Paradigme de Programare

Conf. dr. ing. Andrei Olaru

andrei.olaru@upb.ro
Departamentul de Calculatoare

2026

Cursul 2: Tipuri de date abstracte

TDA

Tipuri de date abstracte

Putem avea tipuri de date făcând abstracție de implementare?

- putem discuta despre tipuri de date fără a face apel la implementarea tipului și la modalitatea de stocare a datelor în memorie / pe disc
- definim tipurile de date într-un mod matematic
- avem nevoie să știm:

Tipuri de date abstracte

Putem avea tipuri de date făcând abstracție de implementare?

- putem discuta despre tipuri de date fără a face apel la implementarea tipului și la modalitatea de stocare a datelor în memorie / pe disc
- definim tipurile de date într-un mod matematic
- avem nevoie să știm:
  1. cum construim valorile tipului de date?
  2. cum operăm cu valorile tipului de date?

Tipuri de date abstracte

Putem avea tipuri de date făcând abstracție de implementare?

- putem discuta despre tipuri de date fără a face apel la implementarea tipului și la modalitatea de stocare a datelor în memorie / pe disc
- definim tipurile de date într-un mod matematic
- avem nevoie să știm:
  1. cum construim valorile tipului de date?
  2. cum operăm cu valorile tipului de date?

Exemplu

- Ce este un număr natural?
- Ce este o listă?
- Ce este un arbore?
- Ce este o expresie aritmetică?

Structura unui TDA

Ce avem nevoie pentru a defini un TDA?

- Operații disponibile pe tipul de date

  Ex | Exemplu

  - numere naturale: adunare, scădere
  - liste: adăugare, calcul lungime
  - expresii aritmetice: evaluare, simplificare

- Moduri de a construi valorile tipului de date

  Ex | Exemplu

  - numere naturale: adunare, scădere, incrementare, verificare pentru 0
  - liste: adăugare, eliminare, concatenare
  - expresii aritmetice: aplicare operatori

Structura unui TDA

Ce avem nevoie pentru a defini un TDA?

- Operații disponibile pe tipul de date → Operații

  Ex | Exemplu

  - numere naturale: adunare, scădere
  - liste: adăugare, calcul lungime
  - expresii aritmetice: evaluare, simplificare

- Moduri de a construi valorile tipului de date

  Ex | Exemplu

  - numere naturale: adunare, scădere, incrementare, verificare pentru 0
  - liste: adăugare, eliminare, concatenare
  - expresii aritmetice: aplicare operatori

Structura unui TDA

Ce avem nevoie pentru a defini un TDA?

- Operații disponibile pe tipul de date → Operații

  Ex | Exemplu

  - numere naturale: adunare, scădere
  - liste: adăugare, calcul lungime
  - expresii aritmetice: evaluare, simplificare

- Moduri de a construi valorile tipului de date → Constructori

  Ex | Exemplu

  - numere naturale: adunare, scădere, incrementare, verificare pentru 0
  - liste: adăugare, eliminare, concatenare
  - expresii aritmetice: aplicare operatori

Structura unui TDA

Ce avem nevoie pentru a defini un TDA?

- Operații disponibile pe tipul de date → Operații

  Ex | Exemplu

  - numere naturale: adunare, scădere
  - liste: adăugare, calcul lungime
  - expresii aritmetice: evaluare, simplificare

- Moduri de a construi valorile tipului de date → Constructori

  Ex | Exemplu

  - numere naturale: adunare, scădere, incrementare, verificare pentru 0
  - liste: adăugare, eliminare, concatenare
  - expresii aritmetice: aplicare operatori

- Axiome care garanteaza funcționarea curentă a operațiilor

Tip de date abstract

Definiție

+ | Tip de date abstract – TDA – Model matematic al unei mulțimi de valori și al operațiilor valide pe acestea.

⋮ Componente

- constructori de bază: cum se generează valorile;
- operatori: ce se poate face cu acestea;
- axiome: cum lucrează operatorii / ce restricții există.

Constructori & Axiome

Constructori de bază

Setul minim de constructori

+ | Constructor de bază – parte a unui set de constructori care este necesar și suficient pentru a obține toate valorile din mulțimea de valori a tipului de date abstract.

Ex | Exemplu

- tipul Natural:

Constructori de bază

Setul minim de constructori

+ | Constructor de bază – parte a unui set de constructori care este necesar și suficient pentru a obține toate valorile din mulțimea de valori a tipului de date abstract.

Ex | Exemplu

- tipul Natural: *zero* : *Natural* ; *succesor* : *Natural* → *Natural*

Constructori de bază

Setul minim de constructori

+ | Constructor de bază – parte a unui set de constructori care este necesar și suficient pentru a obține toate valorile din mulțimea de valori a tipului de date abstract.

Ex | Exemplu

- tipul Natural: *zero* : *Natural* ; *succesor* : *Natural* → *Natural*
- tipul Listă:

Constructori de bază

Setul minim de constructori

+ | Constructor de bază – parte a unui set de constructori care este necesar și suficient pentru a obține toate valorile din mulțimea de valori a tipului de date abstract.

Ex | Exemplu

- tipul Natural: *zero* : *Natural* ; *succesor* : $Natural \rightarrow Natural$
- tipul Listă: *null* : *List* ; *cons* : $T \times List \rightarrow List$

Constructori de bază

Setul minim de constructori

+ | Constructor de bază – parte a unui set de constructori care este necesar și suficient pentru a obține toate valorile din mulțimea de valori a tipului de date abstract.

Ex | Exemplu

- tipul Natural: *zero* : *Natural* ; *succesor* : $Natural \rightarrow Natural$
- tipul Listă: *null* : *List* ; *cons* : $T \times List \rightarrow List$
- tipul Expresie:

Constructori de bază

Setul minim de constructori

+ | Constructor de bază – parte a unui set de constructori care este necesar și suficient pentru a obține toate valorile din mulțimea de valori a tipului de date abstract.

Exemplu

- tipul Natural: *zero* : *Natural* ; *succesor* : $Natural \rightarrow Natural$
- tipul Listă: *null* : *List* ; *cons* : $T \times List \rightarrow List$
- tipul Expresie: *valoare* : $Natural \rightarrow Expresie$;
  operatie : $Expresie \times Operand \times Expresie \rightarrow Expresie$

Constructori de bază

Setul de constructori nu este unic

· Pot avea și alte seturi de constructori de bază.

· E.g. pentru listă pot avea

- *null* : *List*
- *singleton* : $T \rightarrow List$
- *concatenare* : $List \times List \rightarrow List$

Constructori de bază

Tipuri

+ | Constructorii pentru un tip T pot fi de tip:

- Nular – 0 parametri
  Ex | *zero*, *null*
- Extern – doar cu parametri care nu sunt de tip T
  Ex | *singleton*, *valoare*
- Intern – cu cel puțin un parametru de tip T
  Ex | *succesor*, *cons*, *operație*

+ Dimensiunea unei valori este numărul de constructori de bază din formula care produce valoarea

Exemplu

- "2" : $succ(succ(zero))$ — dimensiunea 3
- "[1, 7, 2]" : $cons(1, cons(7, cons(2, null)))$ — dimensiunea 4

- Constructorii nulari și externi produc întotdeauna valori de dimensiune 1
- Constructorii interni produc întotdeauna valori de dimensiune > 1

Axiome
Exemple

Exemplu

- $concatenare(cons(x, L1), L2) = cons(x, concatenare(L1, L2))$
- $concatenare(null, L) = L$

Axiome
Exemple

Exemplu

- $concatenare(cons(x, L1), L2) = cons(x, concatenare(L1, L2))$
- $concatenare(null, L) = L$
- $adunare(succesor(n), m) = succesor(adunare(n, m))$
- $adunare(zero, m) = m$

Axiome
Exemple

Ex
Exemplu

- $concatenare(cons(x, L1), L2) = cons(x, concatenare(L1, L2))$
- $concatenare(null, L) = L$
- $adunare(succesor(n), m) = succesor(adunare(n, m))$
- $adunare(zero, m) = m$
- $evaluare(operatie(E1, +, E2)) = adunare(evaluare(E1), evaluare(E2))$

Axiome

Definiție

+ | Axiomele definesc comportamentul operatorilor pe toate valorile tipului.

Pentru fiecare operator avem nevoie de câte o axiomă pentru fiecare constructor de bază.

Inducție structurală

Inducție

Inducție matematică

· La inducția matematică, demonstrăm cazul de bază (e.g. $n = 0$) și presupunând ca adevărată afirmație pentru valorile până la $n - 1$, demonstrăm adevărul afirmației pentru n.

Putem folosi inducția matematică pentru tipul *Natural*, dar pentru celelalte?

Inducție
Inducție matematică

· La inducția matematică, demonstrăm cazul de bază (e.g. $n = 0$) și presupunând ca adevărată afirmație pentru valorile până la $n - 1$, demonstrăm adevărul afirmației pentru n.

Putem folosi inducția matematică pentru tipul *Natural*, dar pentru celelalte?

Pentru celelate, folosim inducția structurală.

Inducție structurală
Definiție

+ | Inducția structurală permite demonstrarea că dacă o afirmație despre o operație este adevărată pentru cazurile de bază (constructori nulari și interni) și dacă, presupunând că afirmația este adevărată pentru valorile de dimensiune $n-1$, putem demonstra că afirmația este adevărată pentru valorile de dimensiune n, atunci afirmația este adevărată întotdeauna.

Inducție structurală

Exemplu (1)

Exemplu

Pentru $add : Natural \times Natural \to Natural$

cu axiomele

- ADD1 $add(zero, n) = n$
- ADD2 $add(succesor(m), n) = succesor(add(m, n))$

demonstrăm că
$\forall a, b, c \in Natural \,.\, add(a, add(b, c)) = add(add(a, b), c)$ (asociativitate)

Caz de bază:

$add(zero, add(b, c)) = add(b, c)$ (conform ADD1)

$add(add(zero, b), c) = add(b, c)$ (conform ADD1)

Inducție structurală

Exemplu (1)

Exemplu

Pentru $add : Natural \times Natural \to Natural$

cu axiomele

- ADD1 $add(zero, n) = n$
- ADD2 $add(succesor(m), n) = succesor(add(m, n))$

demonstrăm că
$\forall a, b, c \in Natural \,.\, add(a, add(b, c)) = add(add(a, b), c)$ (asociativitate)

Caz de bază:

$add(zero, add(b, c)) = add(b, c)$ (conform ADD1)

$add(add(zero, b), c) = add(b, c)$ (conform ADD1) □

Inducție structurală
Exemplu (2)

cu axiomele

- ADD1 $add(zero, n) = n$
- ADD2 $add(succesor(m), n) = succesor(add(m, n))$

Ipoteză inductivă: $add(x, add(b, c)) = add(add(x, b), c)$

Pas de inducție:

stânga: $add(succ(x), add(b, c)) = succ(add(x, add(b, c)))$ (conform ADD2)

dreapta: $add(add(succ(x), b), c) = add(succ(add(x, b)), c)$ (conform ADD2)
$= succ(add(add(x, b), c))$ (conform ADD2)

dar: $add(x, add(b, c)) = add(add(x, b), c)$ (conform ipoteză)

Inducție structurală
Exemplu (2)

cu axiomele

- ADD1 $add(zero, n) = n$
- ADD2 $add(succesor(m), n) = succesor(add(m, n))$

Ipoteză inductivă: $add(x, add(b, c)) = add(add(x, b), c)$

Pas de inducție:

stânga: $add(succ(x), add(b, c)) = succ(add(x, add(b, c)))$ (conform ADD2)

dreapta: $add(add(succ(x), b), c) = add(succ(add(x, b)), c)$ (conform ADD2)
$= succ(add(add(x, b), c))$ (conform ADD2)

dar: $add(x, add(b, c)) = add(add(x, b), c)$ (conform ipoteză) □

Sfârșitul cursului 2

Elemente esențiale

- Tipuri de date abstracte:
- Constructori, constructori de bază, operatori, axiome
- Inducție structurală

+ Dați feedback la acest curs aici:
[https://forms.gle/ETsHHPvn2nDgF7q27]