

Teoretične osnove računalništva

Zapiski predavanj 2010/2011

25. februar 2011



This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3.0
Unported License

Kazalo

| | | |
|----------|-------------------------------------------------------------------|----------|
| 1 | Uvod | 2 |
| 1.1 | Dokazovaje | 2 |
| 1.1.1 | Dokaz s konstrukcijo | 2 |
| 1.1.2 | Dokaz z indukcijo | 2 |
| 1.1.3 | Dokaz s protislovjem | 2 |
| 2 | Regularni jeziki | 4 |
| 2.1 | Uvod | 4 |
| 2.2 | Regularni Izrazi | 5 |
| 2.2.1 | Jezik regularnih izrazov | 5 |
| 2.2.2 | Primeri regularnih izrazov | 5 |
| 2.3 | Končni avtomati | 6 |
| 2.3.1 | Nedeterministični končni avtomati z ϵ -prehodi | 6 |
| 2.3.2 | Nedeterministični končni avtomati | 6 |
| 2.3.3 | Deterministični končni avtomat | 6 |
| 2.3.4 | Jeziki končnih avtomatov | 6 |
| 2.3.5 | Regularne gramatike | 6 |
| 2.4 | Prevedba med izvedbami regularnih jezikov | 7 |
| 2.4.1 | Končni avtomat \rightarrow Regularni izraz | 7 |
| 2.5 | Primeri izvedb regularnih jezikov | 7 |
| 2.6 | Ohranjanje regularnosti jezikov | 7 |
| 3 | Slovar | 8 |

Poglavje 1

Uvod

1.1 Dokazovaje

1.1.1 Dokaz s konstrukcijo

Dokaz obstoja nekega objekta je to, da nam objekt uspe skonstruirati.

Primeri:

Primer 1: Ali za vsako število elementov, večje od 4, obstaja graf ki ima natanko 3 liste?

Primer 2: $|\mathbb{R}| = |[0, 1)|$

1.1.2 Dokaz z indukcijo

Če je množica induktivni razred¹, lahko z matematično indukcijo dokazujemo neko lastnost članov množice.

Induktivni razred I sestavlja:

- Baza indukcije - najbolj osnovna množica elementov (osnovni razred)
- Pravila generiranja - kako iz elementov baze gradimo nove elemente (množico)

Primeri:

Primer 1: Induktivni razred naravnih števil (\mathbb{N})

- Baza: $1 \in \mathbb{N}$
- Pravila generiranja: $n \in \mathbb{N} \implies n + 1 \in \mathbb{N}$

Primer 2: [Hilbertove krivulje](#)²

1.1.3 Dokaz s protislovjem

Vzamemo nasprotno trditev, od tiste, ki jo želimo preveriti in pokažemo, da to vodi v protislovje.

Primeri:

Primer 1: Praštevil je končno mnogo

- Predpostavimo, da poznamo vsa praštevila:
 $P = \{2, 3, 5, \dots, p\}$, kjer je p zadnje praštevilo
- Po definiciji obstajajo le praštevila in sestavljena števila (to so taka, ki jih lahko razstavimo na prafaktorje).

¹Glej slovarček na koncu.

²http://en.wikipedia.org/wiki/Hilbert_curve

- Če pomnožimo vsa znana praštevila iz P in prištejemo 1 dobimo število, ki se ga ne da razstaviti na prafaktorje iz množice P :
 $q = 2 * 3 * 5 * \dots * p + 1$
- Torej je q ali praštevilo (ker ni sestavljeno), ali pa število, sestavljeno iz prafaktorjev, ki jih ni v množici P .
- Oboje kaže na to, da v množici P nimamo vseh praštevil in da to velja za vsako končno množico praštevil.

Primer 2: $\sqrt[3]{2}$ je racionalno število

- $\sqrt[3]{2} = \frac{a}{b}$ ker je $\frac{a}{b}$ racionalen ulomek ga lahko okrajšamo in si ga od sedaj predstavljamo okrajšanega $GCD(a, b) = 1$
- $2 = \left(\frac{a}{b}\right)^3$
- $2b^3 = a^3$ tukaj vidimo da je a sodo število, torej lahko pišemo $a = 2k$
- $2b = (2k)^3$
- $2b = 8k$
- $b = 4k$ ker je b tudi sodo število, vidimo da $GCD(a, b) = 1$ ne drži, torej smo prišli v protislovje.

Poglavje 2

Regularni jeziki

2.1 Uvod

Oznake

- a - simbol oz. beseda dolžine 1
- Σ - abeceda oz. končna neprazna množica simbolov
- w - besede, nizi oz. poljubno končno zaporedje simbolov $w_1 w_2 \dots w_n$.
- $|w|$ - dolžina niza - je 0 za $w = \varepsilon$
- ε - prazen niz oz. niz dolžine 0
- Σ^* - vse možne besede abecede

Operacije

- Stik
 - Stik nizov:

$$w = w_1 w_2 \dots w_n$$

$$x = x_1 x_2 \dots x_m$$

$$wx = w_1 w_2 \dots w_n x_1 x_2 \dots x_m$$

- Stik množic:

$$A = \{w_1, w_2, \dots, w_n\}$$

$$B = \{x_1, x_2, \dots, x_m\}$$

$$A \circ B = \{w_i x_j \mid w_i \in A \wedge x_j \in B\}$$

- Potenciranje

$$A^0 = \{\varepsilon\}$$

$$A^k = A \circ A \circ \dots \circ A = \bigcirc_{i=0}^k A^i$$

- Iteracija

$$A^* = A^0 \bigcup A^1 \bigcup A^2 \dots = \bigcup_{i=0}^{\infty} A^i$$

Regularni jezik

Def.: Regularni jezik L nad abecedo Σ je poljubna podmnožica Σ^*

$$L \subseteq \Sigma^*$$

Primeri:**Primer 1:** Prazen jezik: $L_1 = \{\}$ **Primer 2:** Jezik, ki vsebuje ε (ni prazen): $L_2 = \{\varepsilon\}$ **Primer 3:** Jezik, ki vsebuje nize "a, aa, ab": $L_3 = \{a, aa, ab\}$ **2.2 Regularni Izrazi****Def.:** Osnovni izrazi:

- \emptyset je opisuje prazen jezik $L(\emptyset) = \{\}$
- ε opisuje jezik $L(\varepsilon) = \{\varepsilon\}$
- a opisuje jezik $L(a) = \{a\}$, $a \in \Sigma$

Def.: Pravila za generiranje kompleksnejših izrazov:

- $(r_1 + r_2)$ opisuje unijo jezikov $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
- $(r_1 r_2)$ opisuje stik jezikov $L(r_1 r_2) = L(r_1) \cdot L(r_2)$
- (r^*) opisuje iteracijo jezika $(L(r))^*$

2.2.1 Jezik regularnih izrazov**Def.:** Jezik ki ga opisuje poljubni regularni izraz, je regularni jezik.**Primeri:****Primer 1:** $\{\}$ je regularni jezik**Primer 2:** $\{0^n 1^n \mid n \geq 0\}$ ni regularni jezik**2.2.2 Primeri regularnih izrazov****Primeri:****Primer 1:** Opiši vse nize, ki se končajo z nizom 00 v abecedi $\Sigma = \{0, 1\}$.

$$r = (0 + 1)^* 00$$

Primer 2: Opiši vse nize, pri katerih so vsi a -ji pred b -ji in vsi b -ji pred c -ji v abecedi $\Sigma = \{a, b, c\}$.

$$a^* b^* c^*$$

Primer 3: Opiši vse nize, ki vsebujejo vsaj dva niza 'aa', ki se ne prekrivata v abecedi $\Sigma = \{a, b, c\}$.

$$(a + b + c)^* aa(a + b + c)^* aa(a + b + c)^*$$

Primer 4: Opiši vse nize, ki vsebuje vsaj dva niza 'aa' ki se lahko prekrivata v abecedi $\Sigma = \{a, b, c\}$

$$(a + b + c)^* aa(a + b + c)^* aa(a + b + c)^* + (a + b + c)^* aaa(a + b + c)^*$$

Primer 5: Opiši vse nize, ki ne vsebujejo niza 11 v abecedi $\Sigma = \{0, 1\}$

$$(\varepsilon + 1)(0^* 01)^* 0^*$$

$$(\varepsilon + 1)(0^* + 01)^*$$

Primer 6: S slovensko abecedo opisi besedo "Ljubljana" v vseh sklonih in vseh mešanicah velikih in malih črk.

$$(L + l)(J + j)(U + u)(B + b)(L + l)(J + j)(A + a)(N + n)((A + a)(O + o)(E + e)(I + i))$$

Koliko različnih nizov opišemo s tem regularnim izrazom?

$$2^8 \cdot 2^3 = 2^{11} \text{ nizov}$$

2.3 Končni avtomati

2.3.1 Nedeterministični končni avtomati z ε -prehodi

Def.: ε NKA je definiran kot peterka $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, kjer je:

- Q - končna množica stanj
- Σ - vhodna abeceda, $\varepsilon \in \Sigma$
- δ - funkcija prehodov ($\delta : Q \times \Sigma \rightarrow 2^Q$)
- q_0 - začetno stanje
- F - množica končnih stanj

2.3.2 Nedeterministični končni avtomati

Def.: NKA je definiran kot peterka $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, kjer je:

- Q - končna množica stanj
- Σ - vhodna abeceda
- δ - funkcija prehodov ($\delta : Q \times \Sigma \rightarrow 2^Q$)
- q_0 - začetno stanje
- F - množica končnih stanj

2.3.3 Deterministični končni avtomat

Def.: DKA je definiran kot petorka $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, kjer je:

- Q - končna množica stanj
- Σ - vhodna abeceda
- δ - funkcija prehodov ($\delta : Q \times \Sigma \rightarrow Q$)
- q_0 - začetno stanje
- F - množica končnih stanj

2.3.4 Jeziki končnih avtomatov

Def.: Jezik ε NKA ter NKA je definiran kot:

$$L = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

kjer je $\hat{\delta}(q, w)$ posplošena funkcija prehodov v večih korakov.

Def.: Jezik DKA je definiran kot:

$$L = \{w \mid \delta(q_0, w) \in F\}$$

Definicije želijo povedati, da so v jeziku točno tiste besede, po katerih je iz začetnega stanja mogoče priti do nekega končnega stanja.

2.3.5 Regularne gramatike

Def.: Regularna gramatika je definirana kot četvorček $G = \langle V, T, P, S \rangle$, kjer je:

- V - množica spremenljivk oz. vmesnih simbolov, $V \subseteq \Sigma$
- T - množica znakov oz. končnih simbolov, $T \subset \Sigma$
- P - množica produkcij, $[\alpha_1 \rightarrow \alpha_2]$
- S - začetni simbol, $S \in V$

Pri tem pa regularne gramatike ločimo na levo in desno-regularne.

- Pri levih so produkcije $P \subset V \times ((V \cup \{\varepsilon\})^*)$
- Pri desnih so produkcije $P \subset V \times (T^* \cdot (V \cup \{\varepsilon\}))$

To pomeni, da imamo pri levo-regularnih gramatikah vmesne simbole lahko le na skrajni levi, pri desno-regularnih pa le na desni.

2.4 Prevedba med izvedbami regularnih jezikov

Regularni izrazi, regularne gramatike in končni avtomati so vsi enako močni in je mogoče pretvarjati med njimi.

2.4.1 Končni avtomat \rightarrow Regularni izraz

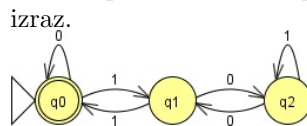
Končni avtomat v regularni izraz prevedemo po metodi z eliminacijo. Pri tej metodi izberemo neko vozlišče za eliminacijo, nato pa njegove sosedne povežemo med seboj, tako, da na nove povezave zapišemo regularne izraze, ki opisujejo dogajanje v tistem vozlišču. Eliminacijo ponavljamo, dokler nam ne ostane le dve stanji, nato pa za končni zapis uporabimo naslednji recept:

$$(R + SQ^*T)^*SQ^*$$

2.5 Primeri izvedb regularnih jezikov

Primeri:

Primer 1: Kako zapišemo DKA za preverjanje deljivosti s 3 v binarnem sistemu? Zapiši še regularni



Regularni izraz:

$$(0 + 1(01^*0)^*1)^*$$

2.6 Ohranjanje regularnosti jezikov

Regularnost jezika ohranjajo operacije:

- Unija \cup
- Stik \circ
- Iteracija (po Def.) $*$
- Presek \cap
- Komplement C
- Obrat oz. reverz R

Omenimo še nekaj sestavljenih operacij (vse izmed njih ohranjajo regularnost):

- Razlika $= \cap^C$
- Ekskluzivni ali $\vee =$

Poglavje 3

Slovar

- Razred - razred je množica elementov, ki ga lahko podamo z naštevanjem elementov ali z opisom lastnosti (opisni ali konceptualni razredi)