Formalni jezici, automati i jezički procesori

školska 2018/2019

Organizacija predmeta

- Predavanja i vežbe (teorijske i praktične) - 4 poena
- Kolokvijumi
  - I kolokvijum – 16 poena teorijski deo + 6 poena praktični deo (Lex)
  - II kolokvijum – 12 poena teorijski deo + 12 poena praktični deo (Yacc)
  - Student može izaći na završni ispit ako na predispitnim obavezama osvoji najmanje **26 poena** i to najmanje **8** na praktičnom delu i najmanje **14** na teorijskom delu.
  - Na svakom kolokvijumu student mora imati više od 4 poena.
- Završni ispit - 50 poena

Razvoj teorije automata

- Računarske nauke počivaju na nekoliko fundamentalnih pitanja:
  - *Šta je algoritam?*
  - *Šta je moguće izračunati, a šta ne?*
  - *Kada algoritam možemo smatrati praktično upotrebljivim?*
- Više od 80 godina naučnici iz oblasti računarstva se bave ovim pitanjima.
- 30tih godina prošlog veka Alan Tjuring je proučavao abstraktne mašine koje su, posmatrano sa stanovišta problema koje bi mogle da izračunaju, imale iste mogućnosti kao savremeni računari.
- Tjuring je imao za cilj da jasno definiše granicu izmedju problema koje mašine mogu da reše i onih koje ne mogu.
- Njegovi zaključi su primenljivi ne samo na abstraktne Tjuringove mašine već i na savremene računare.

Razvoj teorije automata

- Teorija automata se bavi izučavanjem abstraktnih "mašina" za izračunavanje.
- Teorija izračunljivosti se bazira na svega nekoliko elementarnih i diskretnih koncepata, poput *konačni skupovi* i *nizovi*, od se kojih se gradi pojam - *računar*.
- Razvoju teorije izračunljivosti, osim matematičara, doprinose i druge oblasti, poput biologije i lingvistike.
- 40tih i 50tih godina XX veka su proučavane jednostavne mašine – konačni automati.
- Klini je 1956 godine je pokazao ekvivalenciju modela mreže neurona i konačnih automata.
- Trenutno najpopularnija njihova upotreba u tekst editorima i pri leksičkoj analizi teksta.

Razvoj teorije automata

- 50tih godina XX veka lingvist Noam Čomski je počeo da proučava formalne gramatike.
- 1956 godine Čomski je dao matematički model za opis prirodnih jezika koristeći stabla.
- Formalne gramatike su u bliskoj vezi sa automatima i kao takve postale su osnov za razvoj veštačkih jezika i time posale deo kompajlera programskih jezika.
- Kuk je 1969 godine, proširujući Tjuringova istraživanja, razdvojio probleme na
  - one koje mogu uz pomoću računara biti efikasno rešeni i
  - one koji u principu mogu biti rešeni, ali u praksi zahtevaju previše vremena čak i za male primere posmatranog problema – NP-teški problemi.
- Vrlo je verovatno da čak ni eksponencijalni poboljšanja u brzini rada računara neće uticati na našu mogućnost da rešimo velike primere iz ove klase problema.

Zašto proučavati teoriju

- Pojmovi poput konačnih automata i formalnih gramatika se koriste u razvoju softvera
- Tjuringove mašine daju bolje razumevanje šta možemo da očekujemo od softvera
- Potrebno je prepoznati kojoj klasi problema pripada i
  - ako je rešiv napisati softver koji će ga rešiti ili
  - u slučaju NP-problema naći približno rešenje ili iskoristiti neku heuristiku ili nekim drugim metodom smanjiti vreme potrebno za rešavanje problema
- Primeri NP-teških problema
  - Problem trgovačkog putnika
  - Ispitivanje zadovoljivosti formula
  - Pronalaženje faktora broj od 500 cifara
  - problem semafora, rasporeda i slično

Zašto proučavati teoriju

- Konačni automati se mogu koristiti za:
  - softver za dizajn i proveru ponašanja digitalnih kola
  - leksički analizator kompajlera
  - softver koji u velikom tekstu, poput kolekcije veb strana, pronalazi ključne reči, fraze i druge šablone
  - softver za proveru sistema koji imaju konačan niz stanja, poput komunikacionih protokola ili protokola za bezbednu razmenu informacija
- S obzirom na trend razvoja računarske nauke, specifično tehničko znanje postaje neupotrebljivo posle nekoliko godina.
- Poznavanje teorijskih osnova koje se, u ovom slučaju, tiču teorije automata, problema izračunljivosti i kompleksnosti, omugućava lako usvajanje novih tehnologija i aktivno učešće u razvoju istih.

Alfabet

- Računari rade sa tekstovima koje možemo posmatrati kao nizove simbola nad zadatim alfabetom.
- Programi su tekstovi nad alfabetom tastature, ulaz i izlaz su, takođe, tekstovi nad istim alfabetom.
- Polazeći od osnovnih pojmova alfabet, reč, jezik cilj nam je da dođemo do definicije pojmova algoritam, računar, izračunavanje, itd.

Alfabet

Дефиниција

Alfabet (azbuka) Σ *je neki konačan, neprazan skup elemenata. Elementi skupa se nazivaju simboli.*

Primer. Sledeći alfabeti će biti često korišćeni:

- $\Sigma_U = \{|\}$ je tzv. *Unarni alfabet* koji sadrži samo jedan simbol – vertikalnu crtu;
- $\Sigma_{\text{bool}} = \{0, 1\}$ je *Bulov alfabet*;
- $\Sigma_m = \{0, 1, \ldots, m-1\}$, za $m \geqslant 1$, je alfabet za zapisivanje prirodnih brojeva u bazi m ($\Sigma_2 = \Sigma_{\text{bool}}$);
- $\Sigma_{\text{lat}} = \{a, b, c, \ldots, z\}$ latinica;
- $\Sigma_{\text{keyboard}} = \{\mathtt{A}, \mathtt{a}, \mathtt{B}, \mathtt{b}, \ldots, \mathtt{Z}, \mathtt{z}, \sqcup, >, <, (,), \ldots, !\}$ simboli tastature, $\sqcup$ označava blanko znak;
- $\Sigma_{\text{logic}} = \{0, 1, p, (,), \wedge, \vee, \neg\}$ jezik iskazne logike.

Šta je reč?

Definicija

Reč nad alfabetom Σ je svaki konačan niz simbola iz Σ. Prazna reč, u oznaci e (λ, ε), je reč koja ne sadrži ni jedan simbol. dužina reči w, u oznaci $|w|$, je broj simbola u reči w.

- Ako je Σ alfabet, za $m \geqslant 2$, skup $\Sigma^m = \underbrace{\Sigma \times \cdots \times \Sigma}_{m \text{ puta}}$, skup svih m-torki simbola nazivamo skupom svih reči nad Σ sužine m, i umesti $(a_1, \cdots, a_m)$ pišemo $a_1 \cdots a_m$
- Skup Σ^1 identifikujemo sa skupom Σ.
- Skup Σ^0 je jednočlani skup čiji je jedini elemnt ε.
- $\Sigma^+ = \bigcup_{m \geqslant 1} \Sigma^m, \qquad \Sigma^* = \bigcup_{m \geqslant 0} \Sigma^m = \Sigma^+ \cup \{\varepsilon\}.$
- $|\Sigma^m| = |\Sigma|^m$.

Konkatenacija reči

Дефиниција

Neka je Σ bilo koji alfabet i neka su $u, v \in \Sigma^*$. Reč dobijena konkatenacijom (dopisivanjem) reči u i v, čije su dužine m i n respektivno, u oznaci uv ili $u \cdot v$, je reč nad alfabetom Σ dužine $m + n$, takva da je i-to slovo reči uv jednako i-tom slovu reči u ako je $i \leqslant m$, odnosno $i - m$-tom slovu reči v ako je $i > m$.

Prefiks, sufiks, podreč

Дефиниција

Neka je Σ bilo koji alfabet i neka su $u, v \in \Sigma^*$.

1. *u je prefiks reči w ako postoji reč v takva da važi $w = uv$*
2. *u je sufiks reči w ako postoji reč v takva da važi $w = vu$*
3. *u je podreč reči w ako postoje reči v_1 i v_2, maguće i prazne, takve da važi $w = v_1 u v_2$.*

Šta je jezik?

Definicija

Neka je Σ bilo koji alfabet. Jezik L nad Σ je bilo koji podskup skupa Σ^*.

Definicija

Neka je Σ alfabet. Na skupu $\mathcal{P}(\Sigma^*)$ svih jezika nad Σ definišemo sledeće operacije:

1. *skupovne:* $\cap$, $\cup$, $\setminus$, ^{c} *(u odnosu na* Σ^**)*
2. *nadovezivanje (proizvod):* $L_1 \cdot L_2 = \{uv | u \in L_1, v \in L_2\}$
3. *stepen:* $L^0 = \{\varepsilon\}$, $L^{n+1} = L^n \cdot L$
4. *iteracija (Klinijeva zvazdica):*
   $L^* = \bigcup_{n \geqslant 0} L^n = \{u_1 \dots u_n | n \in \mathbf{N}, u_1, \dots, u_n \in L\}$

Problem pripadanja

- Fundamentani problem u vezi sa jezicima jeste problem pripadanja

  Ako je $L \subset \Sigma^*$ naći algoritam koji za ulaz $w \in \Sigma^*$ daje odgovor DA ako $w \in L$, odnosno NE ako $w \notin L$.

- Algoritam treba da oderdi funkciju

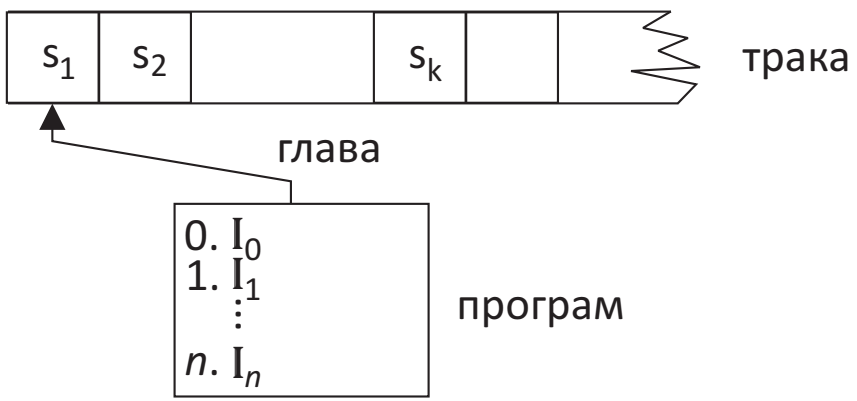
$$A : \Sigma^* \to \{0, 1\}, \qquad A(w) = \begin{cases} 0, & w \notin L \\ 1, & w \in L \end{cases}$$

Regularni jezici

školska 2018/2019

Šta je konačni automat

- Konačni automati reprezentuju najjednostavnije algoritme (programe) kojima se rešavaju problemi pripadanja za pojedine jezike.
- „Mašina" koja zadatu reč ($w = s_1 s_2 s_3 \ldots s_k$) čita simbol po simbol i pri tome izvršava veoma jednostavne instrukcije.



„Instrukcije" konačnog automata

- Neka su reči zadate nad alfabetom $\Sigma_{\text{bool}} = \{0, 1\}$.
- Programi su konačni nizovi instrukcija oblika:

  IF *symbol*=0 THEN goto *i* ELSE goto *j*

  pri čemu *i* i *j* označavaju redne brojeve nekih instrukcija tog programa
- Navedeni opšti oblik instrukcije ekvivalentan sa

  IF *symbol*=1 THEN goto *j* ELSE goto *i*.

„Instrukcije" konačnog automata

- Na početku rada, glava čita sadržaj prve ćelije i izvršava instrukciju $\mathbb{I}_0$.
- Nakon toga, glava se pomera za jedno mesto udesno i izvršava instrukciju na koju je upućena u prethodnom koraku.
- Ponovo se pomera za jedno mesto udesno i izvršava instrukciju na koju je upućena u prethodnom koraku itd.
- Mašina se zaustavlja kada učita sve simbole sa ulaza, tj. kada glava stigne do praznog polja.

Konačni automat nad proizvoljnim alfabetom

- Na potpuno analogan način razmatramo i opšti slučaj kada su na ulazu dozvoljene reči nad bilo kojim alfabetom $\Sigma = \{s_1, s_2, \ldots, s_k\}$.
- Za pisanje programa koristimo naredbe oblika:

```
select   symbol=s_1 goto i_1
         symbol=s_2 goto i_2
         ⋮
         symbol=s_k goto i_k
```

- Naredba `IF symbol=0 THEN goto i ELSE goto j` zapravo samo kraće zapisana naredba:

```
select   symbol=0 goto i
         symbol=1 goto j.
```

Konačni automat nad proizvoljnim alfabetom

- Uopšteno govoreći, konačne automate nad alfabetom $\Sigma = \{s_1, \ldots, s_k\}$ reprezentuju konačni „težinski" usmereni grafovi:
  - čije čvorove nazivamo stanjima, i medju kojima je izdvojeno jedno početno stanje i nekoliko završnih stanja, pri čemu dolazi u obzir i mogućnost da nema završnih stanja, i
  - iz svakog čvora (stanja) polazi tačno k ivica koje su označene simbolima $s_1, \ldots, s_k$.
- Kažemo da neki konačni automat nad Σ prihvata reč $w \in \Sigma^*$ ako se put koji polazi iz početnog stanja i odgovara reči w završava u nekom od završnih stanja.
- Automat prihvata jezik $L \subseteq \Sigma^*$ ako prihvata sve reči iz L, i ne prihvata reči iz $\Sigma^* \setminus L$.

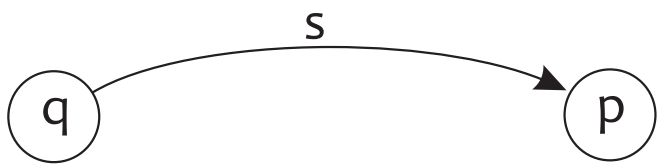
Formalna definicija konačnog automata

Дефиниција

Konačni automat je uredjena petorka $\mathbb{M} = (Q, q_0, F, \Sigma, \delta)$, *gde je*

- Q *konačan skup stanja,*
- $q_0 \in Q$ *početno stanje,*
- $F \subseteq Q$ *skup završnih stanja,*
- Σ *ulazni alfabet,*
- $\delta : Q \times \Sigma \to Q$ *funkcija tranzicije.*

- Jednakost $\delta(q, s) = p$ odgovara strelici iz stanja q u stanje p pri čemu je strelica označena sa s.



Izračunavanje automata

Дефиниција

Neka $\mathbb{M} = (Q, q_0, F, \Sigma, \delta)$ *konačni automat.*

- *Konfiguracija automata* $\mathbb{M}$ *je svaki element iz* $Q \times \Sigma^*$. *Za bilo koje* $q \in Q$ *i* $w \in \Sigma^*$, *konfiguraciju* (q, w) *kraće označavamo* qw. *Specijalno konfiguraciju* $q\varepsilon$ *označavamo* q.
- *Računski korak automata* $\mathbb{M}$ *jeste binarna relacija* $\vdash_{\mathbb{M}}$ *na skupu svih konfiguracija definisana na sledeći način:*

$$qw \vdash_{\mathbb{M}} pv \Leftrightarrow \text{ za neki } s \in \Sigma, w = sv \text{ i } \delta(q, s) = p.$$

- *Izračunavanje automata* $\mathbb{M}$ *jeste svaki konačan niz konfiguracija* C_0, $C_1, \ldots, C_n$, $n \geqslant 1$, *takav da je* $C_i \vdash_{\mathbb{M}} C_{i+1}$, *za svaki* i, $0 \leqslant i < n$.
- *Izračunavanje automata* $\mathbb{M}$ *za ulaz* w *jeste svako izračunavanje* C_0, $C_1, \ldots, C_n$ *takvo da je* $C_0 = q_0 w$ *i* $C_n \in Q \times \{\varepsilon\}$. *Tada se konfiguracije* C_0 *naziva početna konfiguracija, a* C_n *završna konfiguracija.*

Relacija $\vdash_{\mathbb{M}}^{*}$

- Pojam izračunavanja automata $\mathbb{M}$ koristimo da bismo uveli refleksivno i tranzitivno zatvorenje relacije $\vdash_{\mathbb{M}}$, tj, relaciju $\vdash_{\mathbb{M}}^{*}$ medju konfiguracijama definisanu sa: $qw \vdash_{\mathbb{M}}^{*} pv$ akko
  - važi $q = p$ i $w = v$, ili
  - postoji izračunavanje $C_0, C_1, \ldots, C_n$ takvo da je $C_0 = qw$ i $C_n = pv$.

Prihvatanje reči

Дефиниција

Neka $\mathbb{M} = (Q, q_0, F, \Sigma, \delta)$ *konačni automat.*

- *Automat* $\mathbb{M}$ *prihvata reč* w*, ako stanje završne konfiguracije izračunavanja automata* $\mathbb{M}$ *za ulaz* w *pripada skupu (završnih stanja)* F*. U suprotnom, automat* $\mathbb{M}$ *ne prihvata reč* w*.*
- *Jezik koji prihvata automat* $\mathbb{M}$ *jeste skup svih reči nad* Σ *koje prihvata automat* $\mathbb{M}$*:*

$$L(\mathbb{M}) = \{w \in \Sigma^* \mid \mathbb{M} \text{ prihvata reč } w\}.$$

- Jezik koji prihvata automat $\mathbb{M}$ možemo opisati i relacijom $\vdash_{\mathbb{M}}^*$:

$$L(\mathbb{M}) = \{w \in \Sigma^* \mid q_0 w \vdash_{\mathbb{M}}^* p \text{ za neko } p \in F\}.$$

Funkcija $\hat{\delta}$

Дефиниција

Neka je $(Q, q_0, F, \Sigma, \delta)$ *neki konačni automat. Zatvorenje funkcije* $\delta : Q \times \Sigma \to Q$ *jeste funkcija* $\hat{\delta} : Q \times \Sigma^* \to Q$ *data sledećim jednakostima:*

- $\hat{\delta}(q, \varepsilon) = q,\ q \in Q,$
- $\hat{\delta}(q, ws) = \delta(\hat{\delta}(q, w), s),\ q \in Q,\ w \in \Sigma^*,\ s \in \Sigma.$

- Jezik koji prihvata automat $\mathbb{M}$ ima i sledeći opis:

$$L(\mathbb{M}) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}.$$

Lema

Neka je $\mathbb{M} = (Q, q_0, F, \Sigma, \delta)$ *konačni automat. Ako su* $u, v \in \Sigma^*$ *reči takve da je* $\hat{\delta}(q_0, u) = \hat{\delta}(q_0, v)$*, tada za svaku reč* $w \in \Sigma^*$ *važi:*

$$uw \in L(\mathbb{M}) \Leftrightarrow vw \in L(\mathbb{M}).$$