



VDMTools

VDM-SL ソートアルゴリズム ver.1.0



How to contact:

http://fmvdm.org/ http://fmvdm.org/tools/vdmtools inq@fmvdm.org VDM information web site(in Japanese) VDMTools web site(in Japanese) Mail

VDM-SL ソートアルゴリズム 1.0

— Revised for VDMTools v9.0.6

© COPYRIGHT 2016 by Kyushu University

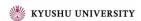
The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement.

This document is subject to change without notice



| \blacksquare | `\/.7 |
|----------------|-------|
| ш. | , i k |
| ш | //\ |

| 1 | はじめに | 1 |
|---|------|---|
| 2 | 仕様 | 1 |





1 はじめに

本書は VDM-SL Toolbox で提供される例題の一部であり、 vdmhome/examples のディレクトリの中に置かれている。本書ではソートアルゴリズムの数々の仕様を例示するが、これは Toolbox の基本的な機能を紹介するために $User\ Manual\ O$ $VDM\ Tools\ ガイドツアー\ O$ 中に用いられている。

2 仕様

最初の例題では、教科書などでよく知られている標準マージソートアルゴリズム を示す。

functions

```
MergeSort: seq of real -> seq of real
MergeSort(1) ==
  cases 1:
    Π
             -> 1,
    [e]
    others \rightarrow let 11^12 in set \{1\} be st abs (len 11 - len 12) < 2
                   let 1_l = MergeSort(l1),
                       l_r = MergeSort(12) in
                    Merge(l_l, l_r)
  end;
Merge: seq of int * seq of int -> seq of int
Merge(11,12) ==
  cases mk_{-}(11,12):
    mk_{-}([],1),mk_{-}(1,[]) \rightarrow 1,
                          -> if hd l1 <= hd l2 then
    others
```



```
[hd 11] ^ Merge(tl 11, 12)
else
[hd 12] ^ Merge(l1, tl 12)
end
pre forall i in set inds l1 & l1(i) >= 0 and
forall i in set inds l2 & l2(i) >= 0
```

次の例題では、ソートアルゴリズムの陰仕様を示す。ImplSort 関数を Toolbox においてここで述べるように処理することはできないが、latex ジェネレーター、型チェッカー、構文チェッカー、といった他の VDM-SL ツールならば全 VDM-SL 言語の処理が可能なため、この仕様も処理可能なものである。

```
PosReal = real
inv r == r >= 0

functions

ImplSort(1: seq of PosReal) r: seq of PosReal
post IsPermutation(r,1) and IsOrdered(r);

IsPermutation: seq of real * seq of real -> bool
IsPermutation(11,12) ==
  forall e in set (elems 11 union elems 12) &
      card {i | i in set inds 11 & 11(i) = e} =
      card {i | i in set inds 12 & 12(i) = e};
```

IsOrdered: seq of real -> bool

IsOrdered(1) ==

types



```
forall i,j in set inds l \& i > j \Rightarrow l(i) >= l(j);
```

以下の例題では、陰関数 ImplSort を陽関数である ExplSort に変更した。ここでは IsPermutation テストをジェネレーター関数に変更している。

```
ExplSort : seq of PosReal -> seq of PosReal
ExplSort (1) ==
  let r in set Permutations(1) be st IsOrdered(r) in r;
Permutations: seq of real -> set of seq of real
Permutations(1) ==
  cases 1:
    [],[-] \rightarrow \{1\},
    others \rightarrow dunion \{\{[l(i)]^j \mid
                          j in set Permutations(RestSeq(1,i))} |
                          i in set inds 1}
  end;
RestSeq: seq of real * nat -> seq of real
RestSeq(1,i) ==
  [l(j) \mid j \text{ in set (inds } l \setminus \{i\})]
pre i in set inds l
post elems RESULT subset elems 1 and
     len RESULT = len l - 1;
```

最後の例もまた標準アルゴリズムのもので、挿入によるソートの原則に基づいた ものである。

