priprema za 1. međuispit iz predmeta:

UVOD U TEORIJU RAČUNARSTVA (teorijski dio)

TEORIJSKA PITANJA SA 1. KONTROLNH ZADAĆA IZ PREDMETA:
AUTOMATI, FORMALNI JEZICI I JEZIČNI PROCESORI 1

1. Formalno definirati DKA i pripadnu funkciju $\hat{\delta}$.

dka = (Q, Σ, δ, q_0, F)
Q - konačan skup stanja;
Σ - konačan skup ulaznih znakova;
δ - funkcija prijelaza $Q \times \Sigma \to Q$
$q_0 \in Q$ - početno stanje;
$F \subseteq Q$ - skup prihvatljivih stanja

funkcija $\hat{\delta} : Q \times \Sigma^* \to Q$
Σ^* - skup svih mogućih nizova ulaznih znakova, uključujući i prazni niz (ε)

(1) $\hat{\delta}(q, \varepsilon) = q$, gdje je ε prazni niz;
(2) za bilo koji niz ulaznih znakova w i za bilo koji ulazni znak a vrijedi:
$\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$, gdje je $w \in \Sigma^*$ i $a \in \Sigma$

2. Opisati postupak pretvorbe ε –NKA u NKA.

ε –NKA M=(Q, Σ, δ, q_0, F) → NKA M'=(Q', Σ, δ', q_0', F')

(1) $Q' = Q$
(2) $q_0' = q_0$
(3) $F' = F \cup \{q_0\}$ ako ε –OKRUŽENJE(q_0) sadrži barem jedno stanje koje je $\in F$, inače $F' = F$
(4) $\delta'(q, a) = \hat{\delta}(q, a)$, $\forall\, a \in \Sigma$ i $\forall\, q \in Q$

3. Navesti svojstva ε –NKA dobivenog postupkom konstrukcije iz regularnog izraza.

(1) Broj stanja ε –NKA M nikad nije veći od broja 2 $|r|$, gdje je $|r|$ broj znakova u regularnom izrazu r. U pojedinim koracima se ne stvara više od 2 stanja.
(2) ε –NKA M=(Q, Σ, δ, i, $\{f\}$) ima samo jedno završno stanje f. Za završno stanje f vrijedi da je $\delta(f, a) = \emptyset$, $\forall a \in (\Sigma \cup \{\varepsilon\})$
(3) Skup $\delta(q, a)$ sadrži najviše jedno stanje za ulazni znak a iz skupa Σ, dok skup $\delta(q, \varepsilon)$ sadrži najviše dva stanja. Ima li čvor dvije izlazne grane, obje grane su označene ε-prijelazima.

4. Dokazati da su regularni jezici zatvoreni s obzirom na nadovezivanje.

Za regularni izraz r_1r_2 koji definira jezik $L(r_1r_2)= L\ (r_1)\ L\ (r_2)$ konstruira se ε –NKA M na sljedeći način. Pretpostavimo da su prethodno izgrađeni ε –NKA $M_1=(Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ i $M_2=(Q_2, \Sigma_2, \delta_2, i_2, \{f_2\})$ takvi da vrijedi $L(M_1)=L\ (r_1)$ i $L(M_2)=L\ (r_2)$ i da nema prijelaza is stanja f_1 i f_2 niti za jedan ulazni znak (tj. $\delta_1(q_1, a) = \emptyset, \forall a \in (\Sigma_1 \cup \{\varepsilon\})$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\})$). Promjenom imena stanja u Q_1 i Q_2 postiže se da je $Q_1 \cap Q_2=\{\}$. Konstruira se ε –NKA $M_1=(Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, i_1, \{f_2\})$. Novo početno stanje jest i_1, a novo prihvatljivo stanje jest f_2. Stanje i_2 nije više početno stanje, te stanje f_1 nije više pirhvatljivo.

5. Opisati jednostavan algoritam pronalaženja istovjetnih stanja (1. algoritam).

Uvjeti se ispituju za sve parove stanja DKA. Neka se npr. ispituju stanja q_1 i $q_3$nekog DKA $M=(\{q_1, q_2, q_3\}, \{c,d\}, \delta, q_0, F)$:

(1) Napravimo tablicu koja sa svaki ulazni znak ima zasebni stupac, izaberu se dva stanja, odnosno taj par označava redak:

	c	d
q_1 i q_3		

(2) Uvjet podudarnosti ispituje se za sve parove novih stanja koji su zapisani u tablicu. Ako par stanja nije podudaran, onda par stanja označen u prvom retku nije istovjetan i algoritam se zaustavlja. Ako su stanja podudarna, odrede se novi parovi stanja za sve ulazne znakove.

(3) Za novi par stanja u koraku (2) su tri mogućnosti:

Par stanja	**Akcija**
Ista stanja	Nema akcije
Različita stanja, za koje postoji zapis u nekom od prethodnih koraka	Nema akcije
Različita stanja, za koje ne postoji zapis u nekom od prethodnih koraka	Stvori novi redak u tablici i upiši u njega novi par stanja

(4) Ako se u koraku (3) ne zapiše novo redak, onda je par stanja iz 1. retka istovjetan, kao i svi parovi stanja u ostalim retcima.

Ako je zapisan novi redak, idemo na korak (2)

6. Opisati postupak pretvorbe NKA u DKA.

NKA $M=(Q, \Sigma, \delta, q_0, F) \rightarrow$ DKA $M'=(Q', \Sigma, \delta', q_0', F')$

(1) $Q' = 2^Q$, skup svih podskupova skupa stanja NKA Q

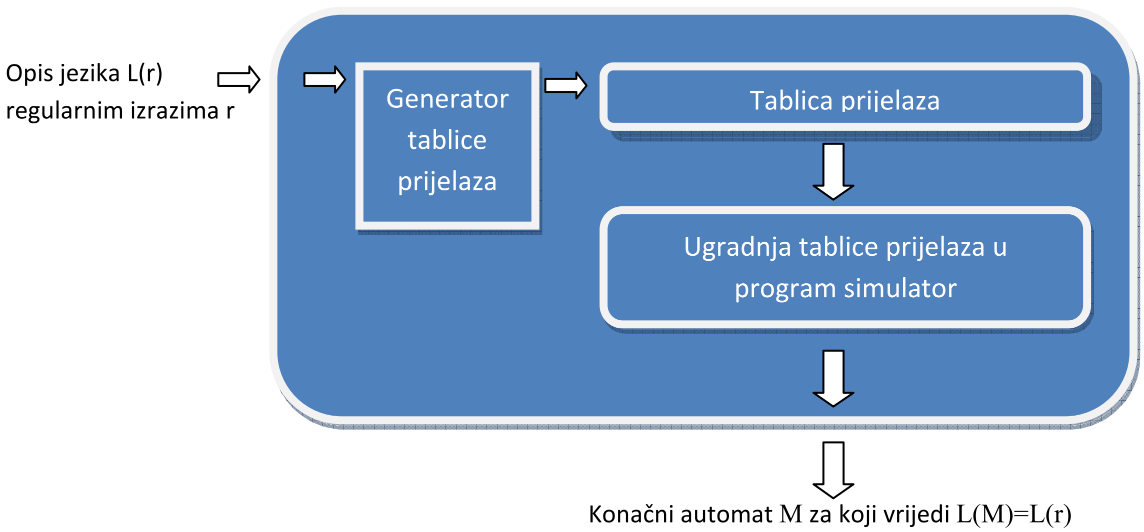
(2) $q_0' = [q_0]$

(3) $F' = [p_1, p_2, \ldots, p_j]$ $p_k \in F$.

(4) $\delta'([p_1, p_2, \ldots, p_l], a) = [r_1, r_2, \ldots, r_j]$, ako i samo ako je $\delta'(\{p_1, p_2, \ldots, p_l\}, a)$ = $\{r_1, r_2, \ldots, r_l\}$, gdje je $a \in \Sigma$

7. Opisati generator konačnog automata.

Generator konačnog automata ostvaruje cjelokupnu ili dio pretvorbe regularnih izraza u DKA.



8. Navesti rekurzivna pravila za regularne izraze te pravila asocijativnosti i prednosti za osnovne operatore koji se koriste u regularnim izrazima.

Rekurzivna pravila:

(1) $\emptyset$ jest regularni izraz i označava jezik $L(\emptyset) = \{\}$

(2) ε jest regularni izraz i označava jezik $L(\varepsilon) = \{\varepsilon\}$

(3) $\forall a \in \Sigma$, a jest regularni izraz i označava jezik $L(a) = \{a\}$

(4) r i s regularni izrazi koji označavju jezike $L(r)$ i $L(s)$:

a) $(r)+(s)$ jest regularni izraz koji označava jezik $L((r)+(s)) = L(r) \cup L(s)$.

b) $(r)(s)$ jest regularni izraz koji označava jezik $L((r)(s)) = L(r)L(s)$.

c) $(r)^*$ jest regularni izraz koji označava jezik $L((r)^*) = L(r)^*$.

Pravila asocijativnosti:

(1) Unarni operator * jest lijevo asocijativan i najveće je prednosti.
(2) Operator nadovezivanja jest lijevo asocijativan i veće je prednosti od operatora +.
(3) Operator + jest lijevo asocijativan i najmanje je prednosti.

9. Navesti definiciju nedohvatljivog stanja i napisati algoritam za pronalaženje nedohvatljivih stanja.

Stanje p DKA M= Q, Σ, δ, q_0, F) jest nedohvatljivo ako ne postoji niti jedan niz $w \in \Sigma$ za koji vrijedi da je p= $\delta(q_0, w)$. Dohvatljiva stanja DKA M=(Q, Σ, δ, q_0, F) se određuju na sljedeći način:

(1) U Listu DS upiše se q_0
(2) Lista DS se proširi skupom stanja $\{p \mid p = \delta(q_0, a), \forall a \in \Sigma\}$
(3) Za sva stanja $q_i \in DS$ proširi se lista skupom stanja $\{p \mid p = \delta(q_0, a)$, stanje p se ne nalazi u listi, $\forall a \in \Sigma\}$

Stanja koja nisu u listi dohvatljivih stanja su nedohvatljiva stanja.

10. Dokazati da su regularni jezici zatvoreni s obzirom na operaciju komplementiranja.

Neka DKA M=(Q, Σ, δ, q_0, F) prihvaća regularni jezik L (M). Za komplement jezika $L(M)^c$ moguće je izgraditi DKA M'=(Q ,Σ, δ, q_0, $Q \setminus F$) koji prihvaća jezik
$L(M')=\{w \mid \delta(q_0, w) \in (Q \setminus F)\} = \{w \mid \delta(q_0, w) \notin F\} = \Sigma^* \setminus \{w \mid \delta(q_0, w) \notin F\} = \Sigma^* \setminus L(M) = L(M)^c$

11. Opisati postupak ispitivanja nepraznosti regularnih jezika.

Neka DKA M prihvaća jezik L (M) i neka jezik DKA M ima n stanja;
Regularni jezik $L(M)$ jest neprazan ako i samo ako DKA M prihvaća niz z duljine manje od n, tj. $|z| < n$.
Za utvrđivanje da li je jezik $L(M)$ koji privaća DKA M neprazan, proširuje se algoritam određivanja nedohvatljivih stanja. Ako je u skupu dohvatljivih stanja barem jedno prhvatljivo stanje, onda je regularni jezik $L(M)$ neprazan.

12. Opisati postupak konstrukcije Mealyevog iz Mooreovog automata.

Mooreov automat M =(Q, Σ, Δ, δ, λ, q_0), istovjetni Mealyev automat M =(Q, Σ, Δ, δ, λ', q_0) gradi se promjenom funkcije izlaza:

(1) $\lambda'(q, a)=\lambda(\delta(q, a))$, za sve q iz Q i za sve a iz Σ.

13. Navesti i objasniti načine programskog ostvarenja funkcije prijelaza.

Funkcija prijelaza ostvaruje se na dva načina: vektorski i listom.

VEKTORSKI PRISTUP definira za svako stanje KA jedan vektor. U vektoru je onoliko elemenata koliko je različitih ulaznih znakova. Osnovna prednost vektorskog pristupa jest brzina računanja novog stanja. Nedostatak vektorskog pristupa je neučinkovito korištenje memorije. Svi vektori su jednake veličine i broj elemenata u svim vektorima jednak je broju ulaznih znakova.

LISTOM se postiže učinkovito korištenje memorije. Za pojedine vektroe definira se lista parova ulaznih znakova i stanja u koje konačni automat prelazi za zadani znak. Računanje novog stanja je duže i računa se u dva koraka:

(1) provjerava se da li je u listi zapis traženog ulaznog stanja, ako je onda se uzme podatak o novom stanju;

(2) ako zapis nije u listi , onda je novo stanje ono koje je navedeno posljednje u listi.

14. Opisati algoritam za utvrđivanje beskonačnosti jezika L(M) kojega prihvaća DKA M.

Neka DKA M prihvaća jezik $L(M)$ i neka jezik DKA M ima n stanja;
Regularni jezik $L(M)$ jest beskonačan ako i samo ako DKA M prihvaća niz duljine l, gdje je $n<l<2n$.
Za utvrđivanje da li je jezik $L(M)$ beskonačan, promatra se dijagram stanja DKA M. Izuzimanjem svih neprihvatljivih stanja, dobije se DKA M' koji prihvaća isti jezik $L(M) = L(M)'$. Ako je u dobivenom dijagramu stanja DKA M' barem jedna zatvorena petlja, onda je regularni jezik $L(M)$ beskonačan.