Sequence Class

Shin Sahara

May, 2001

1 責任

VDM のシーケンス (sequence) を表す。

2 概要

VDM 標準の演算子にないシーケンスの機能を定義する。

class $Sequence$is subclass of $Object$

functions

public

$sum[@elem] : @elem^* \to @elem$

$sum\,(s) \triangleq$

$accsum[@elem]\,(s, 0)$

pre $\mathsf{is\_}(s, \mathbb{Z}^*) \lor \mathsf{is\_}(s, \mathbb{N}^*) \lor$

$\mathsf{is\_}(s, \mathbb{R}^*) \lor$

$\mathsf{is\_}(s, \mathbb{Q}^*)$;

$accsum[@elem] : @elem^* \times @elem \to @elem$

$accsum\,(s, sum) \triangleq$

cases s :

$[] \to sum,$

$[x] \curvearrowright xs \to accsum[@elem]\,(xs, sum + x)$

end;

public

$isAscendingTotalOrder[@elem] : (@elem \times @elem \to \mathbb{B}) \to @elem^* \to \mathbb{B}$

$isAscendingTotalOrder\,(f)(s) \triangleq$

$\forall\, i, j \in \mathsf{inds}\ s \cdot f\,(i, j) \ \Rightarrow\ f\,(s\,(i), s\,(j)) \lor s\,(i) = s\,(j);$

public

$isDescendingTotalOrder[@elem] : (@elem \times @elem \to \mathbb{B}) \to @elem^* \to \mathbb{B}$

$isDescendingTotalOrder\,(f)(s) \triangleq$

$\forall\, i, j \in \mathsf{inds}\ s \cdot f\,(i, j) \ \Rightarrow\ f\,(s\,(j), s\,(i)) \lor s\,(i) = s\,(j);$

public

$isAscending[@elem] : @elem^* \to \mathbb{B}$

$isAscending\,(s) \triangleq$

$isAscendingTotalOrder[@elem]\,(\lambda\, x : @elem, y : @elem \cdot x < y)\,(s);$

public

$isDescending[@elem] : @elem^* \to \mathbb{B}$

$isDescending\,(s) \triangleq$

$isDescendingTotalOrder[@elem]\,(\lambda\, x : @elem, y : @elem \cdot x < y)\,(s);$

public

$sort[@e] : (@e \times @e \to \mathbb{B}) \to @e^* \to @e^*$
$sort\,(f)(s) \triangleq$
cases s :
$[] \to [],$
$[x] \curvearrowright \text{-} \to$
let $l = sort[@e]\,(f)\,([s\,(i) \mid i \in \text{inds } s \cdot f\,(s\,(i), x)]),$
$m = [s\,(i) \mid i \in \text{inds } s \cdot x = s\,(i)],$
$r = sort[@e]\,(f)\,([s\,(i) \mid i \in \text{inds } s \cdot f\,(x, s\,(i))])$ in
$l \curvearrowright m \curvearrowright r$
end;

public

$ascendingSort[@e] : @e^* \to @e^*$
$ascendingSort\,(s) \triangleq$
$sort[@e]\,(\lambda\, x : @e, y : @e \cdot x < y)\,(s);$

public

$descendingSort[@e] : @e^* \to @e^*$
$descendingSort\,(s) \triangleq$
$sort[@e]\,(\lambda\, x : @e, y : @e \cdot x > y)\,(s);$

public

$take[@elem] : \mathbb{Z} \times @elem^* \to @elem^*$
$take\,(i, s) \triangleq$
cases mk- (i, s) :
mk- $(0, \text{-}) \to [],$
mk- $(n, [x] \curvearrowright xs) \to [x] \curvearrowright take[@elem]\,(dec\,(n), xs),$
mk- $(\text{-}, []) \to []$
end;
$dec : \mathbb{Z} \to \mathbb{Z}$
$dec\,(i) \triangleq$
$i - 1;$

public

$drop[@elem] : \mathbb{Z} \times @elem^* \to @elem^*$
$drop\,(i, s) \triangleq$
cases mk-(i, s) :
mk-$(n, [x] \curvearrowright xs) \to$
if $n > 0$
then $drop[@elem]\,(n - 1, xs)$
else $[x] \curvearrowright xs$,
mk-(-, []) $\to$ []
end;

public
$last[@elem] : @elem^* \to @elem$
$last\,(x) \triangleq$
cases x :
$[x] \to x$,
$[\text{-}] \curvearrowright xs \to last[@elem]\,(xs)$
end
pre $x \neq []$;

public
$fmap[@elem1, @elem2] : (@elem1 \to @elem2) \to @elem1^* \to @elem2^*$
$fmap\,(f)(l) \triangleq$
cases l :
$[] \to []$,
$[x] \curvearrowright xs \to [f\,(x)] \curvearrowright (fmap[@elem1, @elem2]\,(f)\,(xs))$
end
end *Sequence*

Test Suite : vdm.tc

Name	#Calls	Coverage
Sequence‘accsum	0	0%
Sequence‘ascendingSort	2	88%
Sequence‘dec	0	0%
Sequence‘descendingSort	0	0%
Sequence‘drop	0	0%
Sequence‘fmap	0	0%

Name	**#Calls**	**Coverage**
Sequence‘isAscending	0	0%
Sequence‘isAscendingTotalOrder	0	0%
Sequence‘isDescending	0	0%
Sequence‘isDescendingTotalOrder	0	0%
Sequence‘last	0	0%
Sequence‘sort	16	96%
Sequence‘sum	0	0%
Sequence‘take	0	0%
Total Coverage		**26%**

索 引