad$ then $[]$

.4 $\quad$ else let $sorted = DoSort\,(\mathsf{tl}\ l)$ in

.5 $\qquad\quad InsertSorted\,(\mathsf{hd}\ l, sorted);$

11.0 $InsertSorted : PosReal \times PosReal^* \to PosReal^*$

.1 $InsertSorted\,(i, l) \triangleq$

.2 $\quad$ cases true :

.3 $\qquad (l = []) \to [i],$

.4 $\qquad (i \leq \mathsf{hd}\ l) \to [i] \frown l,$

.5 $\qquad$ others $\to [\mathsf{hd}\ l] \frown InsertSorted\,(i, \mathsf{tl}\ l)$

.6 $\quad$ end

Test Suite : sort.vdm.ts
Module : DefaultMod

Name	**#Calls**	**Coverage**
DoSort	4	$\surd$
InsertSorted	3	62%
Total Coverage		**76%**