Programare logică

Tipuri abstracte de date

Algebra de termeni

Algebra iniţială

ADT

- Un *tip abstract de date* este o mulţime de date (valori) şi operaţii asociate lor, a căror descriere (specificare) este independentă de implementare.
  abstract=disassociated from any specific instance
- O algebră este este formată dintr-o mulţime de elemente şi o mulţime de operaţii. Algebrele pot modela tipuri de date.
- Două algebre izomorfe au acelaşi comportament, deci trebuie sa fie modele ale acelaşi tip de date. Aceasta asigură independenţa de implementare.

ADT

(S, Σ) signatură multisortată

$Alg(S, \Sigma)$ clasa tuturor (S, Σ)-algebrelor

- Un tip abstract de date este o clasă $\mathcal{C}$ de (S, Σ)-algebre închisă la izomorfism:

$$A \in \mathcal{C},\ A \simeq B \Rightarrow B \in \mathcal{C}.$$

- Un tip abstract de date $\mathcal{C}$ se numeşte monomorfic dacă

  $A \simeq B$ oricare ar fi A, $B \in \mathcal{C}$.

  În caz contrar, $\mathcal{C}$ se numeşte polimorfic.

ADT

- Orice tip abstract de date monomorfic este de forma $[A]_{\mathcal{K}} = \{B \in \mathcal{K} | A \simeq B\}$, unde $\mathcal{K}$ este o clasă de (S, Σ)-algebre şi $A \in \mathcal{K}$ fixată.
- Orice tip abstract de date $\mathcal{C}$ se decompune in tipuri monomorfe $\mathcal{C} = \bigcup \{[A]_{\mathcal{C}} | A \in \mathcal{C}\}$.
- $Alg(S, \Sigma)$ este un tip abstract de date polimorfic.
- În CafeObj modulele `mod*` specifică un tip abstract de date polimorfic.

Algebra iniţială

$\mathcal{K}$ clasă de (S, Σ)-algebre

- O (S, Σ)-algebră I este iniţială în $\mathcal{K}$ dacă $I \in \mathcal{K}$ şi pentru orice $B \in \mathcal{K}$ există un *unic morfism* $f : I \to B$.
- Dacă I iniţială în $\mathcal{K}$, $A \in \mathcal{K}$ şi $A \simeq I$ atunci A iniţială în $\mathcal{K}$.
- Dacă A_1 şi A_2 sunt iniţiale în $\mathcal{K}$ atunci $A_1 \simeq A_2$.
- $\mathcal{I}_{\mathcal{K}} = \{I | \ I \text{ iniţială în } \mathcal{K}\}$
  este un tip abstract de date monomorfic. În CafeObj modulele `mod!` specifică un astfel de tip monomorfic.
- Vom construi o algebră iniţială în $\mathcal{K} = Alg(S, \Sigma)$, şi anume şi anume *algebra termenilor fără variabile*.

Mulţime de variabile

(S, Σ) signatură multisortată

$|\Sigma| := \bigcup_{w,s} \Sigma_{w,s}$

$|X| := \bigcup_{s \in S} X_s$ dacă X mulţime S-sortată

O mulţime de variabile este o mulţime S-sortată X a.î.:

- $X_s \cap X_{s'} = \emptyset$ or. $s \neq s'$
- $|X| \cap |\Sigma| = \emptyset$

simbolurile de variabile sunt distincte între ele şi sunt distincte de simbolurile de operaţii din Σ

(S, Σ) signatură, X mulţime de variabile

Mulţimea S-sortată termenilor cu variabile din X, $T_\Sigma(X)$, este *cea mai mică* mulţime de *şiruri finite* peste alfabetul

$$L = \bigcup_{s \in S} X_s \cup \bigcup_{w,s} \Sigma_{w,s} \cup \{(,)\} \cup \{,\}$$

care verifică următoarele proprietăţi:

- (T1) $X_s \subseteq T_\Sigma(X)_s$
- (T2) dc. $\sigma :\to s$, at. $\boldsymbol{\sigma} \in T_\Sigma(X)_s$,
- (T3) dc. $\sigma : s_1 \cdots s_n \to s$ şi $t_i \in T_\Sigma(X)_{s_i}$ or. $i = 1, \ldots, n$ at. $\boldsymbol{\sigma(t_1, \ldots, t_n)} \in T_\Sigma(X)_s$.

• $Var(t) :=$ mulţimea variabilelor care apar în $t \in T_\Sigma$ p.12

(S, Σ) signatură multisortată

Mulţimea S-sortată termenilor fără variabile, T_Σ, este *cea mai mică* mulţime de *şiruri finite* peste alfabetul

$$L = \bigcup_{w,s} \Sigma_{w,s} \cup \{(,)\} \cup \{,\}$$

care verifică următoarele proprietăţi:

- (T2) dc. $\sigma :\to s$, at. $\boldsymbol{\sigma} \in T_{\Sigma,s}$,
- (T3) dc. $\sigma : s_1 \cdots s_n \to s$ şi $t_i \in T_{\Sigma,s_i}$ or. $i = 1, \ldots, n$
  at. $\boldsymbol{\sigma(t_1, \ldots, t_n)} \in T_{\Sigma,s}$.

• $T_\Sigma = T_\Sigma(\emptyset)$

Exemple

- $STIVA = (S, \Sigma)$

  $S = \{elem, stiva\}$, $X_{elem} = \{x, y\}$, $X_{stiva} = \emptyset$,

  $$\begin{aligned} \Sigma = \{0 :\to elem, empty :\to stiva, \\ push : elem\ stiva \to stiva, \\ pop : stiva \to stiva, \\ top : stiva \to elem\} \end{aligned}$$

- $T_\Sigma(X)_{elem} = \{\mathbf{0}, x, y, \boldsymbol{top}(\boldsymbol{pop}(\boldsymbol{empty})), \boldsymbol{top}(\boldsymbol{push}(x, \boldsymbol{empty})), \ldots\}$

  $T_\Sigma(X)_{stiva} = \{\boldsymbol{empty}, \boldsymbol{push}(y, \boldsymbol{empty}), \boldsymbol{pop}(\boldsymbol{empty}), \boldsymbol{push}(\boldsymbol{top}(\boldsymbol{empty}), \boldsymbol{empty}), \ldots\}$

- şiruri care nu sunt termeni

  $\boldsymbol{pop}(\mathbf{0})$, $(\boldsymbol{pop})\boldsymbol{top}(\boldsymbol{empty})$, $\boldsymbol{empty}(y)$, $\ldots$

Exemple

- $NATBOOL = (S, \Sigma)$
  $S = \{bool, nat\}$
  $\Sigma = \{T :\to bool, F :\to bool, 0 :\to nat,$
  $\quad s : nat \to nat,$
  $\quad \leq: nat\ nat \to bool\}$
- $T_{NATBOOL,nat} = \{\mathbf{0}, \boldsymbol{s}(\mathbf{0}), \boldsymbol{s}(\boldsymbol{s}(\mathbf{0})), \ldots\}$
  $T_{NATBOOL,bool} = \{\boldsymbol{T}, \boldsymbol{F}, \leq (\mathbf{0}, \mathbf{0}), \leq (\mathbf{0}, \boldsymbol{s}(\mathbf{0})), \ldots\}$
- şiruri care nu sunt termeni
  $\leq (\boldsymbol{T}, \boldsymbol{F}), \boldsymbol{s} \leq (\mathbf{0}), \boldsymbol{Ts}(\mathbf{0}), \ldots$

Exemple

- $AUTOMAT = (S, \Sigma)$, $S = \{intrare, stare, iesire\}$,
  $\Sigma = \{s0 :\rightarrow stare,$
  $\quad f : intrare\ stare \rightarrow stare,$
  $\quad g : stare \rightarrow iesire\}$
  $T_{AUTOMAT,stare} = \{\boldsymbol{s0}\}$, $T_{AUTOMAT,intrare} = \emptyset$,
  $T_{AUTOMAT,iesire} = \{\boldsymbol{g(s0)}\}$
- $GRAF = (S, \Sigma)$, $S = \{arc, nod\}$
  $\Sigma = \{v0 : arc \rightarrow nod, \quad v1 : arc \rightarrow nod\}$
  $T_{GRAF,arc} = T_{GRAF,nod} = \emptyset$
- Atunci când signatura defineşte un modul `mod*`, mulţimea termenilor fără variabile poate fi neinteresantă.

Expresii aritmetice

- $NATEXP = (S = \{nat\}, \Sigma)$
- $\Sigma = \{0 :\to nat, s : nat \to nat,$

  $+ : nat\ nat \to nat, * : nat\ nat \to nat\}$
- $T_{NATEXP} = \{\mathbf{0}, \boldsymbol{s}(\mathbf{0}), \boldsymbol{s}(\boldsymbol{s}(\mathbf{0})), \ldots$

  $\boldsymbol{+}(\mathbf{0}, \mathbf{0}), \boldsymbol{*}(\mathbf{0}, \boldsymbol{+}(\boldsymbol{s}(\mathbf{0}), \mathbf{0})), \boldsymbol{*}(\boldsymbol{s}(\mathbf{0}), \boldsymbol{s}(\boldsymbol{s}(\mathbf{0}))), \ldots\}$
- şiruri care nu sunt termeni

  $\boldsymbol{+}(\mathbf{0})$, $\mathbf{0}(\boldsymbol{s})\boldsymbol{s}(\mathbf{0})$, $\boldsymbol{*}(\mathbf{0})$, ...
- T_{NATEXP} este mulţimea expresiilor aritmetice peste $\mathbb{N}$.

Inducţia pe termeni

(S, Σ) signatură multisortată, X mulţime de variabile

Fie $\mathbf{P}$ o proprietate a.î. următoarele condiţii sunt satisfăcute:

- pasul iniţial:
  $\mathbf{P}(x) = true$ or. $x \in X$, $\mathbf{P}(\boldsymbol{\sigma}) = true$ or. $\sigma :\to s$,
- pasul de inducţie:
  dacă $t_1 \in T_\Sigma(X)_{s_1}$, ..., $t_n \in T_\Sigma(X)_{s_n}$ şi
  $\mathbf{P}(t_1) = \cdots = \mathbf{P}(t_n) = true$ atunci
  $\mathbf{P}(\boldsymbol{\sigma(t_1, \ldots, t_n)}) = true$ or. $\sigma : s_1 \ldots s_n \to s$.

Atunci $\mathbf{P}(t) = true$ oricare $t \in T_\Sigma(X)$.

p.6

Algebra de termeni

(S, Σ) signatură, X mulţime de variabile

Mulţimea termenilor $T_\Sigma(X) = \{T_\Sigma(X)_s\}_{s \in S}$ este (S, Σ)-algebră astfel:

- pt. $\sigma :\to s$, operaţia corespunzătoare este $T_\sigma := \boldsymbol{\sigma}$
- pt. $\sigma : s_1 \cdots s_n \to s$, operaţia corespunzătoare este
  $T_\sigma : T_\Sigma(X)_{s_1} \times \cdots \times T_\Sigma(X)_{s_n} \to T_\Sigma(X)_s$
  $T_\sigma(t_1, \ldots, t_n) := \boldsymbol{\sigma(t_1, \ldots, t_n)}$
  or. $t_1 \in T_\Sigma(X)_{s_1}$, ..., $t_n \in T_\Sigma(X)_{s_n}$

$T_\Sigma(X)$ algebra termenilor cu variabile din X

T_Σ algebra termenilor fără variabile ($X = \emptyset$)

Iniţialitate

(S,Σ) signatură multisortată

Teoremă.
Algebra T_Σ este iniţială în $Alg(S,\Sigma)$.

- or. (S,Σ)-algebra B, ex. un unic morfism $f : T_\Sigma \to B$.
- $\mathcal{I}_\Sigma := \{A | A \in Alg(S,\Sigma), A \simeq T_\Sigma\} = [T_\Sigma]$
  este un tip abstract de date monomorfic.
  Acesta reprezintă semantica unui modul `mod!` în CafeOBj care conţine numai declaraţii de sorturi şi operaţii.