

Kazalo

1 Polne mreže, negibne točke in izrek Knaster-Tarskega	2
1.1 Polne mreže	2
1.2 Monotone preslikave in negibne točke	2
1.3 Izrek Knaster-Tarskega	2
1.4 Lastnosti negibnih točk in vloženih negibnih točk	3
2 Odsekoma linearne preslikave	4
2.1 Pogojni linearni izrazi	4
2.2 Teorija linearne aritmetike prvega reda	5
2.3 Lokalni algoritem	5
3 Osnovni algoritmi	7
3.1 Dagger f	7
3.2 Skupne negibne točke	7
3.2.1 Bekičev izrek	8

1 Polne mreže, negibne točke in izrek Knaster-Tarskega

- Polne mreže;
- Monotone preslikave med polni mreži;
- Najmanjša in največja negibni točki monotonih preslikav;

1.1 Polne mreže

- (E, \leq) je delno urejena množica, kjer je \leq delna urejenost, tj. relacija, ki je refleksivna, antisimetrična in tranzitivna;
- Spodnja in zgornja meja množice $X \subseteq E$;
- Najmanjša zgornja $\bigvee X$ (največja spodnja $\bigwedge X$) meja množice $X \subseteq E$;
- Simetrija med inf in sup;

Definicija 1.1. Naj bo (E, \leq) delno urejena množica:

- (E, \leq) je *mreža*, če za poljubna elementa $x, y \in E$, ima množica $\{x, y\}$ najmanjšo zgornjo mejo $x \vee y$ in največjo spodnjo mejo $x \wedge y$.
- (E, \leq) je *polna mreža*, če vsaka podmnožica $X \subseteq E$ ima najmanjšo zgornjo mejo $\bigvee X$ in največjo spodnjo mejo $\bigwedge X$.

Trditev 1.2. Naj bosta E, F polni mreži. Definiramo (produktna urejenost)

$$\forall(x, y), (x', y') \in E \times F . (x, y) \leq (x', y') \iff x \leq_E x' \wedge y \leq_F y'.$$

Tedaj je $(E \times F, \leq)$ polna mreža.

Opomba 1.3. Trditev lahko posplošimo na produkt n polnih mrež.

1.2 Monotone preslikave in negibne točke

Definicija 1.4. Naj bosta (E, \leq_E) in (F, \leq_F) delno urejeni množici. Preslikava $f : E \rightarrow F$ je *monotona*, če

$$\forall x, y \in E . x \leq_E y \implies f(x) \leq_F f(y).$$

Opomba 1.5. Monotona preslikava v tem kontekstu je isto kot naraščajoča preslikava.

Definicija 1.6. Naj bo E poljubna množica in $f : E \rightarrow E$ preslikava. *Negibna točka* preslikave f je element $x \in E$, da velja $f(x) = x$. Množico vseh negibnih točk preslikave f označimo z $\text{Fix}(f)$.

Opomba 1.7. Če je E delno urejena množica, potem tudi $\text{Fix}(f)$ delno urejena množica, ki je lahko prazna.

1.3 Izrek Knaster-Tarskega

Izrek 1.8 (Knaster-Tarski). *Naj bo (E, \leq) polna mreža in $f : E \rightarrow E$ monotona preslikava. Tedaj $\bigvee \text{Fix}(f)$ in $\bigwedge \text{Fix}(f)$ sta elementi $\text{Fix}(f)$.*

Dokaz izreka nam da na način, kako lahko dokažemo, da je podan element najmanjša ali največja negibna točka monotone preslikave f .

Posledica 1.9. *Naj bo E polna mreža in $f : E \rightarrow E$ monotona preslikava. Tedaj je $e \in E$ najmanjša negibna točka preslikave f natanko tedaj, ko*

1. $\forall x \in E . f(x) \leq x \implies e \leq x$ in
2. $f(e) \leq e$.

Podobno za največjo negibno točko.

1.4 Lastnosti negibnih točk in vloženih negibnih točk

Oznake:

- Najmanjša negibna točka preslikave f :
 - $\mu x . f(x)$;
 - $f^\dagger(x)$.
- Največja negibna točka preslikave f :
 - $\nu x . f(x)$.
- Naj bodo E polna mreža, F delno urejena množica in $f : E \times F \rightarrow E$ monotona preslikava v vsaki spremenljivki, tj.

$$\begin{aligned} \forall x \in E . \forall s, t \in F . s \leq t &\implies f(x, s) \leq f(x, t) \leq f(x, s), \\ \forall y \in F . \forall u, v \in E . u \leq v &\implies f(u, y) \leq f(v, y). \end{aligned}$$

Za vsak $y \in F$ definiramo preslikavo $f_y : E \rightarrow E$, $f_y(x) := f(x, y)$. Označimo z $\mu x . f(x, y)$ preslikavo $\mu x . f(x, y) : F \rightarrow E$ s predpisom $\mu x . f(x, y)(y) := \mu x . f_y(x)$.

Trditev 1.10. *Naj bo E polna mreža in F delno urejena množica. Če je preslikava $f : E \times F \rightarrow E$ monotona v obeh spremenljivkah, tedaj sta $\mu x . f(x, y)$ in $\nu x . f(x, y)$ monotoni preslikavi iz F v E .*

Lema 1.11 (Zlata lema μ -računa). *Naj bo E polna mreža in $f : E \times E \rightarrow E$ monotona preslikava v obeh spremenljivkah. Tedaj*

$$\mu x . \mu y . f(x, y) = \mu x . f(x, x) = \mu y . \mu x . f(x, y)$$

in

$$\nu x . \nu y . f(x, y) = \nu x . f(x, x) = \nu y . \nu x . f(x, y).$$

2 Odsekoma linearne preslikave

- Odsekoma linearne preslikave (*piecewise linear functions*)
 - Razdelimo domeno na kosi;
 - Na vsakem kosu imamo linearni izraz (*linear expression*);
 - Preslikava ni nujno zvezna;
 - Osnovna domena je $[0, 1]^n$.
- Pogojni linearni izrazi (*conditioned linear expression*)
 - Racionalni linearni izrazi;
 - Pogojni linearni izrazi.
- Teorija linearne aritmetike prvega reda
- Lokalni algoritem

2.1 Pogojni linearni izrazi

Definicija 2.1. *Linearji izraz* v spremenljivkah x_1, x_2, \dots, x_n je izraz

$$q_0 + q_1 x_1 + q_2 x_2 + \dots + q_n x_n,$$

kjer so $q_0, \dots, q_n \in \mathbb{R}$. Pravimo, da je linearji izraz *racionalni*, če so $q_0, \dots, q_n \in \mathbb{Q}$.

Oznake:

- $e(x_1, \dots, x_n) := q_0 + q_1 x_1 + q_2 x_2 + \dots + q_n x_n$;
- $e(\vec{r}) := q_0 + q_1 r_1 + q_2 r_2 + \dots + q_n r_n$, kjer $\vec{r} = (r_1, \dots, r_n) \in \mathbb{R}^n$.

Linearni izrazi so zaprti za nadomestitev, torej za podan linearji izraz $e(x_1, \dots, x_n)$ in linearni izrazi $e_1(y_1, \dots, y_m), \dots, e_n(y_1, \dots, y_m)$, pišemo $e(e_1, \dots, e_n)$ za nadomestni linearji izraz v spremenljivkah y_1, \dots, y_m , dobljeni z nadomeščanjem spremenljivke x_1 z izrazom $e_1(y_1, \dots, y_m)$, spremenljivke x_2 z izrazom $e_2(y_1, \dots, y_m)$ in tako naprej.

Definicija 2.2. *Pogojni linearji izraz* je par $C \vdash e$, kjer je e linearji izraz in C končna množica neenačb med linearni izrazi, torej vsak element množice C je ene izmed oblik

$$e_1 \leq e_2, \quad e_1 < e_2.$$

Za $\vec{r} \in \mathbb{R}^n$ z $C(\vec{r})$ označimo konjunkcijo neenačb, kjer nadomestimo spremenljivke v C z števili r_1, \dots, r_n . Z pomočjo pogojnega linearnega izraza lahko opišemo en kos odsekoma linearne preslikave:

- Domena kosa je $\{\vec{r} \in \mathbb{R}^n \mid C(\vec{r})\}$, tj. množica vseh vektorjev, ki ustrezajo vsem neenačbam;
- Linearji izraz e podaja linearno preslikavo nad domeno.

Opomba 2.3.

- Domena ni nujno odprta ali zaprta;
- Domena lahko prazna;
- Zaprtje domene je konveksni mnogokotnik ($\approx B^n$).

Definicija 2.4. Pravimo, da je preslikava $f : [0, 1]^n \rightarrow [0, 1]$ *odsekoma linearna*, če obstaja končna množica \mathcal{F} pogojnih linearnih izrazov v spremenljivkah x_1, \dots, x_n , da velja:

1. $\forall \vec{r} \in [0, 1]^n . \exists (C \vdash e) \in \mathcal{F} . C(\vec{r})$ in

2. $\forall \vec{r} \in [0, 1]^n . \forall (C \vdash e) \in \mathcal{F} . C(\vec{r}) \implies f(\vec{r}) = e(\vec{r})$.
 Pravimo, da \mathcal{F} predstavlja preslikavo f .

Opomba 2.5. Dva pogojna linearne izraza ni nujno bosta imela disjunktno domeno. V tem primeru, morajo se linearne izraze e_1 in e_2 ujemati na presek.

2.2 Teorija linearne aritmetike prvega reda

- Linearni izrazi kot logični izrazi (*terms*);
- Atomarne formule so neenačbe med linearni izrazi;
- Enakost $e_1 = e_2$ pišemo kot $(e_1 \leq e_2) \wedge (e_1 \geq e_2)$;
- Negacijo $\neg(e_1 \leq e_2)$ pišemo kot $(e_1 > e_2)$;
- Ima lastnost *eliminacije kvantifikatorjev*, tj. vsaka formula ima ekvivalentno obliko brez kvantifikatorjev;
- Vsako formulo lahko zapišemo v disjunktni normalni formi, torej kot disjunkcijo konjukcij atomarnih formul.

Trditev 2.6. Preslikava $f : [0, 1]^n \rightarrow [0, 1]$ je odsekoma linearna natanko tedaj, ko njen graf $\{(\vec{x}, y) \in [0, 1]^{n+1} \mid f(\vec{x}) = y\}$ lahko definiramo z formulo $F(x_1, \dots, x_n, y)$ v teoriji linearne aritmetike prvega reda.

Izrek 2.7. Naj bo $f : [0, 1]^{n+1} \rightarrow [0, 1]$ odsekoma linearna preslikava, ki je monotona v zadnji spremenljivki x_{n+1} , tj.

$$\forall t, s \in [0, 1] . t \leq s \implies \forall x_1, \dots, x_n \in [0, 1] . f(x_1, \dots, x_n, t) \leq f(x_1, \dots, x_n, s).$$

Tedaj sta

$$\mu x_{n+1} . f(x_1, \dots, x_n, x_{n+1}) \quad \text{in} \quad \nu x_{n+1} . f(x_1, \dots, x_n, x_{n+1})$$

odsekoma linearni preslikavi iz $[0, 1]^n$ v $[0, 1]$.

Opomba 2.8. Zahtevamo monotonost, da smo prepričani, da najmanjša in največja fiksni točki obstajata, kar sledi iz izreka Knaster-Tarski.

2.3 Lokalni algoritem

Za izračun odsekoma linearne preslikave dovolj, da imamo algoritem, ki za podan vektor \vec{r} vrača vrednost $f(\vec{r})$. Za bolj kompleksne manipulacije, kot so izračun preslikave, ki poišče negibno točko izvorne preslikave $\mu x_{n+1} . f(\dots)$, potrebujemo več informacije. Ena možnost je – sprehajanje po množici \mathcal{F} , vendar ta je lahko zelo velika. Zato bomo uporabljali manj eksplicitno predstavitev preslikave f , ki jo imenujemo *lokalni algoritem*. Ta algoritem za podano točko v domeni \vec{r} vrača ustrezni pogojni linearni izraz.

Definicija 2.9. Lokalni algoritem za odsekomo linearno preslikavo $f : [0, 1]^n \rightarrow [0, 1]$, je algoritem, ki za podano točko $\vec{r} \in [0, 1]^n$ vrača pogojni linearni izraz $C \vdash e$, da velja

1. $C(\vec{r})$ drži;
2. $\forall \vec{s} \in [0, 1]^n . C(\vec{s}) \implies f(\vec{s}) = e(\vec{s})$.

Tudi le končno mnogo različnih $C \vdash e$ lahko dobimo.

Opomba 2.10. Vsako eksplicitno predstavitev \mathcal{F} lahko enostavno konvertiramo v lokalni algoritem, z katerim lahko enostavno izračunamo vrednost preslikave f v podani točki.

Za izračun preslikave, ki poišče negibno točko izvirne preslikave $\mu x_{n+1} \cdot f(\dots)$ bomo potrebovali le predstavitev izvirne preslikave f z lokalnim algoritmom. Rezultat dela tega algoritma bo lokalni algoritem za preslikavo $\mu x_{n+1} \cdot f(\dots)$.

3 Osnovni algoritmi

Naj bosta X in Y končni polni mreži.

3.1 Dagger f

Naj bo $f : X \times Y \rightarrow Y$ naraščajoča preslikava. Želimo opisati algoritmom $f^\dagger : X \rightarrow Y$, kjer je

$$\forall x \in X . f^\dagger(x) = \mu y . f(x, y).$$

Algorithm 1: Dagger f

Data: $f \in Y^{X \times Y}$, $x \in X$

Result: $\mu y . f(x, y)$

```
a ← 0                                     /* 0 je najmanjši element Y */
repeat
  prev ← a
  a ← f(x, prev)
until a ≠ prev
return a
```

Od zdaj naprej s terminom *časovna zahtevnost* bomo mislili število aproksimacij.

Trditev 3.1. Algoritem se ustavi.

Trditev 3.2. Algoritem izračuna najmanjšo negibno točko.

Trditev 3.3. Časovna zahtevnost algoritma je $O(\text{height}(Y))$, kjer je $\text{height}(Y)$ dolžina največje verige v Y .

3.2 Skupne negibne točke

Naj bo $F : X \times Y \times \cdots \times Y \rightarrow Y \times \cdots \times Y$, $F = (f_1, \dots, f_n)$ naraščajoča, kjer so $f_i : X \times Y \times \cdots \times Y \rightarrow Y$. Želimo opisati algoritmom $F^\dagger : X \rightarrow Y \times \cdots \times Y$, kjer je

$$\forall x \in X . F^\dagger(x) = \mu \vec{y} . F(x, \vec{y}).$$

Lahko bi modificirali algoritem 1.

Algorithm 2: Dagger F

Data: $F \in (Y^n)^{X \times Y^n}$, $x \in X$

Result: $\mu \vec{y} . F(x, \vec{y})$

```
new array A[n] ← [0, ..., 0]           /* [0, ..., 0] je najmanjši element Y^n */
repeat
  prev ← A
  A ← F(x, prev)
until A ≠ prev
return A
```

Trditev 3.4. Algoritem se ustavi.

Trditev 3.5. Algoritem izračuna najmanjšo negibno točko.

Trditev 3.6. Časovna zahtevnost algoritma je $O(n \text{height}(Y))$.

Opomba 3.7. Tukaj se pojavi prvi odprt problem. Ne vemo, kako lahko posplošimo algoritem na odsekoma linearne funkcije.

3.2.1 Bekičev izrek

Naslednja dva algoritma slonijo na Bekičevem izreku.

Naj bo $F : X \times Y^n \rightarrow Y^n$, $F = (f_0, \dots, f_{n-1})$ naraščajoča, kjer so $f_i : X \times Y^n \rightarrow Y$. Poskušamo rešiti sistem

$$\begin{aligned} y_0 &= f_0(x, y_0, \dots, y_{n-1}) \\ &\vdots \\ y_{n-1} &= f_{n-1}(x, y_0, \dots, y_{n-1}) \end{aligned}$$

z uporabo algoritma 1 za vsako posamezno spremenljivko y_0, \dots, y_{n-1} .

Algorithm 3: Skupne negibne točke f_0, \dots, f_{n-1}

Data: $f_0, \dots, f_{n-1} \in Y^{X \times Y^n}$, $x \in X$
Result: y_0, \dots, y_{n-1}

```

new array  $A[n] \leftarrow [0, \dots, 0]$            /*  $[0, \dots, 0]$  je najmanjši element  $Y^n$  */
 $i \leftarrow 0$ 
repeat
     $prev \leftarrow A[i]$ 
     $A[i] \leftarrow f_i(x, A)$ 
    if  $A[i] = prev$  then
        |  $i \leftarrow i + 1$ 
    else
        |  $j \leftarrow 0$ 
        | while  $j < i$  do
        |   |  $A[j] \leftarrow 0$ 
        |   end
        |    $i \leftarrow 0$ 
    end
until  $i < n$ 
return  $A$ 

```

Trditev 3.8. Algoritem se ustavi.

Trditev 3.9. Algoritem izračuna najmanjšo negibno točko.

Trditev 3.10. Časovna zahtevnost algoritma je $O(\text{height}(Y)^n)$.

Opomba 3.11. Težava tega algoritma je časovna zahtevnost. Plus pa je, da ga lahko posplošimo na odsekoma linearne funkcije.

Ideja novega algoritma je v tem, da pozabimo na notranjo `while` zanko, tj.

Algorithm 4: Skupne negibne točke f_0, \dots, f_{n-1} (nov algoritem)

Data: $f_0, \dots, f_{n-1} \in Y^{X \times Y^n}$, $x \in X$

Result: y_0, \dots, y_{n-1}

new array $A[n] \leftarrow [0, \dots, 0]$ /* $[0, \dots, 0]$ je najmanjši element Y^n */

$i \leftarrow 0$

repeat

```

    prev ← A[i]
    A[i] ← f_i(x, A)
    if A[i] = prev then
        | i ← i + 1
    else
        | i ← 0
    end

```

until $i < n$

return A

Trditev 3.12. Algoritem se ustavi.

Trditev 3.13. Algoritem izračuna najmanjšo negibno točko.

Trditev 3.14. Časovna zahtevnost algoritma je $O(n \text{height}(Y))$.