Prevođenje programskih jezika – beleške sa predavanja
Potisni automati

Milan Banković

*Matematički fakultet,
Univerzitet u Beogradu

Jesenji semestar 2023/24.

Pregled

1 Definicija potisnog automata

2 Potisni automati i kontekstno slobodni jezici

3 Prošireni potisni automati

4 Potisni automati i preduvidni simboli

Motivacija

Ograničenja konačnih automata

- Jedina memorija konačnog automata je njegovo stanje
- Stanja ima konačno mnogo, tako da su mogućnosti pamćenja ograničene
- Na primer, konačni automati ne mogu da „broje" neograničeno:
  - zato nismo mogli da prepoznamo jezik $L = \{a^n b^n \mid n \geq 0\}$
  - bilo je potrebno da zapamtimo koliko smo simbola a do sada pročitali...
  - ...a da nakon toga utvrdimo da li je broj simbola b isti
- Šta ukoliko bismo, dodatno, imali na raspolaganju i stek?

Definicija

Definicija 1

Potisni automat je uređena sedmorka $\mathcal{A} = (\Sigma, Q, \Gamma, q_0, Z_0, K, \delta)$, *gde je:*

- Σ *konačna azbuka*
- Q *konačni skup stanja automata*
- Γ *konačna azbuka simbola steka (ne obavezno disjunktna sa* Σ*)*
- $q_0 \in Q$ *početno stanje automata*
- $Z_0 \in \Gamma$ *početni simbol na steku*
- K *skup završnih konfiguracija*
- δ *funkcija prelaska koja svakoj trojci iz* $Q \times \Sigma \cup \{\varepsilon\} \times \Gamma$ *pridružuje konačni skup parova iz* $Q \times \Gamma^*$

Definicija

Definicija 2

Pod konfiguracijom potisnog automata podrazumevamo uređenu trojku oblika (q, w, γ), *gde je:*

- $q \in Q$ *trenutno stanje automata*
- $w \in \Sigma^*$ *je reč koja sledi na ulazu*
- *a* $\gamma \in \Gamma^*$ *je trenutni sadržaj steka*

Pritom, smatramo da je početni simbol niske γ *na vrhu steka, a njen krajnji simbol na dnu steka. Među konfiguracijama automata definišemo relaciju izračunavanja automata* $\vdash$ *na sledeći način:*

$$(q, aw, Z\gamma) \vdash (q', w, \gamma_1\gamma) \text{ akko } (q', \gamma_1) \in \delta(q, a, Z).$$

pri čemu je $a \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ *i* $\gamma, \gamma_1 \in \Gamma^*$. *Tranzitivno i refleksivno zatvorenje relacije* $\vdash$ *označavaćemo sa* $\vdash^*$.

Definicija

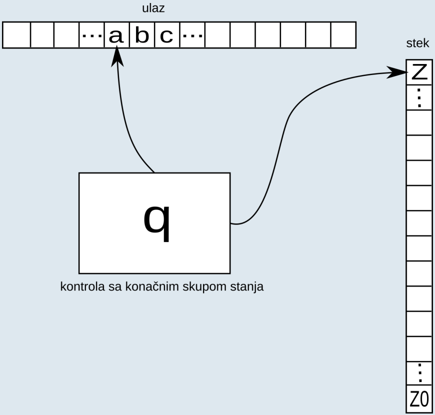
Definicija 3

Jezik automata $\mathcal{A}$ je jezik
$L(\mathcal{A}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \gamma) \ \wedge \ (q, \varepsilon, \gamma) \in K\}$.
Pritom, skup završnih konfiguracija K može biti jednog od sledećih oblika:

- $K = \{(q, w, \varepsilon) \mid q \in Q \wedge w \in \Sigma^*\}$*: za ovakve automate kažemo da prepoznaju jezik praznim stekom*
- $K = \{(q, w, \gamma) \mid q \in F \wedge F \subseteq Q \wedge w \in \Sigma^* \wedge \gamma \in \Gamma^*\}$*: za ovakve automate kažemo da prepoznaju jezik završnim stanjem (to je neki neprazan podskup $F \subseteq Q$).*

Ilustracija

Ilustracija potisnog automata



Napomene

Napomene

- Ako važi $(q_1, aw_1, Z\gamma_1) \vdash (q_2, w_1, \gamma\gamma_1)$, tada važi i $(q_1, aw_2, Z\gamma_2) \vdash (q_2, w_2, \gamma\gamma_2)$
  - relacija $\vdash$ zavisi samo od sledećeg simbola na ulazu i simbola na vrhu steka
- Ako važi $(q_1, w, \gamma_1) \vdash^* (q_2, \varepsilon, \gamma_2)$, tada važi i $(q_1, wu, \gamma_1\gamma) \vdash^* (q_2, u, \gamma_2\gamma)$
  - simboli koji slede kasnije na ulazu i simboli u dubini steka ne utiču na izračunavanje automata dok se do njih ne dođe
- Ako je $(q_2, \gamma_2) \in \delta(q_1, \varepsilon, Z)$, tada važi $(q_1, w, Z\gamma) \vdash (q_2, w, \gamma_2\gamma)$; ovaj prelaz nazivamo ε-prelaz
- Ako je $(q_2, \varepsilon) \in \delta(q_1, a, Z)$, tada važi $(q_1, aw, Z\gamma) \vdash (q_2, w, \gamma)$
  - efektivno, simbol Z se samo uklanja sa vrha steka
- Ako je $(q_2, Z) \in \delta(q_1, a, Z)$, tada važi $(q_1, aw, Z\gamma) \vdash (q_2, w, Z\gamma)$
  - stek ostaje nepromenjen
- Ako je $(q_2, Z_1 Z) \in \delta(q_1, a, Z)$, tada važi $(q_1, aw, Z\gamma) \vdash (q_2, w, Z_1 Z\gamma)$
  - na stek se dodaje simbol Z_1

Primer

Primer

Neka je dat jezik $L = \{a^n b^n \mid n \geq 0\}$. *Setimo se gramatike* $S \to aSb \mid \varepsilon$ *koja je generisala upravo ovaj jezik. Možemo napraviti potisni automat koji simulira izvođenja u ovoj gramatici. Neka je* $Q = \{q_0\}$, $\Gamma = \{S, a, b\}$, *i neka je* S *početni simbol steka. Automat će prepoznavati jezik praznim stekom, a funkcija prelaska je:*

$$\begin{array}{rcl} \delta(q_0, \varepsilon, S) & = & \{(q_0, aSb), (q_0, \varepsilon)\} \\ \delta(q_0, a, a) & = & \{(q_0, \varepsilon)\} \\ \delta(q_0, b, b) & = & \{(q_0, \varepsilon)\} \end{array}$$

Sada izvođenju $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$ *u gramatici odgovara izračunavanje u automatu*
$(q_0, aabb, S) \vdash (q_0, aabb, aSb) \vdash (q_0, abb, Sb) \vdash (q_0, abb, aSbb) \vdash (q_0, bb, Sbb) \vdash (q_0, bb, bb) \vdash (q_0, b, b) \vdash (q_0, \varepsilon, \varepsilon)$.

Primer

Primer

Neka je data gramatika $A \to aBA \mid b, B \to bAB \mid a$*. Jezik generisan ovom gramatikom prepoznaje (praznim stekom) potisni automat kod koga je* $Q = \{q_0\}$, $\Gamma = \{a, b, A, B\}$*, početni simbol steka je* A*, a funkcija prelaska je:*

$$
\begin{array}{rcl}
\delta(q_0, \varepsilon, A) & = & \{(q_0, aBA), (q_0, b)\} \\
\delta(q_0, \varepsilon, B) & = & \{(q_0, bAB), (q_0, a)\} \\
\delta(q_0, a, a) & = & \{(q_0, \varepsilon)\} \\
\delta(q_0, b, b) & = & \{(q_0, \varepsilon)\}
\end{array}
$$

Na primer, izvođenju u gramatici $A \Rightarrow aBA \Rightarrow abABA \Rightarrow abbBA \Rightarrow abbaA \Rightarrow abbab$ *odgovara izračunavanje automata:* $(q_0, abbab, A) \vdash (q_0, abbab, aBA) \vdash (q_0, bbab, BA) \vdash (q_0, bbab, bABA) \vdash (q_0, bab, ABA) \vdash (q_0, bab, bBA) \vdash (q_0, ab, BA) \vdash (q_0, ab, aA) \vdash (q_0, b, A) \vdash (q_0, b, b) \vdash (q_0, \varepsilon, \varepsilon)$.

Pregled

1 Definicija potisnog automata

2 Potisni automati i kontekstno slobodni jezici

3 Prošireni potisni automati

4 Potisni automati i preduvidni simboli

Prosti potisni automati

Definicija 4

Potisni automat koji prepoznaje jezik praznim stekom i ima samo jedno stanje naziva se prost potisni automat.

Napomena

Prosti potisni automati efektivno ne koriste stanja, već im je stek jedina memorija.

Teorema 1

Za svaki kontekstno slobodni jezik postoji prost potisni automat koji ga prepoznaje.

Dokaz

Neka je kontekstno slobodni jezik L *generisan gramatikom* $G = (\Sigma, N, S, P)$*. Neka je* $\mathcal{A}$ *automat koji ima samo jedno stanje* q_0*, neka je* $\Gamma = \Sigma \cup N$ *i neka je* S *početni simbol steka. Imamo dve vrste prelaza:*

- *Za svako pravilo gramatike* $A \to \gamma$ *imamo* $(q_0, \gamma) \in \delta(q_0, \varepsilon, A)$ *[prelazi ekspanzije]*
- *Za svaki simbol* $a \in \Sigma$ *imamo* $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$ *[prelazi sinhronizacije]*

Lako se može videti da opisani automat prepoznaje jezik L *praznim stekom. Ovakav automat nazivamo standardni automat za gramatiku* G*.*

Potisni automati i kontekstno slobodni jezici

Teorema 2

Za svaki potisni automat postoji kontekstno slobodna gramatika koja generiše jezik prepoznat tim automatom.

Dokaz

Videti knjigu profesora Vitasa.

Posledica

Jezik je kontekstno slobodan akko ga je moguće prepoznati nekim potisnim automatom.

Deterministički potisni automati

Da li je izračunavanje prethodnih automata jednoznačno određeno?

- Automati konstruisani na ranije opisani način efektivno simuliraju izvođenja nalevo u gramatici
- Kako možemo imati više pravila za isti nezavršni simbol, dobijeni automat u opštem slučaju nije deterministički

Definicija 5

Za potisni automat kažemo da je deterministički ako važe sledeći uslovi:

- *za svako $q \in Q$, $a \in \Sigma$ i $Z \in \Gamma$ postoji najviše jedan element u skupu $\delta(q, a, Z)$*
- *ako je skup $\delta(q, \varepsilon, Z)$ neprazan za neko $q \in Q$ i $Z \in \Gamma$, tada on mora biti jednočlan i istovremeno svi skupovi $\delta(q, a, Z)$ moraju biti prazni (za svako $a \in \Sigma$)*

Napomena

Navedeni uslovi garantuju da za svaku reč $w \in \Sigma^*$ postoji najviše jedno izračunavanje u automatu.

Primer

Primer

S obzirom da je standardni automat za datu gramatiku po pravilu nedeterministički, možemo pokušati da nekim ad-hoc pristupom formiramo neki drugačiji automat koji prepoznaje isti jezik, a zadovoljava svojstva determinizma. Na primer, za gramatiku $A \rightarrow aBA \mid b, B \rightarrow bAB \mid a$ *iz prethodnog primera možemo konstruisati automat na sledeći način:* $Q = \{q_0\}$, $\Gamma = \{A, B\}$, *početni simbol steka je* A, *a funkcija prelaska je definisana ovako:*

$$\begin{array}{rcl} \delta(q_0, a, A) & = & \{(q_0, BA)\} \\ \delta(q_0, b, A) & = & \{(q_0, \varepsilon)\} \\ \delta(q_0, a, B) & = & \{(q_0, \varepsilon)\} \\ \delta(q_0, b, B) & = & \{(q_0, AB)\} \end{array}$$

Ovaj automat je deterministički i prepoznaje isti jezik kao i prethodni. Na primer, izvođenju u gramatici $A \Rightarrow aBA \Rightarrow abABA \Rightarrow abbBA \Rightarrow abbaA \Rightarrow abbab$ *odgovara izračunavanje automata:* $(q_0, abbab, A) \vdash (q_0, bbab, BA) \vdash (q_0, bab, ABA) \vdash (q_0, ab, BA) \vdash (q_0, b, A) \vdash (q_0, \varepsilon, \varepsilon)$.

Napomena

Na žalost, ovaj pristup nije univerzalni i ne može se primeniti na bilo koju gramatiku. Na primer, nije ga moguće primeniti na gramatiku $S \rightarrow aSb \mid \varepsilon$.

Primer

Primer

Drugi način razmišljanja je da probamo da formiramo automat koji prepoznaje završnim stanjem*. Pretpostavimo da opet imamo gramatiku $S \to aSb \mid \varepsilon$. Formiramo automat sa skupom stanja $Q = \{q_0, q_a, q_b, q_F\}$ (gde je q_0 početno stanje, a $\{q_0, q_F\}$ skup završnih stanja), azbukom steka $\Gamma = \{Z, a\}$, početnim simbolom steka Z i funkcijom prelaska:*

$$\begin{aligned}
\delta(q_0, a, Z) &= \{(q_a, aZ)\} \\
\delta(q_a, a, a) &= \{(q_a, aa)\} \\
\delta(q_a, b, a) &= \{(q_b, \varepsilon)\} \\
\delta(q_b, b, a) &= \{(q_b, \varepsilon)\} \\
\delta(q_b, \varepsilon, Z) &= \{(q_F, Z)\}
\end{aligned}$$

Ovaj deterministički automat prepoznaje jezik $L = \{a^n b^n \mid n \geq 0\}$ završnim stanjem (q_0 je završno, kako bi i prazna niska bila prepoznata). Na primer, izračunavanje koje prepoznaje nisku aabb izgleda ovako:
$(q_0, aabb, Z) \vdash (q_a, abb, aZ) \vdash (q_a, bb, aaZ) \vdash (q_b, b, aZ) \vdash (q_b, \varepsilon, Z) \vdash (q_F, \varepsilon, Z)$.

Deterministički kontekstno slobodni jezici

Da li svakom nedeterminističkom potisnom automatu odgovara neki deterministički automat?

- Odgovor na ovo pitanje je: NE
- Ovo znači da postoje kontekstno slobodni jezici koji se ne mogu prepoznati determinističkim potisnim automatom

Definicija 6

Za kontekstno slobodni jezik kažemo da je deterministički ako postoji deterministički potisni automat koji ga prepoznaje. U suprotnom je nedeterministički.

Primer

Neka je dat jezik L opisan gramatikom $S \to aSa \mid bSb \mid \varepsilon$ *(jezik svih palindroma parne dužine nad azbukom* $\{a, b\}$*). Može se pokazati da je ovaj jezik nedeterministički. Intuitivno, automat bi trebalo da do pola ulazne niske prebacuje simbole sa ulaza na stek, a da nakon toga skida simbole sa steka i upoređuje ih sa preostalim simbolima na ulazu. Nevolja je u tome što ne možemo da znamo kada smo „na pola”. Za formalni dokaz ove činjenice, pogledati knjigu profesora Vitasa.*

Prazan stek vs. završna stanja

Teorema 3

Potisni automat koji prepoznaje jezik L praznim stekom postoji akko postoji potisni automat koji prepoznaje jezik L završnim stanjem.

Dokaz

Neka je automat $\mathcal{A}$ takav da prepoznaje jezik L praznim stekom. Tada dodajemo novi početni simbol $\perp$ u skup Γ (tzv. marker dna steka), kao i novo početno stanje q_p. Zatim dodajemo i prelaz $\{(q_0, Z_0\perp)\} \in \delta(q_p, \varepsilon, \perp)$, gde su Z_0 i q_0 početni simbol steka i početno stanje polaznog automata, respektivno. Najzad, dodajemo prelaz $(q_F, \perp) \in \delta(q, \varepsilon, \perp)$, za svako stanje q polaznog automata, pri čemu je q_F novododato završno stanje. Ovaj automat prepoznaje isti jezik završnim stanjem q_F. Primetimo da ovaj postupak čuva determinizam polaznog automata.

Obratno, neka je $\mathcal{A}$ automat koji prepoznaje neki jezik L skupom završnih stanja. Slično kao i malopre, dodajemo marker dna steka $\perp$ u skup Γ, novo početno stanje q_p, kao i prelaz $\{(q_0, Z_0\perp)\} \in \delta(q_p, \varepsilon, \perp)$. Za svako završno stanje q polaznog automata i za svaki simbol $Z \in \Gamma \cup \{\perp\}$ dodajemo prelaz $(q_F, \varepsilon) \in \delta(q, \varepsilon, Z)$, gde je q_F novododato stanje, kao i prelaz $(q_F, \varepsilon) \in \delta(q_F, \varepsilon, Z)$ za svaki simbol $Z \in \Gamma \cup \{\perp\}$. Ovim prelazima je omogućeno da kad god se nađemo u završnom stanju, mi možemo da ,,očistimo'' stek, bez dodatnog čitanja sa ulaza. Ovaj postupak ne čuva determinističnost u opštem slučaju.

Prefiksni kontekstno-slobodni jezici

Zbog čega su nam potrebna stanja?

- Prost potisni automat – ima samo jedno stanje i prepoznaje jezik praznim stekom
  - prosti potisni automati mogu prepoznati bilo koji kontekstno slobodno jezik (nedeterministički)
- Opšti potisni automat – može imati više stanja
  - može prepoznavati jezik bilo praznim stekom, bilo završnim stanjem
  - ne može prepoznati ništa više od kontekstno slobodnih jezika
- Zbog čega smo uopšte i uvodili ovu opštiju definiciju potisnog automata?
  - ispostavi se da je klasa determinističkih jezika koji se mogu prepoznati praznim stekom znatno uža od klase svih determinističkih jezika
  - setimo se da onaj postupak za transformaciju automata koji prepoznaje završnim stanjem u automat koji prepoznaje praznim stekom nije čuvao determinizam

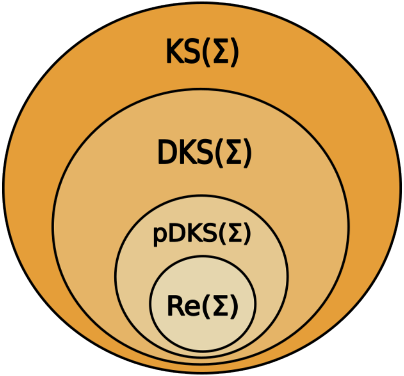
Definicija 7

Za jezik L kažemo da je prefiksni ako ni za jednu reč $w \in L$ ne postoji njen prefiks koji je takođe u L.

Napomena

Može se pokazati da deterministički potisni automati koji prepoznaju praznim stekom mogu prepoznati samo prefiksne jezike. Jezik $L = \{a^n b^n \mid n \geq 0\}$ nije takav, jer je $\varepsilon \in L$, a ε je prefiks svake reči. Otuda se ovaj jezik nije mogao deterministički prepoznati praznim stekom (ali jeste završnim stanjem, vidi slajd 16).

Odnos klasa jezika nad Σ



Jezici nad Σ

- $KS(\Sigma)$ - kontekstno-slobodni (KS) jezici
- $DKS(\Sigma)$ - deterministički KS jezici
- $pDKS(\Sigma)$ - prefiksni deterministički KS jezici (prepoznati praznim stekom)
- $Re(\Sigma)$ - regularni jezici

Pregled

1 Definicija potisnog automata

2 Potisni automati i kontekstno slobodni jezici

3 Prošireni potisni automati

4 Potisni automati i preduvidni simboli

Prošireni potisni automati

Definicija 8

Prošireni potisni automati su potisni automati kod kojih je dozvoljeno da se u svakom koraku sa vrha steka skida nula ili više simbola (umesto tačno jednog). Formalno, funkcija prelaska δ *predstavlja konačnu parcijalnu funkciju*[1] *koja trojkama iz* $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*$ *pridružuje konačne podskupove skupa* $Q \times \Gamma^*$.

[1] Ovo znači da imamo samo konačno mnogo trojki iz $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*$ za koje postoje prelazi

Primer

Primer

Setimo se ponovo gramatike $S \to aSb \mid \varepsilon$. Posmatrajmo automat koji ima samo jedno stanje q_0, a azbuka steka je $\Gamma = \{a, S, b, \perp\}$, pri čemu je $\perp$ početni simbol steka. Neka je funkcija prelaska definisana na sledeći način:

$$
\begin{array}{rcl}
\delta(q_0, a, \varepsilon) & = & \{(q_0, a)\} \\
\delta(q_0, b, \varepsilon) & = & \{(q_0, b)\} \\
\delta(q_0, \varepsilon, aSb) & = & \{(q_0, S)\} \\
\delta(q_0, \varepsilon, \varepsilon) & = & \{(q_0, S)\} \\
\delta(q_0, \varepsilon, \perp S) & = & \{(q_0, \varepsilon)\}
\end{array}
$$

Ovaj prošireni automat prepoznaje jezik praznim stekom. Prva dva prelaza nazivaju se prelazima prebacivanja – njima se simboli sa ulaza prebacuju na stek. Druga dva prelaza predstavljaju prelaze redukcije – njima se ,,unazad" primenjuju pravila, tj. desne strane sa vrha steka se zamenjuju levim stranama. Ovim automatom se, zapravo, izvođenje nadesno u gramatici simulira unazad. Kada se stigne do početnog simbola gramatike, primenjuje se poslednji prelaz, kojim se dobija prazan stek. Na primer, izvođenju u gramatici $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$ odgovara izračunavanje u automatu $(q_0, aabb, \perp) \vdash (q_0, abb, \perp a) \vdash (q_0, bb, \perp aa) \vdash (q_0, bb, \perp aaS) \vdash (q_0, b, \perp aaSb) \vdash (q_0, b, \perp aS) \vdash (q_0, \varepsilon, \perp aSb) \vdash (q_0, \varepsilon, \perp S) \vdash (q_0, \varepsilon, \varepsilon)$.

Prošireni potisni automati

Teorema 4

Za svaku kontekstno-slobodnu gramatiku G postoji prošireni potisni automat koji prepoznaje jezik $L(G)$.

Dokaz

Neka je $G = (\Sigma, N, S, P)$. Azbuka steka proširenog potisnog automata biće $\Gamma = \Sigma \cup N \cup \{\perp\}$, pri čemu je $\perp$ početni simbol steka. Jedino stanje automata biće q_0, a imaćemo sledeće prelaze:

- *Za svaki simbol $a \in \Sigma$ imaćemo prelaz $(q_0, a) \in \delta(q_0, a, \varepsilon)$ [prelaz prebacivanja]*
- *Za svako pravilo $A \to \gamma$ imaćemo prelaz $(q_0, A) \in \delta(q_0, \varepsilon, \gamma)$ [prelaz redukcije]*
- *Postoji prelaz $(q_0, \varepsilon) \in \delta(q_0, \varepsilon, \perp S)$*

Ovaj automat (nedeterministički, u opštem slučaju) prepoznaje jezik $L(G)$, praznim stekom. Nazivamo ga i standardni prošireni automat za gramatiku G.

Napomena

Kod proširenih automata, po konvenciji, kada zapisujemo da je stanje steka niska $\gamma \in \Gamma^*$, podrazumevamo da je vrh steka poslednji simbol niske γ, za razliku od običnih, gde smo podrazumevali da je na vrhu steka početni simbol niske γ.

Prošireni potisni automati

Teorema 5

Za svaki prošireni potisni automat $\mathcal{A}$ postoji običan potisni automat $\mathcal{A}'$ koji prepoznaje isti jezik.

Dokaz

Dokaz pogledati u Aho-Ulman-u. Ideja dokaza je da se uvode dodatni simboli steka, kao i dodatna stanja automata kojima se može simulirati rad proširenog potisnog automata običnim automatom.

Napomena

Prošireni potisni automati i dalje prepoznaju samo kontekstno slobodne jezike. Suština je u nešto drugačijem načinu rada kojima se omogućava tzv. parsiranje „naviše" (dok obični potisni automati u svom standardnom obliku omogućavaju parsiranje „naniže"). Više o ovim osnovnim tehnikama parsiranja u nastavku ovih predavanja.

Pregled

1 Definicija potisnog automata

2 Potisni automati i kontekstno slobodni jezici

3 Prošireni potisni automati

4 Potisni automati i preduvidni simboli

Preduvidni simboli

Nedeterminizam i kako ga se rešiti?

- Nedeteminizam kod potisnih automata je najčešće uzrokovan postojanjem ε-prelaza:
  - za isto stanje i isti simbol na vrhu steka imamo i ε-prelaz i prelaz po nekom simbolu azbuke, ili
  - za isto stanje i isti simbol na vrhu steka imamo dva ε-prelaza
- Ponekad se ovaj nedeterminizam može otkloniti tako što „gvirnemo" na ulaz, tj. utvrdimo koji je sledeći simbol na ulazu, bez njegovog čitanja
  - ovaj mehanizam se naziva preduvid, a simbol na ulazu naziva se preduvidni simbol
  - preduvidom se, ponekad, može unapred odrediti koji je prelaz potrebno primeniti i na taj način razrešiti nedeterminizam (ali ne uvek)
- Na žalost, formalna definicija potisnog automata ne omogućava „gvirkanje" na ulaz, bez njegovog čitanja
  - drugim rečima, potisni automat zahteva da unapred odlučimo hoćemo li primeniti prelaz sa čitanjem ulaza ili ε-prelaz, kao i koji od ε-prelaza, u slučaju da ih ima više
- Dobra vest je da se ovakvo ponašanje može simulirati potisnim automatima, uvođenjem dodatnih stanja

Marker kraja ulaza

Marker kraja ulaza

- Još jedan od problema sa preduvidima je u tome što je potrebno da uvek postoji sledeći simbol na ulazu na koji možemo „gvirnuti”:
  - Šta se dešava kada pročitamo celu reč sa ulaza?
  - Tada na ulazu više ne postoji preduvid, pa se ovaj mehanizam od tog trenutka ne može koristiti
  - Ponekad je preduvid potreban i u ovoj završnoj fazi prepoznavanja reči (npr. za prelaze kojima se prazni stek)
- Ovaj problem se razrešava uvođenjem tzv. markera kraja ulaza:
  - u pitanju je novi simbol (obično se obeležava sa $\dashv$) koji se dodaje u azbuku i za koji pretpostavljamo da se uvek nalazi na kraju reči (i samo tu)
  - sada će nakon čitanja cele reči na ulazu ostati marker kraja ulaza, a preduvidom u ovaj simbol možemo odlučiti o daljim koracima
  - u praksi, ulogu markera kraja ulaza igra *EOF* token (koji leksički analizator šalje sintaksnom analizatoru kada pročita kraj ulaza)

Proširene gramatike

Formalno, marker kraja ulaza se uvodi pomoću pojma proširene gramatike:

Definicija 9

Neka je data gramatike $G = (\Sigma, N, S, P)$. Proširena gramatika koja je pridružena ovoj gramatici je $G' = (\Sigma \cup \{\dashv\}, N \cup \{S'\}, S', P \cup \{S' \rightarrow S \dashv\})$.

Teorema 6

$L(G') = \{w \dashv \mid w \in L(G)\}$.

Napomene

- Gramatika G' izvodi iste reči kao i G, uz dodatak markera kraja ulaza na kraju svake reči
- Marker kraja ulaza se nikada ne nalazi u sredini reči, već isključivo na kraju

Važno

U nastavku ćemo uvek podrazumevati da radimo sa proširenom gramatikom date gramatike i to nećemo posebno naglašavati

Primer

Primer

Neka je data gramatika $S \to aSb \mid \varepsilon$. Proširena gramatika koja odgovara ovoj gramatici je $S' \to S \dashv, S \to aSb \mid \varepsilon$. Standardni nedeterministički automat koji prepoznajue jezik $L(G')$ dat je prelazima:

$$
\begin{array}{rcl}
\delta(q_0, \varepsilon, S') & = & \{(q_0, S \dashv)\} \\
\delta(q_0, \varepsilon, S) & = & \{(q_0, aSb), (q_0, \varepsilon)\} \\
\delta(q_0, a, a) & = & \{(q_0, \varepsilon)\} \\
\delta(q_0, b, b) & = & \{(q_0, \varepsilon)\} \\
\delta(q_0, \dashv, \dashv) & = & \{(q_0, \varepsilon)\}
\end{array}
$$

Da bismo determinizovali ovaj automat, možemo dodati stanja q_a, q_b i $q_\dashv$. Ova stanja intuitivno označavaju preduvidni simbol. Naime, mi ćemo uvek pročitati preduvidni simbol i inkorporirati ga u stanje automata. Kada se nalazimo npr. u stanju q_a, tada znamo da je sledeći simbol na ulazu a (on formalno više nije na ulazu, jer smo ga pročitali i ubacili u stanje), te možemo da uz pomoć te informacije odlučujemo o sledećem prelazu automata. Prelazi determinističkog automata dati su na sledećem slajdu.

Primer

Primer

(nastavak) Prelazi determinističkog automata koji prepoznaje isti jezik su:

$$\begin{array}{rcll}
\delta(q_0, a, Z) & = & \{(q_a, Z)\}, & \forall Z \in \Gamma = \Sigma \cup N \\
\delta(q_0, b, Z) & = & \{(q_b, Z)\}, & \forall Z \in \Gamma = \Sigma \cup N \\
\delta(q_0, \dashv, Z) & = & \{(q_\dashv, Z)\}, & \forall Z \in \Gamma = \Sigma \cup N \\
\delta(q_a, \varepsilon, S') & = & \{(q_a, S \dashv)\} & \\
\delta(q_a, \varepsilon, S) & = & \{(q_a, aSb)\} & \\
\delta(q_b, \varepsilon, S) & = & \{(q_b, \varepsilon)\} & \\
\delta(q_\dashv, \varepsilon, S') & = & \{(q_\dashv, S \dashv)\} & \\
\delta(q_\dashv, \varepsilon, S) & = & \{(q_\dashv, \varepsilon)\} & \\
\delta(q_a, \varepsilon, a) & = & \{(q_0, \varepsilon)\} & \\
\delta(q_b, \varepsilon, b) & = & \{(q_0, \varepsilon\} & \\
\delta(q_\dashv, \varepsilon, \dashv) & = & \{q_0, \varepsilon)\} &
\end{array}$$

Intuitivno, stanje q_0 služi za čitanje preduvida i pamćenje te informacije u vidu odgovarajućeg stanja (prva tri prelaza). Kada je pročitani preduvid isti kao i simbol na vrhu steka, tada se skida taj simbol sa vrha steka i vraćamo se u stanje q_0 (poslednja tri prelaza). Kada je na vrhu steka neterminal, tada se on zamenjuje desnom stranom odgovarajućeg pravila (ovo pravilo nije uvek jedinstveno određeno, pa ovako dobijeni automat ne mora biti deterministički. U narednim lekcijama ćemo se baviti uslovima pod kojima ćemo imati determinizam).

Realizacija u praksi

Kako se ovakav pristup realizuje u praksi?

- Pri praktičnoj (programskoj) realizaciji, najpre se čita sledeći simbol sa ulaza i smešta u promenljivu stanja
  - ova promenljiva se često naziva i preduvidna promenljiva, što je ekvivalentno, jer preduvidi odgovaraju stanjima automata
- Sada na osnovu stanja (tj. preduvidnog simbola) i simbola na vrhu steka određujemo koji prelaz automata primeniti
- Nakon što se preduvidni simbol upotrebi (tj. vratimo se u stanje q_0), čitamo sledeći simbol sa ulaza i smeštamo ga u promenljivu stanja
- Informacije o prelascima (na osnovu stanja i simbola na vrhu steka) se obično nalaze u tablici prelaska automata:
  - Način formiranja tablice prelaska zavisi od konkretnog tipa automata koji koristimo (standardni ili standardni prošireni)
  - Ovi postupci, kao i uslovi pod kojima su dobijeni automati deterministički, su ono čime se bavimo u sledećoj lekciji

Prefiksni jezici i preduvid

Nelogičnost ili ne?

- Automat u prethodnom primeru je prepoznavao jezik praznim stekom
- Ranije smo konstatovali da automati koji prepoznaju praznim stekom mogu deterministički prepoznavati samo prefiksne jezike
- Jezik $\{a^n b^n \mid n \geq 0\}$ nije bio prefiksni, te nije mogao biti deterministički prepoznat na ovaj način
- Ipak, u prethodnom primeru smo napravili deterministički automat koji prepoznaje ovaj jezik praznim stekom. Odkud to??
  - Stvar je u tome što to nije taj jezik, već jezik $\{a^n b^n \dashv \mid n \geq 0\}$ (svim rečima je dodat marker kraja ulaza)
  - Ovako modifikovan jezik postaje prefiksni jezik (jer svaka njegova reč mora da se završava sa $\dashv$), te može biti prepoznat praznim stekom na deterministički način