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

Куча

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

```
record Σ {a b} (A : Set a) (B : A → Set b)
  : Set (a ⊔ b) where -- Зависимая пара
  constructor _,_
  field fst : A ; snd : B fst
open Σ

_×_ : ∀ {a b} (A : Set a) → (B : Set b) → Set (a ⊔ b)
A × B = Σ A (λ _ → B) -- Декартово произведение
infixr 5 _×_ _,_
```

Отношения

$\mathsf{Rel}_2 : \mathsf{Set} \to \mathsf{Set}_1$
$\mathsf{Rel}_2\ A = A \to A \to \mathsf{Set}$

Трихотомия

$$\text{data Tri } \{A : \text{Set}\}\ (\_<\_\ \_==\_\ \_>\_ : \text{Rel}_2\ A)\ (a\ b : A)$$
$$: \text{Set where}$$
$$\text{tri}< :\ \ (a < b) \to \neg\ (a == b) \to \neg\ (a > b)$$
$$\to \text{Tri}\ \_<\_\ \_==\_\ \_>\_\ a\ b$$ - меньше
$$\text{tri}= : \neg\ (a < b) \to\ \ (a == b) \to \neg\ (a > b)$$
$$\to \text{Tri}\ \_<\_\ \_==\_\ \_>\_\ a\ b$$ - равно
$$\text{tri}> : \neg\ (a < b) \to \neg\ (a == b) \to\ \ (a > b)$$
$$\to \text{Tri}\ \_<\_\ \_==\_\ \_>\_\ a\ b$$ - больше

$$\mathrm{flip}_1 : \forall \{A\ B : \mathrm{Set}\}\ \{C : \mathrm{Set}_1\}$$
$$\quad \to (A \to B \to C) \to B \to A \to C$$
$$\mathrm{flip}_1\ f\ a\ b = f\ b\ a$$

$$\mathrm{Cmp} : \{A : \mathrm{Set}\} \to \mathrm{Rel}_2\ A \to \mathrm{Rel}_2\ A \to \mathrm{Set}$$
$$\mathrm{Cmp}\ \{A\}\ \_<\_\ \_{==}\_ = \forall\ (x\ y : A) \to$$
$$\quad \mathrm{Tri}\ (\_<\_)\ (\_{==}\_)\ (\mathrm{flip}_1\ \_<\_)\ x\ y$$

Транзитивность и симметричность

$$
\begin{aligned}
&\mathsf{Trans} : \{A : \mathsf{Set}\} \to \mathsf{Rel}_2\ A \to \mathsf{Set} \\
&\mathsf{Trans}\ \{A\}\ \_rel\_ = \{a\ b\ c : A\} \\
&\quad \to (a\ rel\ b) \to (b\ rel\ c) \to (a\ rel\ c)
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{Symmetric} : \forall\ \{A : \mathsf{Set}\} \to \mathsf{Rel}_2\ A \to \mathsf{Set} \\
&\mathsf{Symmetric}\ \_rel\_ = \forall\ \{a\ b\} \to a\ rel\ b \to b\ rel\ a
\end{aligned}
$$

$$\_\mathsf{Respects}\_ : \forall\ \{\ell\}\ \{A : \mathsf{Set}\}$$
$$\to (A \to \mathsf{Set}\ \ell) \to \mathsf{Rel}_2\ A \to \mathsf{Set}\ \_$$
$$P\ \mathsf{Respects}\ \_rel\_ = \forall\ \{x\ y\} \to x\ rel\ y \to P\ x \to P\ y$$

$$\_\mathsf{Respects}_2\_ : \forall\ \{A : \mathsf{Set}\}$$
$$\to \mathsf{Rel}_2\ A \to \mathsf{Rel}_2\ A \to \mathsf{Set}$$
$$P\ \mathsf{Respects}_2\ \_rel\_ =$$
$$(\forall\ \{x\} \to P\ x\ \ \mathsf{Respects}\ \_rel\_) \times$$
$$(\forall\ \{y\} \to \mathsf{flip}\ \ P\ y\ \mathsf{Respects}\ \_rel\_)$$

Обобщенное <=

```
data _<=_ {A : Set}
  {_<_ : Rel₂ A}
  {_==_ : Rel₂ A} : Rel₂ A where
  le : ∀ {x y} → x < y → x <= y
  eq : ∀ {x y} → x == y → x <= y
```

min

$$\mathsf{min} : \{A : \mathsf{Set}\}\ \{\_<\_ : \mathsf{Rel}_2\ A\}\ \{\_==\_ : \mathsf{Rel}_2\ A\}$$
$$\to (cmp : \mathsf{Cmp}\ \_<\_\ \_==\_) \to A \to A \to A$$
$$\mathsf{min}\ cmp\ x\ y\ \mathsf{with}\ cmp\ x\ y$$
$$\ldots \mid \mathsf{tri}{<}\ \_\ \_\ \_ = x$$
$$\ldots \mid \_ = y$$

```
lemma-<=min : {A : Set}
  {_<_ : Rel₂ A}{_==_ : Rel₂ A}
  {cmp : Cmp _<_ _==_} {a b c : A}
  → (_<=_ {_<_ = _<_} {_==_} a b)
  → (_<=_ {_<_ = _<_} {_==_} a c)
  → (_<=_ {_<_ = _<_} {_==_} a (min cmp b c))
lemma-<=min {cmp = cmp} {_} {b} {c}
  ab ac with cmp b c
... | tri< _ _ _ = ab
... | tri= _ _ _ = ac
... | tri> _ _ _ = ac
```

```
min3 : {A : Set} {_<_ : Rel₂ A} {_==_ : Rel₂ A}
  → (cmp : Cmp _<_ _==_) → A → A → A → A
min3 cmp x y z with cmp x y
... | tri< _ _ _ = min cmp x z
... | _ = min cmp y z

lemma-<=min3 : {A : Set}
  {_<_ : Rel₂ A}{_==_ : Rel₂ A}
  {cmp : Cmp _<_ _==_} {x a b c : A}
  → (_<=_ {_<_ = _<_} {_==_} x a)
  → (_<=_ {_<_ = _<_} {_==_} x b)
  → (_<=_ {_<_ = _<_} {_==_} x c)
  → (_<=_ {_<_ = _<_} {_==_} x (min3 cmp a b
```

Свойства <=

```
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
```

Расширение

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

```
data _<E_ : Rel₂ (expanded A) where
base : ∀ {x y : A} → x < y → (# x) <E (# y)
ext  : ∀ {x : A} → (# x) <E top

data _=E_ : Rel₂ (expanded A) where
base : ∀ {x y} → x == y → (# x) =E (# y)
ext  : top =E top
```

Свойства

```
lemma-<E : ∀ {x} {y} → (# x) <E (# y) → x < y
trans<E : Trans _<E_

lemma-=E : ∀ {x} {y} → (# x) =E (# y) → x == y
sym=E   : Symmetric _=E_
trans=E : Trans _=E_

respE : _<E_ Respects₂ _=E_
```

$\_\leq\_$: Rel$_2$ (expanded A)
$\_\leq\_$ = _<=_ {expanded A} {_<E_} {_=E_}

trans≤ : Trans $\_\leq\_$
trans≤ = trans<= respE sym=E trans=E trans<E

resp≤ : $\_\leq\_$ Respects$_2$ _=E_
resp≤ = resp<= respE trans=E 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)
```

$$\mathrm{cmpE}\ (\#\ x)\ \mathrm{top} = \mathrm{tri}{<}\ \mathrm{ext}\ (\lambda\ ())\ (\lambda\ ())$$
$$\mathrm{cmpE}\ \mathrm{top}\ (\#\ y) = \mathrm{tri}{>}\ (\lambda\ ())\ (\lambda\ ())\ \mathrm{ext}$$
$$\mathrm{cmpE}\ \mathrm{top}\ \mathrm{top} = \mathrm{tri}{=}\ (\lambda\ ())\ \mathrm{ext}\ (\lambda\ ())$$

minE

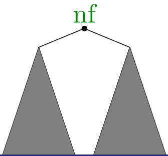
```
minE : (x y : expanded A) → expanded A
minE = min cmpE
lemma-<=minE : ∀ {a b c} →
  a ≤ b → a ≤ c → a ≤ (minE b c)
lemma-<=minE =
  lemma-<=min {expanded A}{_<E_}{_=E_}{cmpE

min3E : (expanded A) → (expanded A)
  → (expanded A) → (expanded A)
min3E x y z = min3 cmpE x y z
lemma-<=min3E : ∀ {x a b c}
  → x ≤ a → x ≤ b → x ≤ c → x ≤ (min3E a b c)
lemma-<=min3E =
  lemma-<=min3 {expanded A}{_<E_}{_=E_}{cmpE
```

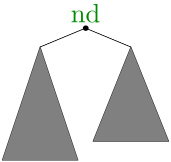
HeapState

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

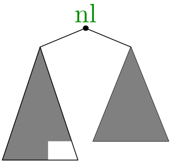
data Heap : (expanded A) -- минимум
$\to$ (h : $\mathbb{N}$) -- высота
$\to$ HeapState -- заполненность
$\to$ Set where
eh : Heap top zero full -- Пустая куча
nf : $\forall$ $\{n\}$ $\{x\ y\}$ $\to$ (p : A)
  $\to$ (i : (# p) $\leq$ x) $\to$ (j : (# p) $\leq$ y)
  $\to$ (a : Heap x n full) $\to$ (b :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



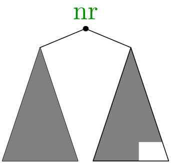
$$\mathrm{nd} : \forall \; \{n\} \; \{x \; y\} \to (p : A)$$
$$\to (i : (\# \; p) \leq x) \to (j : (\# \; p) \leq y)$$
$$\to (a : \mathrm{Heap} \; x \; (\mathrm{succ} \; n) \; \mathrm{full})$$
$$\to (b : \mathrm{Heap} \; y \; n \; \mathrm{full}) \text{ - a b разной высоты}$$
$$\to \mathrm{Heap} \; (\# \; p) \; (\mathrm{succ} \; (\mathrm{succ} \; n)) \; \mathrm{almost}$$



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



$$\mathrm{nr} : \forall \{n\} \{x\ y\} \to (p : A)$$
$$\to (i : (\#\ p) \le x) \to (j : (\#\ p) \le y)$$
$$\to (a : \mathrm{Heap}\ x\ (\mathrm{succ}\ n)\ \mathrm{full})$$ - a - полная
$$\to (b : \mathrm{Heap}\ y\ (\mathrm{succ}\ n)\ \mathrm{almost})$$
$$\to \mathrm{Heap}\ (\#\ p)\ (\mathrm{succ}\ (\mathrm{succ}\ n))\ \mathrm{almost}$$



Вставка в полную кучу

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

Вставка в неполную кучу

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

OR

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

Слияние двух полных куч одной высоты

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

Извлечение минимума из полной кучи

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

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

$$\begin{aligned}&\mathrm{makeH} : \forall\ \{x\ y\ h\} \to (p : A)\\&\quad \to \mathrm{Heap}\ x\ h\ \mathrm{full} \to \mathrm{Heap}\ y\ h\ \mathrm{full}\\&\quad \to \mathrm{Heap}\ (\mathrm{min3E}\ x\ y\ (\#\ p))\ (\mathrm{succ}\ h)\ \mathrm{full}\end{aligned}$$

Вспомогательные леммы

lemma-resp : $\forall$ $\{x\ y\ a\ b\}$
  $\to x == y \to (\#\ x) \leq a \to (\#\ x) \leq b$
  $\to (\#\ y) \leq$ minE $a\ b$
lemma-resp $x{=}y$ i j = lemma-<=minE
  (snd resp≤ (base $x{=}y$) i)
  (snd resp≤ (base $x{=}y$) j)
lemma-trans : $\forall$ $\{x\ y\ a\ b\}$
  $\to y < x \to (\#\ x) \leq a \to (\#\ x) \leq b$
  $\to (\#\ y) \leq$ minE $a\ b$
lemma-trans $y{<}x$ i j = lemma-<=minE
  (trans≤ (le (base $y{<}x$)) i)
  (trans≤ (le (base $y{<}x$)) j)

Слияние поддеревьев nd

$$\begin{aligned}
&\mathsf{ndmerge} : \forall\ \{x\ y\ h\} \\
&\to \mathsf{Heap}\ x\ (\mathsf{succ}\ (\mathsf{succ}\ h))\ \mathsf{full} \\
&\to \mathsf{Heap}\ y\ (\mathsf{succ}\ h)\ \mathsf{full} \\
&\to \mathsf{Heap}\ (\mathsf{minE}\ x\ y)\ (\mathsf{succ}\ (\mathsf{succ}\ (\mathsf{succ}\ h)))\ \mathsf{almost}
\end{aligned}$$

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

$$
\begin{aligned}
&\text{afmerge} : \forall\ \{h\ x\ y\} \\
&\quad \to \text{Heap}\ x\ (\text{succ}\ (\text{succ}\ h))\ \text{almost} \\
&\quad \to \text{OR}\ (\text{Heap}\ y\ (\text{succ}\ h)\ \text{full}) \\
&\qquad (\text{Heap}\ y\ (\text{succ}\ (\text{succ}\ h))\ \text{full}) \\
&\quad \to \text{OR}\ (\text{Heap}\ (\text{minE}\ x\ y)\ (\text{succ}\ (\text{succ}\ h))\ \text{full}) \\
&\qquad (\text{Heap}\ (\text{minE}\ x\ y)\ (\text{succ}\ (\text{succ}\ (\text{succ}\ h)))\ \text{almost})
\end{aligned}
$$

Извлечение минимума из неполной кучи

$$\mathsf{apop} : \forall \{m\ h\} \to \mathsf{Heap}\ m\ (\mathsf{succ}\ h)\ \mathsf{almost}$$
$$\to \mathsf{OR}\ (\Sigma\ (\mathsf{expanded}\ A)$$
$$(\lambda\ x \to (\mathsf{Heap}\ x\ (\mathsf{succ}\ h)\ \mathsf{almost}) \times (m \leq x)))$$

$$(\Sigma\ (\mathsf{expanded}\ A)$$
$$(\lambda\ x \to (\mathsf{Heap}\ x\ h\ \mathsf{full}) \times (m \leq x)))$$

Спасибо за внимание!