Vježba 1. Konačni automati (DKA, NKA, ε-NKA)

Deterministički konačni automat (DKA)

Deterministički konačni automat opisuje rad mnogih tehničkih sustava. Teorija konačnog automata ima znatnu ulogu u gradnji tih sustava. Konačni automati imaju primjenu u području modeliranja u medicini, psihologiji, biologiji i dr.

DKA čini:

- skup stanja Q s početnim stanjem $q_0 \in Q$ i skupom prihvatljivih stanja $F \subseteq Q$
- funkcija prijelaza δ: $Q \rightarrow \Sigma \times Q$ jednoznačno određena znakom na ulazu iz skupa Σ i stanjem iz skupa Q

$$\mathrm{dka} = (Q, \Sigma, \delta, q_0, F)$$

DKA prihvaća niz ako nakon svih pročitanih znakova niza x prijeđe iz početnog stanja q_0 u jedno od prihvatljivih stanja $F \subseteq Q$ tj ako vrijedi:

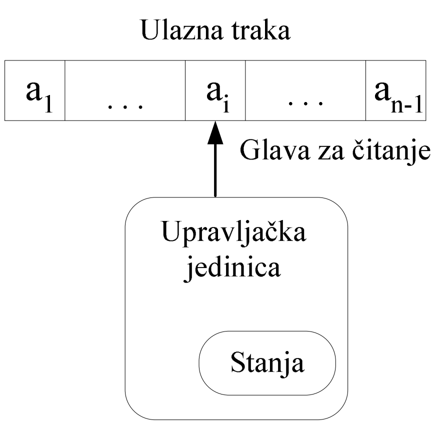
$$\delta(q_0, x) = p, \qquad p \in F$$

DKA prihvaća skup $L(DKA) \subseteq \Sigma^*$ ako je:

$$L(dka) = \{x | \delta(q_0, x) \in F\}$$

gdje je L(DKA) podskup svih mogućih nizova. Za nizove koji nisu u skupu L(DKA) kaže se da ih DKA ne prihvaća.

Model DKA



- Traka je konačna
- Glava samo čita i ne može pisati
- Glava se pomiče samo u desno

Slika 1. Model konačnog automata

Minimizacija konačnog automata

Postoji beskonačno DKA koji prihvaćaju isti jezik. Bilo bi učinkovito izgraditi DKA sa što manjim brojem stanja.

Istovjetnost stanja (ekvivalentnost)

- stanje p DKA M istovjetno je stanju p' DKA M' ako M u p prihvaća isti skup nizova kao M' u p'
- vrijedi:

$$\delta(p,w)\in F \wedge \delta'(p',w)\in F' \;\; ili \;\; \delta(p,w)\notin F \wedge \delta'(p',w)\notin F'$$

Smanjivanje broja stanja

- grupu istovjetnih stanja zamijeniti jednim
- prvo istovjetna stanja označiti istim imenom
- sve prijelaze označiti tim imenom
- u skupu Q ostaviti samo jedno stanje
- po potrebi funkciju prijelaza

Dva automata su istovjetna ako su istovjetna njihova početna stanja. Uvjeti istovjetnosti stanja npr. p i q su:

- uvjet **podudarnosti**: p i q moraju biti prihvatljiva ili neprihvatljiva
- uvjet **napredovanja**: za bilo koji znak, slijedeća stanja $\delta(p,a)= \delta(q,a)$

Za određivanje istovjetnosti stanja p i q postoje tri algoritma:

1. Metoda primitivne tablice
2. Huffman-Mealy algoritam
3. Metoda tablice implikanata

Nedohvatljiva stanja

Stanje p DKA M=(Q, Σ, δ, q_0, F) je nedohvatljivo ako ne postoji nit jedan niz w $\in$ Σ^* za koji vrijedi da je p=(q_0, w). Dohvatljiva stanja određuju se sljedećim algoritmom:

1. U listu dohvatljivih stanja upiše se početno stanje q_0.
2. Lista se proširi sa skupom stanja {p| p= $\delta(q_0, a)$, za sve a $\in$ Σ^*}
3. Za sva stanja $q_i \in$ DS proširi se lista skupom stanja {p| p= $\delta(q_i, a)$, stanje p prethodno zapisano u listu, za sve a $\in$ Σ^*}.

Minimalni DKA

Odbacivanjem istovjetnih i nedohvatljivih stanja dobije se istovjetni DKA s minimalnim brojem stanja. Ne postoji nijedan drugi DKA koji prihvaća isti jezik s manjim brojem stanja. Pošto je postupak odbacivanja nedohvatljivih stanja jednostavniji od postupka odbacivanja istovjetnih stanja, učinkovitije je naprije prvo primjeniti postupak odbacivanja nedohvatljivih stanja, a zatim postupak odbacivanja istovjetnih stanja.

Sinteza DKA

DKA možemo konstruirati na više načina:

1. Ukoliko je regularni jezik zadan regularnim izrazom DKA konstruiramo sljedećim putem:

   RI -> ε-NKA -> NKA -> DKA

2. Transformacijom regularne gramatike (LLRG ili DLRG):

   LLRG -> DLRG -> Jednostavna RG -> NKA -> DKA

3. Putem regularnih izraza, koristeći pravila indeksiranja i rasprostiranja:

   - označimo mjesta unutar regularnog izraza
   - mjesta mogu biti osnovna i predosnovna
   - indeksiramo osnovna mjesta
   - rasprostiremo indekse koristeći pravila o rasprostiranju
   - dodijelimo izlazni simbol akceptorskim mjestima
   - reduciramo indekse koristeći pravila o redukciji
   - ispišemo primitivnu tablicu automata

Izgradnja DKA iz zadanog NKA

- Za bilo koji NKA M={Q, Σ, δ, q_0, F} moguće je izgraditi istovjetni DKA M'={Q',Σ',δ',q0',F'}
- NKA i DKA su istovjetni ako prihvaćaju isti jezik L(M)=L(M')
- ako je Q={q0,...,qi} izgradimo Q'=2Q, [p0,...,pj] ∈ Q', p_k ∈ Q; pa vrijedi: Q'={[∅], [q0],..., [qi], [q0, q1],..., [q0, qj],..., [q0,...,qj] }
- F'=skup svih [p0,...,pj] gdje je barem jedan pk∈ F
- početno stanje je q0'=[q0]
- funkcija prijelaza DKA jest: δ'([p0,...,pl],a) = [r0,...,rj] ako i samo ako je δ({p0,...,pl},a) = {r0,...,rj}
- mnoga stanja dobivenog DKA su nedostupna, pa ih eliminiramo
- DKA minimiziramo radi učinkovite programske realizacije

Primjer 1. Izgradnja DKA iz NKA
Zadan je NKA M=({q0,q1}, {0,1}, δ, q0, {q1})

- tablica prijelaza:

	0	1	⊥
q0	{q0,q1}	{q1}	0
q1	{}	{q0,q1}	1

Q’={[∅], [q0], [q1], [q0, q1]} F’={[q1], [q0, q1]}; q0’=[q0]

δ’ je:

δ’([q0], 0)=[q0, q1] δ’([q0],1)=[q1]
δ’([q1], 0)=[∅] δ’([q1],1)=[q0,q1]
δ’([q0, q1], 0)=[q0, q1] δ’([q0,q1],1)=[q0,q1]
δ’([∅], 0)=[∅] δ’([∅], 1)=[∅]

dobiven je DKA M’=(Q’, {0,1}, δ’, [q0], {[q1], [q0, q1]})

	0	1	⊥
[q0]	[q0,q1]	[q1]	0
[q1]	[∅]	[q0,q1]	1
[q0,q1]	[q0,q1]	[q0,q1]	1
[∅]	[∅]	[∅]	0

Primjer 2. Sinteza DKA putem regularnih izraza koristeći pravila o indeksiranju i rasprostiranju.

Napisati regularni izraz za automat koji na izlazu daje 1 ako je posljednjih 5 bita sekvence bez strukture bilo 10110 (dozvoljeno preklapanje).

Regularni izraz za traženi automat glasi:

$$R=\left(u_0 \vee u_1 u_0 u_0\right)^* u_1\left(u_1 \vee u_0 u_1 u_1 u_1\right)^* u_0\left(u_1 u_0\right)^* u_1 u_1 u_0 \Big|_{i_1}$$

– označimo mjesta:

$$R=|(|u_0| \vee |u_1| u_0| u_0|)^*| u_1|(|u_1| \vee |u_0| u_1| u_1| u_1|)^*| u_0|(|u_1| u_0|)^*| u_1| u_1| u_0|$$

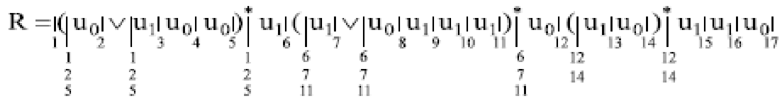
– indeksiramo osnovna mjesta redom:

$$R=\underset{1}{|}(|u_0\underset{2}{|} \vee |u_1\underset{3}{|} u_0\underset{4}{|} u_0\underset{5}{|})^*| u_1\underset{6}{|}(|u_1\underset{7}{|} \vee |u_0\underset{8}{|} u_1\underset{9}{|} u_1\underset{10}{|} u_1\underset{11}{|})^*| u_0\underset{12}{|}(|u_1\underset{13}{|} u_0\underset{14}{|})^*| u_1\underset{15}{|} u_1\underset{16}{|} u_0\underset{17}{|}$$

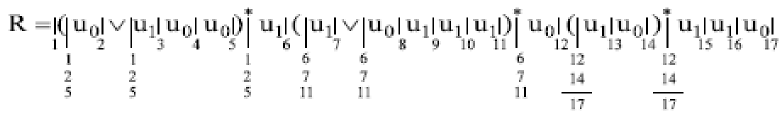
– primijenimo prvo i drugo pravilo rasprostiranja:

$$R=\underset{1}{|}(\underset{1}{|}u_0\underset{2}{|} \vee \underset{1}{|}u_1\underset{3}{|} u_0\underset{4}{|} u_0\underset{5}{|})^*\underset{1}{|} u_1\underset{6}{|}(\underset{6}{|}u_1\underset{7}{|} \vee \underset{6}{|}u_0\underset{8}{|} u_1\underset{9}{|} u_1\underset{10}{|} u_1\underset{11}{|})^*\underset{6}{|} u_0\underset{12}{|}(\underset{12}{|}u_1\underset{13}{|} u_0\underset{14}{|})^*\underset{12}{|} u_1\underset{15}{|} u_1\underset{16}{|} u_0\underset{17}{|}$$

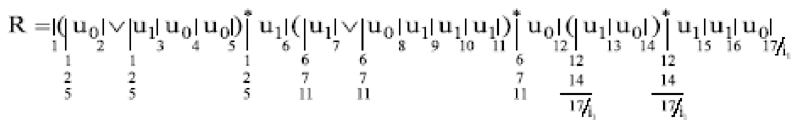
– primijenimo treće i četvrto pravilo rasprostiranja:



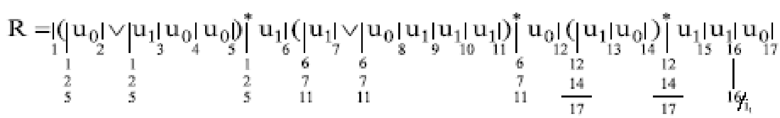
– peto pravilo, dozvolimo preklapanje:



– dodijelimo izlazni simbol, MOORE

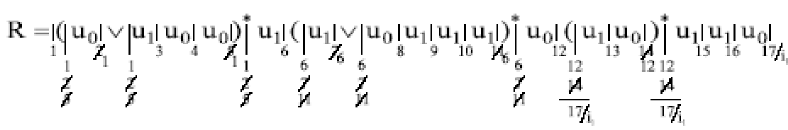


– dodijelimo izlazni simbol, MEALY

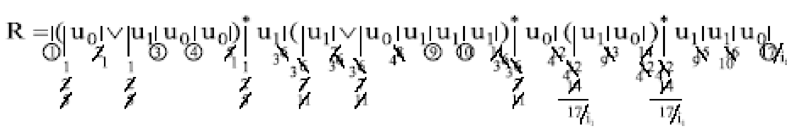


– nastavimo s MOORE automatom

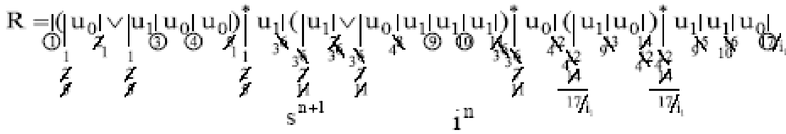
– primijenimo prvo pravilo za redukciju indeksa:



– primijenimo drugo pravilo za redukciju indeksa:

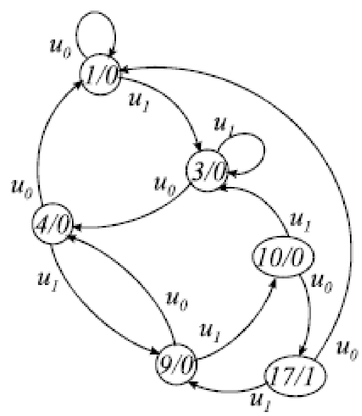


– ispišimo primitivnu tablicu za MOORE automat



	s^{n+1} u_0	u_1	i^n
1	1	3	0
3	4	3	0
4	1	9	0
9	4	10	0
10	17	3	0
17	1	9	1

– nacrtajmo graf za MOORE automat (10110):



Nedeterministički konačni automat (NKA)

Za razliku od determinističkog konačnog automata funkcija prijelaza NKA je nedeterministička. Umjesto δ(q,a)=p imamo: δ(q,a)={p_0, p_1, ...}.

Za ulazni niz w postoji više staza, niz se prihvaća ako bar jedna staza završava prihvatljivim stanjem. Svaki put kada imamo više mogućih prijelaza, stvori se toliki broj DKA koji paralelno obrađuju niz. Niz se prihvaća ako staza makar jednog DKA završi prihvatljivim stanjem. Za bilo koji NKA N moguće je izgraditi DKA D tako da je L(N)=L(D).

NKA čini:

- skup stanja Q s početnim stanjem $q0 \in Q$ i skupom prihvatljivih stanja $F \subseteq Q$
- funkcija prijelaza δ: $Q \rightarrow \Sigma \times 2^Q$ određena znakom na ulazu iz skupa Σ i skupom stanja iz skupa Q
- 2^Q je skup svih podskupova skupa stanja Q

$$\text{nka} = (Q, \Sigma, \delta, q_0, F)$$

NKA prihvaća niz ako je u P makar jedno stanje iz F:

$$\delta(q_0, x) = P; \quad P \cap F \neq \emptyset$$

NKA prihvaća skup L(NKA) $\subseteq \Sigma^*$

$$L(nka) = \{x | \delta(q_0, x) \cap F \neq \emptyset\}$$

Za nizove koji nisu u skupu L(NKA) kaže se da ih NKA ne prihvaća.

Sinteza NKA

NKA možemo konstruirati na dva načina:

1. Ukoliko je regularni jezik zadan regularnim izrazom NKA konstruiramo sljedećim putem:

   RI -> ε-NKA -> NKA

2. Ako je regularni jezik zadan regularnom gramatikom (LLRG ili DLRG) postupak sinteze je sljedeći:

   LLRG -> DLRG -> Jednostavna RG -> NKA

Izgradnja NKA iz zadanog ε-NKA

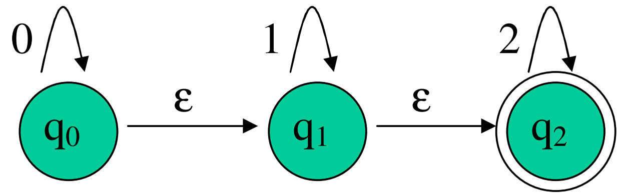
- Za bilo koji ε-NKA M={Q, Σ, δ, q0, F}moguće je izgraditi istovjetni NKA M'={Q',Σ',δ',q0',F'}
- ε- NKA i NKA su istovjetni ako prihvaćaju isti jezik L(M)=L(M')
- izgradimo Q'= Q ={q0,...,qi}
- početno stanje je q0'=q0
- F'=F∪{q0} ako ε-OKRUŽENJE(q0) sadrži barem jedno stanje pk∈ F, inače F'=F
- $\delta'(q, a) = \hat{\delta}(q, a)$

Primjer 3. Izgradnja NKA iz ε-NKA

Zadan je ε-NKA automat koji prihvaća nizove:

- koji započinju proizvoljnim brojem nula,
- nastavljaju proizvoljnim brojem jedinica i
- završavaju proizvoljnim brojem dvojki

L={$0^n 1^m 2^1$ | n, m ,l ≥ 0}, uključuje i prazni niz ε



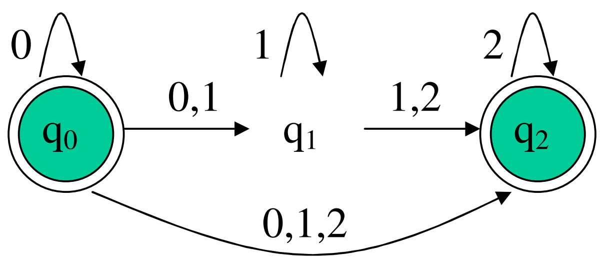
- izgradimo Q' = Q = {q0, q1, q2}
- početno stanje je q0'=q0
- F' = F∪{q0} = {q2}∪{q0} = {q0, q2}
  jer je ε-OKRUŽENJE(q0)∩F = {q0, q1, q2}∩{q2} = {q2}
- funkcija δ': $\delta'(q_0,0)=\{q_0,q_1,q_2\}$ jer je:

$$\hat{\delta}(q_0,0)=\varepsilon-\mathrm{OKRUŽENJE}\big(\delta\big(\hat{\delta}(q_0,\varepsilon),0\big)\big)=\{q_0,q_1,q_2\}$$

- itd.

Dobije se NKA:

	0	1	2	⊥
q0	{q0,q1,q2}	{q1,q2}	{q2}	1
q1	∅	{q1,q2}	{q2}	0
q2	∅	∅	{q2}	1



Nedeterministički konačni automat s ε-prijelazima (ε-NKA)

ε-NKA automati omogućavaju promjenu stanja za prazni niz ε (epsilon). Promjena stanja bez čitanja znaka naziva se ε-prijelaz. Slično NKA automatu, ε-NKA automat prihvaća niz ako postoji najmanje jedan slijed prijelaza iz početnog u prihvatljivo stanje,uključujući ε-prijelaze uz uvjet da se pročitaju svi znakovi niza.

ε-NKA čini:

- skup stanja Q s početnim stanjem $q_0 \in Q$ i skupom prihvatljivih stanja $F \subseteq Q$
- funkcija prijelaza δ: $Q\times(\cup\Sigma\{\varepsilon\})\rightarrow 2^Q$ određena znakom na ulazu iz skupa Σ prošireno s ε i skupom stanja iz skupa Q
- 2^Q je skup svih podskupova skupa stanja Q

$$\varepsilon - nka = \left(Q, \Sigma, \delta, q_0, F\right)$$

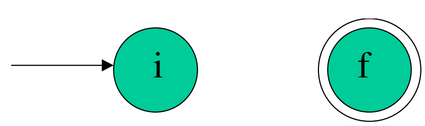
Definiranje funkcije prijelaza zahtijeva i definiranje funkcije ε-okruženje(q) (ε-closure). Stanju $q \in Q$ dodijeljujemo skup $R \subseteq Q$ tako da R sadrži sva ona stanja u koja q prelazi isključivo ε prijelazom:

ε-OKRUŽENJE(q) = {p| p jest q ili ε-NKA prelazi iz q u p isključivo ε-prijelazom}

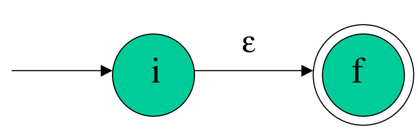
Izgradnja ε-NKA iz zadanog RI

Za bilo koji regularni izraz R moguće je izgraditi ε-NKA M tako da vrijedi L(M) = L(R).

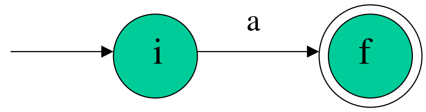
- za **RI ∅** koji označava jezik L(∅)={} izgradimo ε-NKA
  M=({i,f}, Σ, { }, i, {f}) koji ne prihvaća ni jedan niz:
-



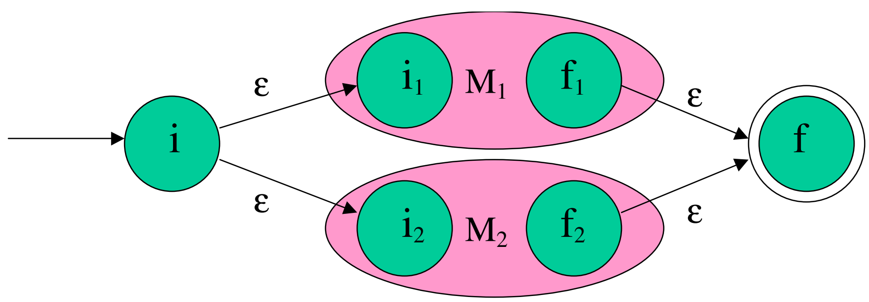
- za **RI ε** koji označava jezik L(ε)={ε} izgradimo ε-NKA
  M=({i,f}, Σ, {δ(i,ε)=f}, i, {f}) koji prihvaća prazni niz ε



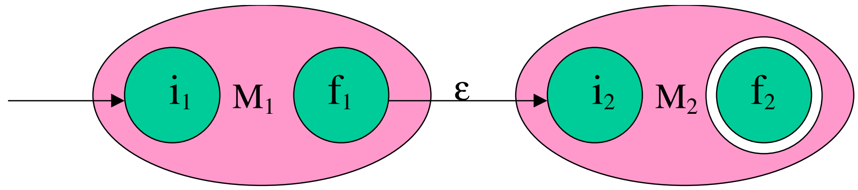
- za **RI a** koji označava jezik L(a)={a} izgradimo ε-NKA
  M=({i,f}, Σ, {δ(i,a)=f}, i, {f}) koji prihvaća niz a, ne prihvaća ε niti b od b∈Σ



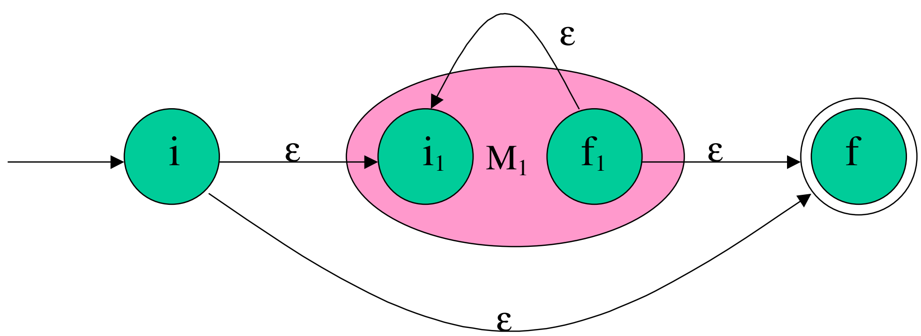
- za **RI $\mathbf{r_1 \vee r_2}$** koji označava jezik $L(r_1 \vee r_2) = L(r_1) \cup L(r_2)$ izgradimo ε-NKA M postupkom:
    1. izgradimo ε-NKA $M_1 = (Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ da je $L(M_1) = L(r_1)$
    2. izgradimo ε-NKA $M_2 = (Q_2, \Sigma_2, \delta_2, i_2, \{f_2\})$ da je $L(M_2) = L(r_2)$
    3. f_1 i f_2 nemaju prijelaza,
       tj. $\forall a \in (\Sigma_1 \cup \{\varepsilon\}) : \delta_1(f_1,a) = \varnothing$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\}) : \delta_2(f_2,b) = \varnothing$
    4. stanja iz Q_1 i Q_2 imenujemo tako da je $Q_1 \cap Q_2 = \varnothing$
    5. izgradimo ε-NKA $M = (Q_1 \cup Q_2 \cup \{i,f\}, \Sigma_1 \cup \Sigma_2, \delta, i, \{f\})$
       f je novo prihvatljivo stanje, a f_1 i f_2 više nisu prihvatljiva
- konstruiramo δ:
    1. $\delta(i, \varepsilon) = \{i_1, i_2\}$
    2. $\delta(q, a) = \delta_1(q, a) \quad \forall q \in (Q_1 \backslash \{f_1\})$ i $\forall a \in (\Sigma_1 \cup \{\varepsilon\})$
    3. $\delta(q, b) = \delta_2(q, b) \quad \forall q \in (Q_2 \backslash \{f_2\})$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\})$
    4. $\delta(f_1, \varepsilon) = \delta(f_2, \varepsilon) = \{f\}$



- za **RI $\mathbf{r_1 r_2}$** koji označava jezik $L(r_1 r_2) = L(r_1)L(r_2)$ izgradimo ε-NKA M postupkom:
    1. izgradimo ε-NKA $M_1 = (Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ da je $L(M_1) = L(r_1)$
    2. izgradimo ε-NKA $M_2 = (Q_2, \Sigma_2, \delta_2, i_2, \{f_2\})$ da je $L(M_2) = L(r_2)$
    3. f_1 i f_2 nemaju prijelaza,
       tj. $\forall a \in (\Sigma_1 \cup \{\varepsilon\}) : \delta_1(f_1,a) = \varnothing$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\}) : \delta_2(f_2,b) = \varnothing$
    4. stanja iz Q_1 i Q_2 imenujemo tako da je $Q_1 \cap Q_2 = \varnothing$
    5. izgradimo ε-NKA $M = (Q_1 \cup Q_2 \cup \{i,f\}, \Sigma_1 \cup \Sigma_2, \delta, i_1, \{f_2\})$
    6. i_1 je novo početno stanje, a f_1 više nije prihvatljivo
    7. i_2 više nije početno stanje, a f_2 je novo prihvatljivo stanje
- konstruiramo δ:
    1. $\delta(q, a) = \delta_1(q, a) \quad \forall q \in (Q_1 \backslash \{f_1\})$ i $\forall a \in (\Sigma_1 \cup \{\varepsilon\})$
    2. $\delta(q, b) = \delta_2(q, b) \quad \forall q \in (Q_2 \backslash \{f_2\})$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\})$
    3. $\delta(f_1, \varepsilon) = \{i_2\}$

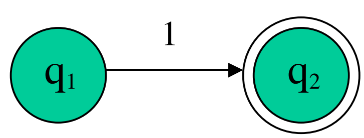


- za **RI r_1*** koji označava jezik $L(r_1{}^*) = L(r_1)^*$ izgradimo ε-NKA M postupkom:
  1. izgradimo ε-NKA $M_1 = (Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ da je $L(M_1) = L(r_1)$
  2. f_1 nema prijelaza, tj. $\forall a \in \Sigma_1$: $\delta_1(f_1,a) = \varnothing$
  3. izgradimo ε-NKA $M = (Q_1 \cup \{i,f\}, \Sigma_1, \delta, i, \{f\})$
  4. i_1 više nije početno stanje, a f_1 više nije prihvatljivo
- konstruiramo δ:
  1. $\delta(i, \varepsilon) = \delta(f_1, \varepsilon) = \{i_1, f\}$
  2. $\delta(q, a) = \delta_1(q, a) \quad \forall q \in (Q_1 \setminus \{f_1\}) \quad i \quad \forall a \in (\Sigma_1 \cup \varepsilon)$

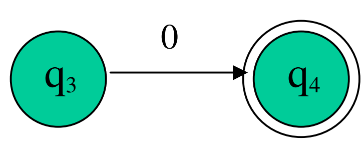


Primjer 4.
Iz regularnog izraza $r = 01^* \vee 1$ konstruirati ε-NKA automat.

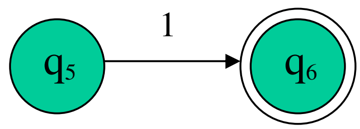
- RI r rastavimo na $r = r1 \vee r2$; $r1 = 01^*$ i $r2 = 1$
- za $r2 = 1$ izgradimo ε-NKA:



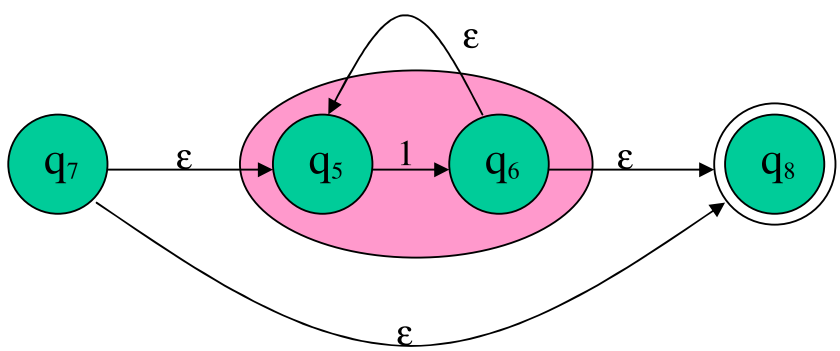
- $r1 = 01^*$ rastavimo na $r1 = r3r4$; $r3 = 0$ i $r4 = 1^*$
- za $r3 = 0$ izgradimo ε-NKA:



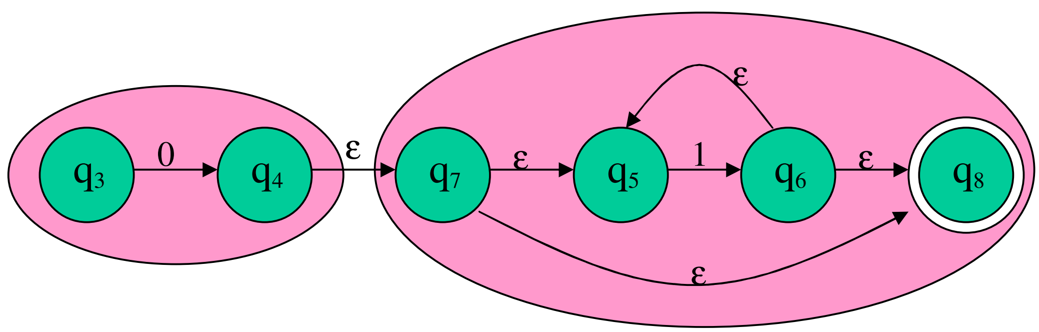
- r4 = 1* rastavimo na r4 = r5* ; r5 = 1
- za r5 = 1 izgradimo ε-NKA:



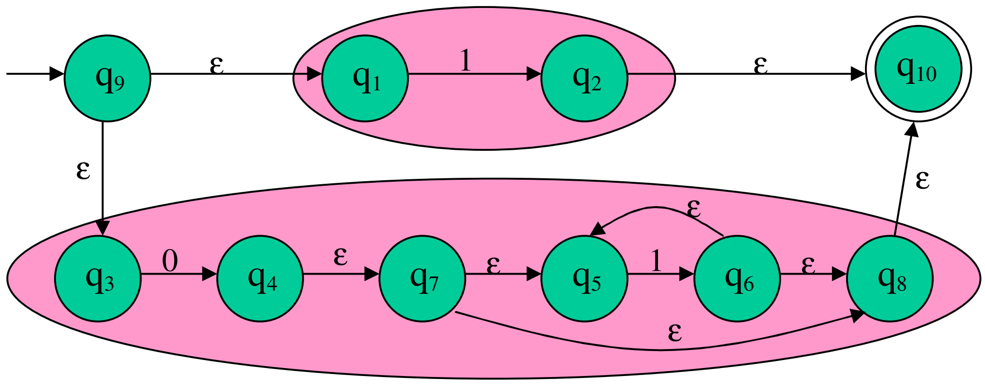
- za r4 = 1* izgradimo ε-NKA prema pravilu p6:



- za r1 = r3r4 izgradimo ε-NKA prema pravilu p5:



- za r = r1∨ r2 izgradimo ε-NKA prema pravilu p4:



Zadatak 1. Iz zadanog regularnog izraza konstruirati deterministički konačni automat s izlazom (Moore-ov model) koristeći pravila o indeksiranju i rasprostiranju.

Zadatak 2. Iz zadanog regularnog izraza izgraditi NKA automat sljedećim putem:

RI -> ε-NKA -> NKA