

Представление структур данных индуктивными семействами и доказательства их свойств

Рыбак Андрей Викторович , группа 4538
НР: Малаховски Ян Михайлович

НИУ ИТМО

Июнь 2014

Индуктивное семейство — индуктивный (рекурсивный) тип данных, который может зависеть от других типов и значений.

- AVL-деревья
- 2-3-деревья

- двоичное дерево
- заполняется слева
- значение в узле \leq значений в корнях поддеревьев

```

record  $\Sigma$  {a b} (A : Set a) (B : A  $\rightarrow$  Set b)
  : Set (a  $\sqcup$  b) where - Зависимая пара
  constructor _,_
  field fst : A ; snd : B fst
open  $\Sigma$ 

```

```

_  $\times$  _ :  $\forall$  {a b} (A : Set a)  $\rightarrow$  (B : Set b)  $\rightarrow$  Set (a  $\sqcup$  b)
A  $\times$  B =  $\Sigma$  A ( $\lambda$  _  $\rightarrow$  B) - Декартово произведение
infixr 5 _  $\times$  _ _,_

```

$\text{Rel}_2 : \text{Set} \rightarrow \text{Set}_1$

$\text{Rel}_2 A = A \rightarrow A \rightarrow \text{Set}$

```
data Tri {A : Set} (_ <_ _ ==_ _ >_ : Rel2 A) (a b : A)
  : Set where
tri< : (a < b) → ¬ (a == b) → ¬ (a > b)
      → Tri _ <_ _ ==_ _ >_ a b - меньше
tri= : ¬ (a < b) → (a == b) → ¬ (a > b)
      → Tri _ <_ _ ==_ _ >_ a b - равно
tri> : ¬ (a < b) → ¬ (a == b) → (a > b)
      → Tri _ <_ _ ==_ _ >_ a b - больше
```

$$\begin{aligned} \text{flip}_1 &: \forall \{A \ B : \text{Set}\} \{C : \text{Set}_1\} \\ &\rightarrow (A \rightarrow B \rightarrow C) \rightarrow B \rightarrow A \rightarrow C \\ \text{flip}_1 \ f \ a \ b &= f \ b \ a \end{aligned}$$

$$\begin{aligned} \text{Cmp} &: \{A : \text{Set}\} \rightarrow \text{Rel}_2 \ A \rightarrow \text{Rel}_2 \ A \rightarrow \text{Set} \\ \text{Cmp} \ \{A\} \ _ < _ \ _ == _ &= \forall (x \ y : A) \rightarrow \\ \text{Tri} \ (_ < _) \ (_ == _) \ (\text{flip}_1 \ _ < _) &x \ y \end{aligned}$$

$\text{Trans} : \{A : \text{Set}\} \rightarrow \text{Rel}_2 A \rightarrow \text{Set}$

$\text{Trans } \{A\} \text{ _rel_} = \{a \ b \ c : A\}$
 $\rightarrow (a \ \text{rel } b) \rightarrow (b \ \text{rel } c) \rightarrow (a \ \text{rel } c)$

$\text{Symmetric} : \forall \{A : \text{Set}\} \rightarrow \text{Rel}_2 A \rightarrow \text{Set}$

$\text{Symmetric } \text{ _rel_} = \forall \{a \ b\} \rightarrow a \ \text{rel } b \rightarrow b \ \text{rel } a$

$$\begin{aligned} \text{_Respects_} &: \forall \{ \ell \} \{ A : \text{Set} \} \\ &\rightarrow (A \rightarrow \text{Set } \ell) \rightarrow \text{Rel}_2 A \rightarrow \text{Set } \text{_} \\ P \text{ Respects } \text{_rel_} &= \forall \{ x \ y \} \rightarrow x \text{ rel } y \rightarrow P \ x \rightarrow P \ y \end{aligned}$$

$$\begin{aligned} \text{_Respects}_2\text{_} &: \forall \{ A : \text{Set} \} \\ &\rightarrow \text{Rel}_2 A \rightarrow \text{Rel}_2 A \rightarrow \text{Set} \\ P \text{ Respects}_2 \text{_rel_} &= \\ &(\forall \{ x \} \rightarrow P \ x \ \text{Respects } \text{_rel_}) \times \\ &(\forall \{ y \} \rightarrow \text{flip } P \ y \ \text{Respects } \text{_rel_}) \end{aligned}$$

```
data _<=_ {A : Set}
  {_<_ : Rel2 A}
  {_==_ : Rel2 A} : Rel2 A where
le :  $\forall \{x\ y\} \rightarrow x < y \rightarrow x \leq y$ 
eq  :  $\forall \{x\ y\} \rightarrow x == y \rightarrow x \leq y$ 
```

$$\text{min} : \{A : \text{Set}\} \{ _ < _ : \text{Rel}_2 A \} \{ _ == _ : \text{Rel}_2 A \}$$

$$\rightarrow (cmp : \text{Cmp } _ < _ _ == _) \rightarrow A \rightarrow A \rightarrow A$$

$$\text{min } cmp \ x \ y \text{ with } cmp \ x \ y$$

$$\dots \mid \text{tri} < _ _ _ = x$$

$$\dots \mid _ = y$$

lemma- \leq =min : {A : Set}

{_ < _ : Rel₂ A} {_ == _ : Rel₂ A}

{cmp : Cmp _ < _ == _} {a b c : A}

→ (_ <= _ { _ < _ = _ < _ } { _ == _ } a b)

→ (_ <= _ { _ < _ = _ < _ } { _ == _ } a c)

→ (_ <= _ { _ < _ = _ < _ } { _ == _ } a (min cmp b c))

lemma- \leq =min {cmp = cmp} { _ } {b} {c}

ab ac with cmp b c

... | tri< _ _ _ = ab

... | tri= _ _ _ = ac

... | tri> _ _ _ = ac

$$\text{min3} : \{A : \text{Set}\} \{ _ < _ : \text{Rel}_2 A \} \{ _ == _ : \text{Rel}_2 A \} \\ \rightarrow (\text{cmp} : \text{Cmp} _ < _ == _) \rightarrow A \rightarrow A \rightarrow A \rightarrow A$$

$$\text{min3 cmp } x \ y \ z \text{ with cmp } x \ y \\ \dots \mid \text{tri} < _ _ _ = \text{min cmp } x \ z \\ \dots \mid _ = \text{min cmp } y \ z$$

$$\text{lemma-}<=\text{min3} : \{A : \text{Set}\} \\ \{ _ < _ : \text{Rel}_2 A \} \{ _ == _ : \text{Rel}_2 A \} \\ \{ \text{cmp} : \text{Cmp} _ < _ == _ \} \{ x \ a \ b \ c : A \} \\ \rightarrow (_ <= _ \{ _ < _ = _ < _ \} \{ _ == _ \} \times a) \\ \rightarrow (_ <= _ \{ _ < _ = _ < _ \} \{ _ == _ \} \times b) \\ \rightarrow (_ <= _ \{ _ < _ = _ < _ \} \{ _ == _ \} \times c) \\ \rightarrow (_ <= _ \{ _ < _ = _ < _ \} \{ _ == _ \} \times (\text{min3 cmp } a \ b \ c))$$

```

resp<= : {A : Set} {_<_ : Rel2 A}
  {_==_ : Rel2 A}
  → (resp : _<_ Respects2 _==_)
  → (trans== : Trans _==_)
  → (sym== : Symmetric _==_)
  → (_<= _ {A}{_<_}{_==_}) Respects2 _==_
resp<= {A}{_<_}{_==_} resp trans sym = left , right
left : ∀ {a b c : A} → b == c → a <= b → a <= c
left b=c (le a<b) = le (fst resp b=c a<b)
left b=c (eq a=b) = eq (trans a=b b=c)
right : ∀ {a b c : A} → b == c → b <= a → c <= a
right b=c (le a<b) = le (snd resp b=c a<b)
right b=c (eq a=b) = eq (trans (sym b=c) a=b)

```

Транзитивность $_<=_$.

```
trans<= : {A : Set}
  {_<_ : Rel2 A} {_==_ : Rel2 A}
  → _<_ Respects2 _==_ → Symmetric _==_
  → Trans _==_ → Trans _<_
  → Trans (_<=_ {A}{_<_}{_==_})
trans<= r s t== t< (le a<b) (le b<c)
  = le (t< a<b b<c)
trans<= r s t== t< (le a<b) (eq b=c)
  = le (fst r b=c a<b)
trans<= r s t== t< (eq a=b) (le b<c)
  = le (snd r (s a=b) b<c)
trans<= r s t== t< (eq a=b) (eq b=c)
  = eq (t== a=b b=c)
```



```
module Heap (A : Set) (_ <_ _ ==_ : Rel2 A)
  (cmp : Cmp _ <_ _ ==_)
  (sym== : Symmetric _ ==_)
  (trans== : Trans _ ==_)
  (trans< : Trans _ <_)
  (resp : _ <_ Respects2 _ ==_)
where
```

```
data expanded (A : Set) : Set where  
  # : A → expanded A - элемент исходного типа  
  top : expanded A - элемент расширения
```

```
data __<E__ : Rel2 (expanded A) where
base :  $\forall \{x\ y : A\} \rightarrow x < y \rightarrow (\# x) <E (\# y)$ 
ext   :  $\forall \{x : A\} \rightarrow (\# x) <E \text{top}$ 
```

```
data __=E__ : Rel2 (expanded A) where
base :  $\forall \{x\ y\} \rightarrow x == y \rightarrow (\# x) =E (\# y)$ 
ext   :  $\text{top} =E \text{top}$ 
```

lemma-<E : $\forall \{x\} \{y\} \rightarrow (\# x) <E (\# y) \rightarrow x < y$
 trans<E : Trans _<E_

lemma=E : $\forall \{x\} \{y\} \rightarrow (\# x) =E (\# y) \rightarrow x == y$
 sym=E : Symmetric _=E_
 trans=E : Trans _=E_

respE : _<E_ Respects₂ _=E_

$_ \leq _ : \text{Rel}_2 \text{ (expanded } A)$

$_ \leq _ = _ \leq _ \{ \text{expanded } A \} \{ _ < E _ \} \{ _ = E _ \}$

$\text{trans} \leq : \text{Trans } _ \leq _$

$\text{trans} \leq = \text{trans} \leq \text{resp} E \text{ sym} = E \text{ trans} = E \text{ trans} < E$

$\text{resp} \leq : _ \leq _ \text{Respects}_2 _ = E _$

$\text{resp} \leq = \text{resp} \leq \text{resp} E \text{ trans} = E \text{ sym} = E$

```

cmpE : Cmp {expanded A} _<E_ _=E_
cmpE (# x) (# y) with cmp x y
cmpE (# x) (# y) | tri< a b c = tri<
  (base a)
  (contraposition lemma-=E b)
  (contraposition lemma-<E c)
cmpE (# x) (# y) | tri= a b c = tri=
  (contraposition lemma-<E a)
  (base b)
  (contraposition lemma-<E c)
cmpE (# x) (# y) | tri> a b c = tri>
  (contraposition lemma-<E a)
  (contraposition lemma-=E b)
  (base c)

```

```
cmpE (# x) top = tri< ext (λ ()) (λ ())  
cmpE top (# y) = tri> (λ ()) (λ ()) ext  
cmpE top top   = tri= (λ ()) ext (λ ())
```

$\text{minE} : (x\ y : \text{expanded } A) \rightarrow \text{expanded } A$

$\text{minE} = \text{min cmpE}$

$\text{lemma-}\leq\text{minE} : \forall \{a\ b\ c\} \rightarrow$

$a \leq b \rightarrow a \leq c \rightarrow a \leq (\text{minE } b\ c)$

$\text{lemma-}\leq\text{minE} =$

$\text{lemma-}\leq\text{min } \{\text{expanded } A\} \{ _ < E _ \} \{ _ = E _ \} \{ \text{cmpE} \}$

$\text{min3E} : (\text{expanded } A) \rightarrow (\text{expanded } A)$

$\rightarrow (\text{expanded } A) \rightarrow (\text{expanded } A)$

$\text{min3E } x\ y\ z = \text{min3 cmpE } x\ y\ z$

$\text{lemma-}\leq\text{min3E} : \forall \{x\ a\ b\ c\}$

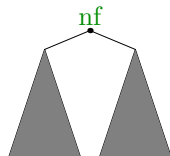
$\rightarrow x \leq a \rightarrow x \leq b \rightarrow x \leq c \rightarrow x \leq (\text{min3E } a\ b\ c)$

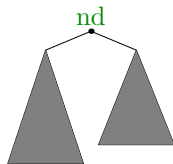
$\text{lemma-}\leq\text{min3E} =$

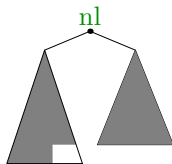
$\text{lemma-}\leq\text{min3 } \{\text{expanded } A\} \{ _ < E _ \} \{ _ = E _ \} \{ \text{cmpE} \}$

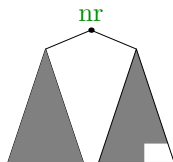

```
data HeapState : Set where  
full almost : HeapState
```

`data Heap` : (expanded A) - минимум
→ ($h : \mathbb{N}$) - высота
→ `HeapState` - заполненность
→ `Set where`
`eh` : `Heap top zero full` - Пустая куча
`nf` : $\forall \{n\} \{x\ y\} \rightarrow (p : A)$
→ ($i : (\# p) \leq x$) → ($j : (\# p) \leq y$)
→ ($a : \text{Heap } x\ n\ \text{full}$) → ($b : \text{Heap } y\ n\ \text{full}$)
→ `Heap` ($\# p$) (`succ` n) `full` - Полная куча



$$\begin{aligned}
& \text{nd} : \forall \{n\} \{x \ y\} \rightarrow (p : A) \\
& \rightarrow (i : (\# \ p) \leq x) \rightarrow (j : (\# \ p) \leq y) \\
& \rightarrow (a : \text{Heap } x \ (\text{succ } n) \ \text{full}) \\
& \rightarrow (b : \text{Heap } y \ n \ \text{full}) - \text{a b разной высоты} \\
& \rightarrow \text{Heap } (\# \ p) \ (\text{succ } (\text{succ } n)) \ \text{almost}
\end{aligned}$$


$$\begin{aligned}
& nl : \forall \{n\} \{x \ y\} \rightarrow (p : A) \\
& \rightarrow (i : (\# p) \leq x) \rightarrow (j : (\# p) \leq y) \\
& \rightarrow (a : \text{Heap } x \ (\text{succ } n) \ \text{almost}) \\
& \rightarrow (b : \text{Heap } y \ n \ \text{full}) - \text{b} - \text{полная} \\
& \rightarrow \text{Heap } (\# p) \ (\text{succ } (\text{succ } n)) \ \text{almost}
\end{aligned}$$


$$\begin{aligned}
& \text{nr} : \forall \{n\} \{x \ y\} \rightarrow (p : A) \\
& \rightarrow (i : (\# p) \leq x) \rightarrow (j : (\# p) \leq y) \\
& \rightarrow (a : \text{Heap } x \ (\text{succ } n) \ \text{full}) - \text{a} - \text{полная} \\
& \rightarrow (b : \text{Heap } y \ (\text{succ } n) \ \text{almost}) \\
& \rightarrow \text{Heap } (\# p) \ (\text{succ } (\text{succ } n)) \ \text{almost}
\end{aligned}$$


Вставка в полную кучу

$$\begin{aligned}
 \text{finsert} &: \forall \{h\ m\} \rightarrow (z : A) \\
 &\rightarrow \text{Heap } m\ h\ \text{full} \\
 &\rightarrow \Sigma \text{HeapState} \\
 &\quad (\text{Heap } (\text{minE } m\ (\# z))\ (\text{succ } h))
 \end{aligned}$$

Вставка в неполную кучу

$$\begin{aligned}
 \text{ainsert} &: \forall \{h \ m\} \rightarrow (z : A) \\
 &\rightarrow \text{Heap } m \ h \ \text{almost} \\
 &\rightarrow \Sigma \text{HeapState} \\
 &\quad (\text{Heap } (\text{minE } m \ (\# \ z)) \ h)
 \end{aligned}$$

```
data OR (A B : Set) : Set where  
  orA : A → OR A B  
  orB : B → OR A B
```


Слияние двух полных куч одной высоты

$$\begin{aligned}
 & \text{fmerge} : \forall \{x \ y \ h\} \\
 & \rightarrow \text{Heap } x \ h \text{ full} \rightarrow \text{Heap } y \ h \text{ full} \\
 & \rightarrow \text{OR} (\text{Heap } x \ \text{zero full} \times (x \equiv y) \times (h \equiv \text{zero})) \\
 & \quad (\text{Heap } (\text{minE } x \ y) (\text{succ } h) \text{ almost})
 \end{aligned}$$

Извлечение минимума из полной кучи

$$\begin{aligned}
 \text{fpop} : & \forall \{m\ h\} \rightarrow \text{Heap } m \ (\text{succ } h) \ \text{full} \\
 & \rightarrow \text{OR} \\
 & (\sum (\text{expanded } A) \\
 & \quad (\lambda x \rightarrow (\text{Heap } x \ (\text{succ } h) \ \text{almost}) \times (m \leq x)) \\
 &) \\
 & (\text{Heap } \text{top } h \ \text{full})
 \end{aligned}$$

Составление полной кучи высотой $h + 1$ из двух куч
высотой h и одного элемента

$$\begin{aligned} \text{makeH} : \forall \{x \ y \ h\} \rightarrow (p : A) \\ \rightarrow \text{Heap } x \ h \text{ full} \rightarrow \text{Heap } y \ h \text{ full} \\ \rightarrow \text{Heap } (\text{min3E } x \ y \ (\# \ p)) \ (\text{succ } h) \text{ full} \end{aligned}$$

lemma-resp : $\forall \{x \ y \ a \ b\}$
 $\rightarrow x == y \rightarrow (\# \ x) \leq a \rightarrow (\# \ x) \leq b$
 $\rightarrow (\# \ y) \leq \text{minE } a \ b$

lemma-resp $x=y \ i \ j = \text{lemma-}<=\text{minE}$
 $(\text{snd resp} \leq (\text{base } x=y) \ i)$
 $(\text{snd resp} \leq (\text{base } x=y) \ j)$

lemma-trans : $\forall \{x \ y \ a \ b\}$
 $\rightarrow y < x \rightarrow (\# \ x) \leq a \rightarrow (\# \ x) \leq b$
 $\rightarrow (\# \ y) \leq \text{minE } a \ b$

lemma-trans $y<x \ i \ j = \text{lemma-}<=\text{minE}$
 $(\text{trans} \leq (\text{le } (\text{base } y<x)) \ i)$
 $(\text{trans} \leq (\text{le } (\text{base } y<x)) \ j)$

Слияние поддеревьев nd

$ndmerge : \forall \{x \ y \ h\}$
 $\rightarrow \text{Heap } x \ (\text{succ } (\text{succ } h)) \ \text{full}$
 $\rightarrow \text{Heap } y \ (\text{succ } h) \ \text{full}$
 $\rightarrow \text{Heap } (\text{minE } x \ y) \ (\text{succ } (\text{succ } (\text{succ } h))) \ \text{almost}$

Слияние неполной кучи высотой $h + 2$ и полной кучи высотой $h + 1$ или $h + 2$

```
afmerge :  $\forall \{h \times y\}$ 
   $\rightarrow$  Heap  $x$  (succ (succ  $h$ )) almost
   $\rightarrow$  OR (Heap  $y$  (succ  $h$ ) full)
    (Heap  $y$  (succ (succ  $h$ )) full)
   $\rightarrow$  OR (Heap (minE  $x \times y$ ) (succ (succ  $h$ )) full)
    (Heap (minE  $x \times y$ ) (succ (succ (succ  $h$ ))) almost)
```

Извлечение минимума из неполной кучи

$$\begin{aligned}
 \text{арор} : \forall \{m\ h\} \rightarrow & \text{Heap } m \ (\text{succ } h) \ \text{almost} \\
 \rightarrow & \text{OR } (\Sigma \ (\text{expanded } A) \\
 & (\lambda x \rightarrow (\text{Heap } x \ (\text{succ } h) \ \text{almost}) \times (m \leq x))) \\
 & (\Sigma \ (\text{expanded } A) \\
 & (\lambda x \rightarrow (\text{Heap } x \ h \ \text{full}) \times (m \leq x)))
 \end{aligned}$$

Спасибо за внимание!