Teori Gjuhësh:

Bazat e Informatikës Teorike

Cikël leksionesh

nga v. Claus dhe E.-R.
Olderog
Përshtatur nga Denis
Saatçiu

Pasqyra e lendes

Kapitulli I

Perkufizimet baze

§1 Logjika, bashkesite, relacionet dhe funksionet

Logjika

Ne formulat logjike perdorim:

- lidhezat logjike
  - $\neg$ (mohim), $\wedge$ (dhe), $\vee$ (ose),
  - $\Rightarrow$ (implikim) dhe $\Leftrightarrow$ (ekuivalence),
- sasioret
  - $\forall$ (sasiorin universal)
  - $\exists$ (sasiorin ekzistencial).

Bashkesite

Mund te paraqesim bashkesite e fundme duke listuar elementet e tyre, p.sh., {a, b, c, d} ose {kafe, caj, sheqer}.

Bashkesia boshe percaktohet nga $\emptyset$.

Nje rast i bashkesise se pafundme eshte bashkesia N e numrave natyrore: N = {0, 1, 2, 3, ...}.

$x \in X$ percakton qe x eshte nje element I bashkesise X, dhe $x \notin X$ percakton qe x nuk eshte nje elementi i bashkesise X.

Parimi i zgjerueshmerise: dy bashkesi jane te barabarta nese ato kane te njejtet elemente. P.sh,
{a, b, c, d} = {a, a, b, c, d, d}.

$X \subseteq Y$ tregon qe X eshte nje nenbashkesi e Y ,p.sh. $\forall x \in X : x \in Y$.

Nenbashkesite mund te zgjidhen nga nje bashkesi e dhene duke perdorur formula logjike.
P.sh, $M = \{n \in N \mid n \bmod 2 = 0\}$tregon bashkesine e numrave natyrore cift.

Per bashkesite X, $Y \subseteq Z$ ekzistojne operacionet e meposhtme:

- Bashkimi $X \cup Y=\{z \in Z \mid z \in X \vee z \in Y \}$
- Prerja $X \cap Y =\{z \in Z \mid z \in X \wedge z \in Y \}$
- Diferenca $X - Y = X\backslash Y = \{z \in Z \mid z \in X \wedge z \in Y \}$

- Komplementi X = Z – X

|X| dhe kard(X) percaktojne kardinalitetin ose madhesine e bashkesise X. |X| eshte numri I elementeve ne bashkesine X kur X eshte I fundem. P.sh., |{kafe, caj, sheqer}| = 3.

P(Z) percakton fuqine e nje bashkesie Z, p.sh bashkesia e te gjithe nenbashkesive te Z: $P(Z) =\{X \mid X \subseteq Z\}$. Vecanerisht $\emptyset \in P(Z)$ dhe $Z \in P(Z)$.

$X \times Y$ percakton prodhimin kartezian te dy bashkesive X dhe Y, qe permban te gjitha dyshet ku elementi I pare eshte nga X dhe elementi I dyte eshte nga Y: $X \times Y = \{(x, y) \mid x \in X \wedge y \in Y\}$.

Ne pergjithesi, $X_1 \times \ldots \times X_n$ percakton bashkesine te te gjitha n-sheve, ku

$\forall i \in \{1, \ldots, n\}$ kemi qe X_i permban elementin e i-te: $X_1 \times \cdots \times X_n = \{(x_1, \ldots, x_n) \mid x_1 \in X_1 \wedge \ldots \wedge x_n \in X_n\}$.

Relacionet

Relacionet jane bashkesi speciale. Nje relacion dysh R I dy bashkesive X dhe Y eshte nje nenbashkesi e prodhimit $X \times Y$, p.sh. $R \subseteq X \times Y$. Paraqitja infix xRy perdoret shpesh per te percaktuar lidhjet e elementeve $(x, y) \in R$.

Domain te R do te quajme

$$\mathrm{dom}(R) = \{x \in X \mid \exists\, y \in Y : (x, y) \in R\}$$

dhe rang te R do te quajme

$$\mathrm{ran}(R) = \{y \in Y \mid \exists\, x \in X : (x, y) \in R\}.$$

Relacioni $R \subseteq X \times Y$ eshte

- Unik nga e majta, nqs $\forall\, x_1, x_2 \in X, y \in Y : (x_1, y) \in R \wedge (x_2, y) \in R \Rightarrow x_1 = x_2$,
- Unik nga djathta, nqs $\forall\, x \in X, y_1, y_2 \in Y : (x, y_1) \in R \wedge (x, y_2) \in R \Rightarrow y_1 = y_2$,
- Total nga e majta, nqs $X = \mathrm{dom}(R)$,
- Total nga e djathta, nqs $\mathrm{ran}(R) = Y$.

Relacion te identitetit ne X do te quajme id_X: $id_X = \{(x, x) \mid x \in X\}$.

Relacion invers te $R \subseteq X \times Y$ eshte $R^{-1} \subseteq Y \times X$ qe percaktohet si me poshte:

$$\forall x \in X, y \in Y : (x, y) \in R \Leftrightarrow (y, x) \in R^{-1}$$

Kompozimi $\circ$ I dy relacioneve $R \subseteq X \times Y$ dhe $S \subseteq Y \times Z$ percaktohet si me poshte:

$$\forall\, x \in X, z \in Z \text{ kemi } (x, z) \in R \circ S \Leftrightarrow \exists\, y \in Y : (x, y) \in R \text{ dhe } (y, z) \in S.$$

Relacion dysh ne bashkesine X eshte relacioni $R \subseteq X \times X$. Ky relacion eshte

- refleksiv, nqs $\forall\, x \in X : (x, x) \in R$, ose ne terma te relacionit: $id_X \subseteq R$,
- jorefleksiv, nqs $\neg\, \exists\, x \in X : (x, x) \in R$, ose ne terma te relacionit: $R \cap id_X = \emptyset$,

- simetrik, nqs $\forall\, x, y \in X : (x, y) \in R \Rightarrow (y, x) \in R$, ose ne terma te relacionit: $R = R^{-1}$,
- antisimetrik, nqs $\forall\, \cancel{x,}\; y \in X: (x, y) \in R \wedge (y, x) \in R \Rightarrow x=y$, ose ne terma te relacionit: $R \cap R^{-1} \subseteq id_X$,
- kalimtar, nqs $\forall\, x, y, z \in X : (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$, ose ne terma te relacionit: $R \circ R \subseteq R$

R eshte nje relacion ekuivalence, nqs R eshte refleksiv, simetrik dhe kalimtar.

Relacioni ekuivalences R ne X e ndan bashkesine X ne nenbashkesi pa prerje, p.sh. secila nenbashkesi permban cifte elementesh. $[x]_R$ eshte nje klase ekuivalence e x per elementet $x \in X$, p.sh. bashkësia $[x]_R = \{y \in X \mid (x, y) \in R\}$.

Nje element I nje klase ekuivalene quhet perfaqesues I kesaj klase, sepse e gjithe klasa mund te identifikohet nga perfaqesuesi dhe relacioni R. Per me teper elemente te cfaredoshem mund te zgjidhen si elemente perfaqesues te klases $\forall x, y \in X : (x, y) \in R \Leftrightarrow [x]_R = [y]_R$.

Indeks te R do te quajme kardinalitetin e te gjitha klasave te ekuivalences te R ne X.

Verrejtje:

$$\mathrm{Index}(R) = |\{ [x]_R \mid x \in X\}|$$

Nqs mund te percaktohet nga konteksti se cila R konsiderohet, mund te shkruajme [x] ne vend te $[x]_R$.

Nje relacion ekuivalence R ne X quhet nje permiresim I nje relacioni ekuivalence S ne X, nqs $R \subseteq S$. Atehere, cdo klase ekuivalence e R eshte nje nenbashkesi e ndonje klase ekuivalence te S.

Fuqia e n-te e $R \subseteq X \times X$ percaktohet induktivisht:

$$R^0 = id_X \text{ dhe } R^{n+1} = R \circ R^n$$

Mbyllja kalimtare R^+ dhe mbyllja kalimtare refleksive R^* e R percaktohet si meposhte:

$$R^+ = \bigcup_{n \in \mathbb{N}\setminus\{0\}} R^n \qquad \text{and} \qquad R^* = \bigcup_{n \in \mathbb{N}} R^n$$

Funksionet

Funksionet jane relacione te vecanta. Nje relacion $f \subseteq X \times Y$ quhet funksion I pjesshem (ose imazh I pjesshem) nga X ne Y, nqs f eshte unik nga e djathta. Percaktohet si $f: X \rightarrow part_Y$.

Ne vend te $(x, y) \in f$, shkruajme $f(x) = y$. f eshte nje funksion (total) nga X ne Y, nqs f eshte edhe total nga e djathta. Kjo percaktohet $f : X \rightarrow Y$.

Nje funksion $f : X \rightarrow Y$ eshte

- ✓ injektiv, nqs f eshte total nga e majta,
- ✓ surjektiv, nqs f eshte total nga e djathta,
- ✓ bijektiv, nqs f eshte injektiv dhe surjektiv.

Nje funksion bijektiv quhet ndryshe edhe bijeksion.

§2 Alfabetet, stringjet dhe gjuhet

Me poshte do I kushtohet vemendje analizimit te gjuheve formale. Per kete qellim do perdoren perkufizimet e meposhtme

- Alfabet = bashkesi e fundme karakteresh (simbole) = bashkesi karakteresh

Zakonisht perdoren A, B, Σ, Γ si emra tipike per alfabetet dhe a, b si emra tipike per karakteret, p.sh. germat e alfabeteve.

- String mbi nje alfabet Σ = sekuence e fundme karakteresh nga Σ.

Me perkufizim: ε - string bosh. Si emra tipike per stringjet perdoren u, v, w.

Shembull: Le te jete Σ = {1, 2, +}. Atehere 1 + 2 dhe 2 + 1 jane stringje mbi Σ.

Σ^*=bashkesia e te gjithe stringjeve mbi Σ, $\Sigma \subseteq \Sigma^*$.

Σ^+= bashkesia e te gjithe stringjeve jo bosh mbi Σ, i.e. $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$.

$|u|$ = gjatesia e stringut u = numri karaktereve qe ndodhen u.

Me perkufizim: $|\varepsilon| = 0$.

Simbolet e stringut u me gjatesi n do te shenohen $u_1, \ldots, u_n$.

Ngjitja e u dhe v perbehet nga karakteret e u te ndjekur nga karakteret e v dhe shenohet me $u \cdot v$ ose thjesht me uv.

Shembull: Ngjitje e 1+ dhe 2 + 0 eshte 1 + 2 + 0.

- Stringu v quhet nenstring I stringut w, nqs ∃u1, u2: w = u1vu2.
- Stringu v quhet prefiks I stringut w, nqs ∃u : w = vu.
- Stringu v quhet sufiks i stringut w, nqs ∃u : w = uv.

Nje nenstring mund te ndodhet disa here ne stringun w.

Shembull: Stringu w = 1 + 1 + 1 permban dy here nenstringun 1+ dhe tre here nenstringun 1

- Nje gjuhe (formale) A mbi nje alfabet Σ eshte nje nenbashkesi e Σ^*.

Si emer tipik per nje gjuhe perdoret L.

Operacionet standarte mbi gjuhet:

- ✓ $L_1 \cup L_2$ (bashkim)
- ✓ $L_1 \cap L_2$ (prerje)
- ✓ $L_1 - L_2$ ose $L_1 \setminus L_2$ (diference)
- ✓ $\overline{L} =_{df} \Sigma^* - L$ (komplement)

Operacione shtese mbi gjuhet.

Bashkimi I stringjeve zgjerohet ne gjuhet L_1 dhe L_2:

$$L_1 \cdot L_2 = \{u \cdot v \mid u \in L_1 \text{ dhe } v \in L_2\}.$$

Fuqia e n-te e gjuhes L percaktohet ne menyre induktive: $L^0 = \{\varepsilon\}$ and $L^{n+1} = L \cdot L^n$

Operatori * (Kleene) (ose mbyllja e Kleene) I nje gjuhe L eshte

$$L^* = \bigcup_{n \in N} L^n = \{w_1 \ldots w_n \mid n \in N \text{ dhe } w_1, \ldots, w_n \in L\}.$$

Per te gjitha L kemi $\varepsilon \in L^*$

Kapitulli II

Automatet e fundem dhe gjuhet e regullta

Ne kapitujt ne vijim, do trajtojme nje model te thjeshte te nje makine: automatin e fundem. Do verejme qe:

- ✓ automatet e fundem perdoren gjeresisht ne shkencat kompiuterike,
- ✓ gjuhet e njohura nga automatet e fundem kane shume vecori strukturore, psh perfaqesimi nga shprehjet e rregullta,
- ✓ pyetjet per perllogaritjet e kryera nga automatet e fundem jane te percaktuese (*decidable*).

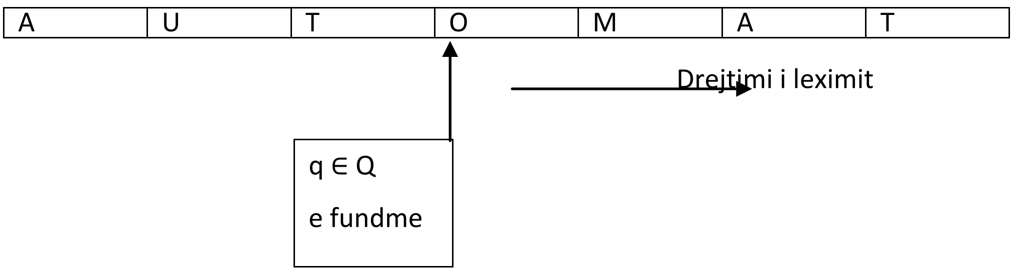
§1 Automatet e fundem

Automatet e fundem do I perdorim per te njohur gjuhet. Nje automat I fundem mund te mendohet si nje makine me gjendje fundore, e cila lexon karaktere nga nje kasete, mund te levize koken vetem djathtas dhe nuk mund te printoje simbole te reja ne kasete.

.

Prandaj, funksioni I kalimit te automatit te fundem percaktohet si $\delta : Q \times \Sigma \rightarrow Q$, ku Q eshte nje bashkesi gjendjesh dhe Σ eshte nje alfabet hyres per automatin. Ky automat mund ti aplikohet stringut $w \in \Sigma^*$ per te kontrolluar ne eshte I pranueshem.

Figura:



Paraqitja e automatit si sisteme kalimi eshte me e pershtatshme per paraqitje grafike dhe per percaktimin e sjelljes pranuese te automatit.(sjellja pranuese e automatit: si automati njeh gjuhen)

.

Automati I fundem determinist

Percaktimi : Nje automat I fundem determinist (pranues), shkurtimisht AFD, eshte nje 5-she $A= (\Sigma, Q, \delta, q_0, F)$ ose $A = (\Sigma, Q, \rightarrow, q_0, F)$ me vecorite e meposhtme:

1. Σ, alfabet hyres, eshte I fundem
2. Q eshte nje bashkesi e fundme gjendjesh∈
3. $\delta : Q \times \Sigma \rightarrow Q$ eshte funskioni I kalimit

   resp. $\rightarrow \subseteq Q \times \Sigma \times Q$ eshte nje relacion kalimi determinist, p.sh,$\forall$ q $\in$ Q $\forall$ a $\in \Sigma$ $\exists$ vetem nje $q^1 \in$ Q : (q, a, q^1) $\in \rightarrow$,

4. $q_0 \in$ Q eshte gjendja fillestare
5. F $\subseteq$ Q eshte bashkesia e gjendjeve fundore

Lidhja e dy paraqitjeve te automatit eshte dhene me poshte:

$\delta(q, a) = q^1 \Leftrightarrow (q, a, q^1) \in \rightarrow$

Elementet (q, a, q^1) $\in \rightarrow$ quhen kalime. Pergjithesisht shkruhet q $\xrightarrow{a}$ q^1 ne vend te (q, a, q^1) $\in \rightarrow$.

Per a $\in \Sigma$ perdorim $\xrightarrow{a}$ edhe si nje relacion dysh $\xrightarrow{a}$ $\subseteq Q \times Q$:

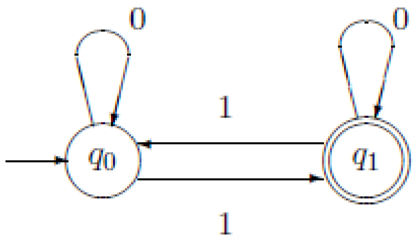
$\forall q, q^1 \in Q : q \xrightarrow{a} q^1 \Leftrightarrow (q, a, q^1) \in \rightarrow$

Nje AFD mund te paraqitet grafikisht me ane te nje diagrame me numer te fundem gjendjesh. Ky eshte nje graf I orientuar, I cili permban nje etikete te emertuar me *q* per cdo gjendje *q* te automatit dhe nje shigjete te orientuar per cdo tranksasion q $\xrightarrow{a}$ q^1. Gjendja fillestare q_0 shenjohet me nje shigjete hyrese. Gjendjet fundore paraqiten me rrathe shtese

Shembull : Kemi A_1= ({0, 1}, {q_0, q_1}, $\rightarrow$, q_0, {q_1}) ku

$$\rightarrow = \{(q_0, 0, q_0), (q_0, 1, q_1), (q_1, 0, q_1), (q_1, 1, q_0)\}.$$

Automati A_1 do paraqitet me ane te diagrames me poshte:



Percaktime (Pranueshmeria dhe kapshmeria)

Le te jete A= (Σ, Q, $\rightarrow$, q0, F) resp.A= (Σ, Q, δ, q0 , F) te jete nje AFD.

1. Zgjerojme relacionet e kalimit $\xrightarrow{a}$ nga simboli hyres a $\in \Sigma$ te stringjet e simboleve,w $\in \Sigma^*$ dhe percaktojme relacionet respektive $\xrightarrow{w}$ induktivisht:

- q $\xrightarrow{\varepsilon}$ q^1 vetem nqs $q=q^1$ ose ne terma te relacionit $\xrightarrow{\varepsilon}$ $=id_Q$
- q $\xrightarrow{aw}$ q^1 vetem nqs $\exists$ $q^{ii} \in$ Q : q $\xrightarrow{a}$ q^{ii} dhe q^{ii} $\xrightarrow{w}$ q^1

  Ose ne terma te relacionit $\xrightarrow{aw}$ = $\xrightarrow{a} \circ \xrightarrow{w}$

Ne menyre te ngashme, percaktojme funksionin e zgjeruar te kalimit δ^*

$\delta^*: Q \times \Sigma^* \rightarrow Q$ me $\delta^*(q, \varepsilon) = q$ dhe $\delta^*(q, aw) = \delta^*(\delta(q, a), w)$ per te gjitha q, $q^1 \in$ Q, a $\in \Sigma$ and w $\in \Sigma^*$.

Pra:

$$\delta^*(q, w) = q^1 \Leftrightarrow q \xrightarrow{w} q^1$$

2. Gjuha e pranuar nga automati A eshte

L(A) = $\{w \in \Sigma^* \mid \exists q \in F : q_0 \xrightarrow{w} q\}$ resp. L(A) = $\{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$

Nje gjuhe L pranohet, nqs ekziston nje AFD A, ku L=L(A).

3.Nje gjendje $q \in Q$ eshte e kapshme ne A, nqs $\exists w \in \Sigma^*: q_0 \xrightarrow{w} q$.

Shembull: Per automatin e shembullit me lart kemi:

$L(A_1) = \{w \in \{0, 1\}^* \mid w \text{ ka numer tek } 1\}$.

Shenim:Per te gjitha a1, . . . , an$\in \Sigma$ and q, $q^1 \in Q$:

$$q \xrightarrow{a_1 \cdot \ldots \cdot a_n} q' \quad \Leftrightarrow \quad \exists\, q_1, \ldots, q_n \in Q : q \xrightarrow{a_1} q_1 \quad \ldots \quad q_{n-1} \xrightarrow{a_n} q_n = q'.$$

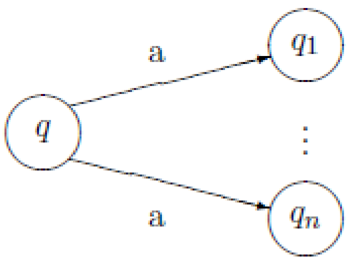
Ose ne terma te relacionit:

$$\xrightarrow{a_1 \cdot \ldots \cdot a_n} = \xrightarrow{a_1} \circ \cdots \circ \xrightarrow{a_n}$$

Sekuenca $q \xrightarrow{a_1} q_1 \quad \ldots \quad q_{n-1} \xrightarrow{a_n} q_n$ quhet sekuenca e kalimeve.

Automati I fundem jodeterminist

Ne shume aplikacione, automatet e fundem mund te thjeshtezohen nqs lejohet jodeterminizmi.
Psh. kemi relacionin e kalimit $\rightarrow \subseteq Q \times \Sigma \times Q$. Mund te ndodhi qe per ndonje $q \in Q$, $a \in \Sigma$, te kemi gjendje pasardhese te ndryshme $q_1, \ldots, q_n$, pra ne paraqitjen grafike do te kemi:



Pasi pranon karakterin **a,** automati leviz ne menyre jodeterministe nga gjendja q te njera prej gjendjeve pasardhese $q_1, \ldots, q_n$. Rast I vecante eshte kur n=0; atehere per ndonje $q \in Q$, $a \in \Sigma$ nuk ka pasardhes q^1 qe $q \xrightarrow{a} q^1$. Ne kete rast, automati ndalon dhe refuzon karakterin a.

Percaktimi : Nje automat I fundem jodeterminist (pranues), shkurtimisht AFJD, eshte nje 5-she

$B = (\Sigma, Q, \delta, q_0, F)$ ose $B = (\Sigma, Q, \rightarrow, q_0, F)$

ku Σ, Q, q_0 dhe F percaktohen si ne rastin e AFD. Funksioni kalimit $\rightarrow$ eshte $\rightarrow \subseteq Q \times \Sigma \times Q$
Kalimet shkruhen ($q \xrightarrow{a} q^1$) dhe zgjerohet te stringjet ($q \xrightarrow{w} q^1$) si te AFD.
Paraqitja grafike e diagrames se gjendjeve behet ne te njejten menyre si te AFD.

Percaktim (pranueshmeria dhe ekuivalenca):

(i)Gjuha qe pranohet nga AFJD $B = (\Sigma, Q, \rightarrow, q_0, F)$ eshteL(B) $= \{w \in \Sigma^* \mid \exists\, q \in F : q_0 \xrightarrow{w} q\}$.
(ii) Dy AFJD B_1dhe B_2 quhen ekuivalente, nese $L(B_1) = L(B_2)$.

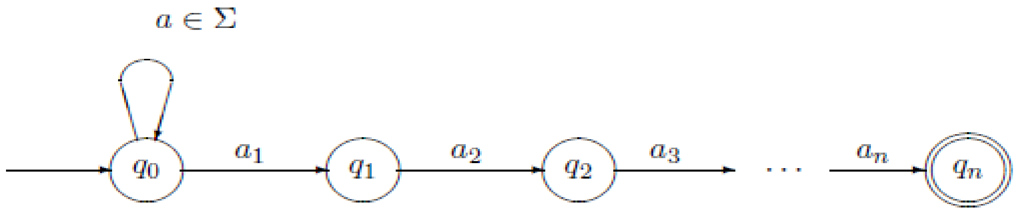
AFJD njeh stringun w, nqs duke lexuar w, ne mund te arrijme ne nje nga gjendjet finale. Mund te ndodhe qe gjate leximit te w, mund te ekzistojne sekuenca te tjera kalimesh, te cilat perfundojne ne gjendje jofinale.

Pra, AFD eshte nje rast I vecante I AFJD, keshtu qe mund te vendoset nje ekuivalence midis automateve te fundem.

Shembull (njohja e sufiksit): Kemi alfabetin Σ dhe nje stringun $v = a_1 . . . a_n \in \Sigma^*$ ku $a_i \in \Sigma$ per i= 1,. . . , n.

Ne duam te njohim gjuhen $L_V=\{wv \mid w \in \Sigma^*\}$, qe permban te gjithe stringjet mbi alfabetin Σ me sufiks v.

Do konsiderojme AFJD: $B_V= (\Sigma, \{q_0, . . . , q_n\}, \rightarrow, q_0, \{q_n\})$, ku $\rightarrow$ percaktohet nga diagrama e meposhtme e gjendjeve B_V:



B_V sillet si jodeterminist ne gjendjen fillestare q_0: gjate leximit te nje stringu, B_V mund te vendose sa here qe vjen a_1, nese do perpiqet te njohe sufiksin V. per kete, B_V shkon te q_1 dhe prêt per $a_2. . . a_n$ si sufiks.

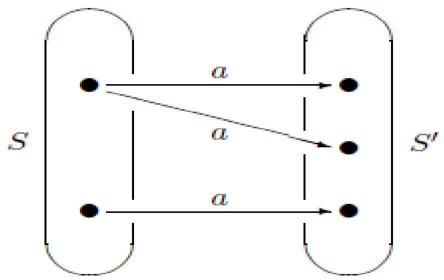
Por, a pranon AFJD me shume gjuhe sesa AFD? Pergjigjia eshte jo.

Teorema (Rabin and Scott, 1959): Per cdo AFJD ekziston nje AFD ekuivalent.

Vertetim: Kemi AFJD $B= (\Sigma, Q, \rightarrow, q_0, F)$. Ndertojme AFD A me L(A) =L(B) duke perdorur percaktimin me poshte: $A= (\Sigma, P(Q), \rightarrow_A, \{q_0\}, F_A)$, ku per $S, S_1 \subseteq Q$ dhe $a \in \Sigma$ kemi:

$S \overset{a}{\rightarrow}_A S_1$ nqs dhe vetem nqs $S_1=\{q_1 \in Q \mid \exists\, q \in S : q \overset{a}{\rightarrow} q_1\}$, $F_A=\{S \subseteq Q \mid S \cap F \neq \emptyset\}$.

Per secilen nenbashkesi S te bashkesise Q te B, ekziston nje gjendje S^1 ne A ku $S \overset{a}{\rightarrow}_A S^1$ nqs dhe vetem nqs S^1 eshte bashkesia e te gjitha gjendjeve pasardhese qe mund te arrihen me kalime me a nga AFJD B prej gjendjes S. Grafikisht:



.Veme re qe $\rightarrow_A$ eshte determinist, pra $\forall\, S \subseteq Q \quad \forall\, a \in \Sigma \quad \exists$vetem nje $S^1 \subseteq Q : S \overset{a}{\rightarrow}_A S^1$.

Per me teper, per cdo $q, q^1 \in Q$, $S, S^1 \subseteq Q$, $a \in \Sigma$ kemi:

(i) If $q \overset{a}{\rightarrow} q^1$ and $q \in S$, atehere $\exists\, S^1 \subseteq Q : S \overset{a}{\rightarrow}_A S^1$ and $q^1 \in S^1$.

(ii) If $S \overset{a}{\rightarrow}_A S1$ dhe $q^1 \in S1$, atehere $\exists\, q \in Q : q \overset{a}{\rightarrow} q^1$ dhe $q \in S$.

Pra mund te tregojme qet L(A) =L(B). Le te jete $w=a_1 . . . a_n \in \Sigma^*$ ku $a_i \in \Sigma$ per i= 1, . . . , n. Atehere:

$w \in L(B) \Leftrightarrow \exists\, q \in F: q_0 \overset{w}{\rightarrow} q \Leftrightarrow \exists\, q_1, . . . , q_n \in Q : q_0 \overset{a_1}{\rightarrow} q . . . q_{n-1} \overset{a_n}{\rightarrow} q_n$ ku $q_n \in F$

$\Leftrightarrow$ {"⇒" rrjedh nga (i) dhe "⇐"rrjedh nga (ii)} $\exists\, S_1, . . . , S_n \subseteq Q : \{q_0\} \overset{a_1}{\rightarrow}_A S_1 . . . S_{n-1} \overset{a_n}{\rightarrow}_A S_n$ ku $S_n \cap F \neq \emptyset$

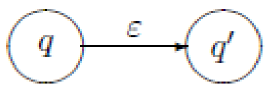
$\Leftrightarrow \exists\, S \in F_A : \{q_0\} \overset{w}{\rightarrow}_A S \Leftrightarrow w \in L(A)$.

Kalimet spontane

Automatet mund te percaktohen melehtesisht nese lejohen edhe kalimet me ε. Keto kalime, te quajtura ε-kalime, kryhen spontanisht nga automati pa lexuar ndonje symbol nga stringu hyres.

Percaktimi : Nje automat I fundem jodeterminist (pranues) me ε-kalime, shkurtimisht ε-AFJD, eshte nje 5-she $B= (\Sigma, Q, \delta, q_0, F)$ ose $B = (\Sigma, Q, \rightarrow, q_0, F)$ ku Σ, Q, q_0 dhe F percaktohen si ne rastin e AFJD ose AFD. Funksioni kalimit $\rightarrow$ eshte $\rightarrow \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$

Kalimi $q \overset{\varepsilon}{\rightarrow} q^1$ quhet ε-kalim dhe paraqitet ne diagram si me poshte:



Per te percaktuar pranueshmerine e ε-AFJD, duhet percaktojme nje zgjerim te relacionit te kalimit $q \overset{w}{\Rightarrow} q^1$.

Verrejtje:

- $\forall \alpha \in \Sigma \cup \{\varepsilon\}$, percaktohet relacioni dysh $\overset{\alpha}{\rightarrow} \subseteq Q \times Q$

  $\forall q, q^1 \in Q$: $q \overset{\alpha}{\rightarrow} q^1 \Leftrightarrow (q, \alpha, q^1) \in \rightarrow$

  $\overset{\alpha}{\rightarrow}$ quhet relacioni α-kalim
- Per keto relacione kalimi, mund te perdorim kompozimin $\circ$. $\forall \alpha, \beta \in \Sigma \cup \{\varepsilon\}$ $\overset{\alpha}{\rightarrow} \circ \overset{\beta}{\rightarrow}$ percaktohet:
- Per $q, q^1 \in Q$: $q \overset{\alpha}{\rightarrow} \circ \overset{\beta}{\rightarrow} q^1 \Leftrightarrow \exists\, q^{11} \in Q : q \overset{\alpha}{\rightarrow} q^{11}$ dhe $q^{11} \overset{\beta}{\rightarrow} q^1$

Percaktim : Lete jete $B = (\Sigma, Q, \rightarrow, q_0, F)$ nje ε-AFJD. Relacioni dysh $\overset{w}{\Rightarrow} \subseteq Q \times Q$ percaktohet induktivisht per cdo string $w \in \Sigma^*$.

- $q \overset{\varepsilon}{\Rightarrow} q^1$, nqs $\exists\, n \geq 0 : q \overset{\varepsilon}{\rightarrow} \circ \cdots \circ \overset{\varepsilon}{\rightarrow} q^1$
- $q \overset{aw}{\Longrightarrow} q^1$, nqs $q \overset{\varepsilon}{\Rightarrow} \circ \overset{a}{\rightarrow} \circ \overset{w}{\Rightarrow} q^1$,

ku $q, q^1 \in Q$, $a \in \Sigma$ and $w \in \Sigma^*$..

Verrejtje: $\forall q, q^1 \in Q$ and $a_1, \ldots, a_n \in \Sigma$

i. $q \overset{\varepsilon}{\Rightarrow} q$, por $q \overset{\varepsilon}{\Rightarrow} q^1$ nuk sjell qe $q = q^1$

ii. $q \overset{a1 \ldots an}{\Longrightarrow} q^1 \Leftrightarrow q \overset{\varepsilon}{\Rightarrow} \circ \overset{a1}{\rightarrow} \circ \overset{\varepsilon}{\Rightarrow} \ldots \circ \overset{\varepsilon}{\Rightarrow} \circ \overset{an}{\rightarrow} \circ \overset{\varepsilon}{\Rightarrow} \circ\, q^1 \Leftrightarrow q \overset{a1}{\Rightarrow} \circ \ldots \circ \overset{an}{\Rightarrow} q^1$

iii. Eshte i vendimtar, nese $q \overset{\varepsilon}{\Rightarrow} q^1$ eshte I vertete per $q, q^1 \in Q$

Percaktim (pranueshmeria dhe ekuivalenca):

(i)Gjuha qe pranohet nga ε -AFJD $B= (\Sigma, Q, \rightarrow, q_0, F)$ eshte $L(B) =\{w \in \Sigma^* \mid \exists\, q \in F : q_0 \overset{w}{\Rightarrow} q\}$.

(ii) Dy ε -AFJD B_1dhe B_2 quhen ekuivalente, nese $L(B_1) =L(B_2)$.

Pra AFJD eshte nje rast i vecante i ε-AFJD. Kjo na lejon te percaktojem ekuivalence midis AFJD dhe ε-AFJD.

Teorema: Per cdo ε-AFJD ekziston nje AFJD ekuivalent.

Vertetim: Kemi ε-AFJD B= (Σ, Q, $\rightarrow$, q_0, F). Ndertojme AFJD A= (Σ, Q, $\rightarrow_A$, q_0, F_A), ku per q, $q^1 \in Q \land a \in \Sigma$:

- $q \overset{a}{\rightarrow}_A q^1 \Leftrightarrow q \overset{a}{\Rightarrow} q^1$ ne B, pra $\overset{\varepsilon}{\Rightarrow} \circ \overset{a}{\rightarrow} \circ \overset{\varepsilon}{\Rightarrow} q^1$ ne B
- $q \in F_A \Leftrightarrow \exists q^1 \in F : q \overset{\varepsilon}{\Rightarrow} q^1$

Vihet re qe A eshte nje AFJD pa ε-kalime ku $F \subseteq F_A$. Mbetet te tregojem qe L(A) =L(B).

Le te jete $w=a_1 \dots a_n \in \Sigma^*$ ku $n \geq 0$ dhe $a_i \in \Sigma$ per i= 1, . . . , n. Atehere:

$$
\begin{aligned}
w \in L(A) &\Leftrightarrow \exists\, q \in F_A : q_0 \overset{w}{\rightarrow}_A q \\
&\Leftrightarrow \exists\, q \in F_A : q_0 \overset{a1}{\rightarrow}_A \circ \dots \circ \overset{an}{\rightarrow}_A q \\
&\Leftrightarrow \exists\, q \in F_A : q_0 \overset{a1}{\Rightarrow} \circ \dots \circ \overset{an}{\Rightarrow} q \\
&\Leftrightarrow \{\text{“}\Rightarrow\text{”: zgjidh } q^1 \text{ ku } q \overset{\varepsilon}{\Rightarrow} q^1 \\
&\text{“}\Leftarrow\text{”:zgjidh } q=q^1.\} \\
&\quad \exists\, q^1 \in F: \quad q \overset{a1}{\Rightarrow} \circ \dots \circ \overset{an}{\Rightarrow} q^1 \\
&\Leftrightarrow \exists\, q^1 \in F: \quad q \overset{w}{\Rightarrow} q^1 \\
&\Leftrightarrow \quad w \in L(B)
\end{aligned}
$$

Ne rastin kur w=ε:

$$
\begin{aligned}
\varepsilon \in L(A) &\Leftrightarrow \exists\, q_0 \in F_A \\
&\Leftrightarrow \exists\, q \in F : q_0 \overset{\varepsilon}{\Rightarrow} q \\
&\Leftrightarrow \varepsilon \in L(B)
\end{aligned}
$$

Verrejtje:

AFJD A mund te krijohet nga nje ε- AFJD B, sepse relacioni $q \overset{\varepsilon}{\Rightarrow} q^1$ eshte vendimtar

Nqs kemi ε-cikle te ε- AFJD B, atehere bashkesia e gjendjeve te AFJD A mund te reduktohet. Cdo ε-cikel mund te zevendesohet nga nje gjendje.

Pra arrijme ne konkluzionin e meposhtem:

AFD = AFJD = ε- AFJD

Si rrjedhoje, klasat e gjuheve, te cilat pranohen nga AFD, AFJD dhe ε- AFJD, perputhen me njera tjetren.Ne do te marrim klasen e gjuhëve te pranuara ne menyre te fundme, dhe do te tregojme vecorite e kesaj klase. Per kete do te perdorim automate te ndryshme ne raste te ndryshme.

§2 Veçoritë e mbylljes

Më poshtë tregohen veprimet e mbyllura në klasën gjuhëve të pranuara në mënyrë të fundme. Fillimish do të paraqiten veprimet mbi bashkësitë: bashkimi, prerja, komplementi, diferenca, si dhe veprimet mbi gjuhët: ngjitja dhe iteracioni (operatori i Kleene).

2.1 **Teoremë** : Klasa e gjuhëve të pranuara në mënyrë të fundme është e mbyllur nën veprimet më poshtë::

1. bashkimi,
2. komplementi,

3. prerja,
4. diferenca,
5. ngjitja,
6. iteracioni.

Vërtetim : Le të jetë $L_1, L_2 \subseteq \Sigma^*$ gjuhë të pranuara në mënyrë të fundme. Do të kemi AFD $A_i = (\Sigma, Q_i, \rightarrow_i, q_{0i}, F_i)$ ku $L_i = L(A_i)$, $i = 1, 2$, dhe $Q_1 \cap Q_2 = \emptyset$. Do tregojmë që $L_1 \cup L_2$, $\overline{L_1}$, $L_1 \cap L_2$, $L_1 \setminus L_2$, $L_1 \cdot L_2$ dhe L^* janë të pranuara në mënyrë të fundme.

1. $L_1 \cup L_2$: Ndërtojmë ε-AFjD $B = (\Sigma, \{q_0\} \cup Q_1 \cup Q_2, \rightarrow, q_0, F_1 \cup F_2)$, ku $q_0 \notin Q_1 \cup Q_2$ dhe
$$\rightarrow = \{(q_0, \varepsilon, q_{01}), (q_0, \varepsilon, q_{02})\} \cup \rightarrow_1 \cup \rightarrow_2$$
B paraqitet grafikisht si më poshtë:

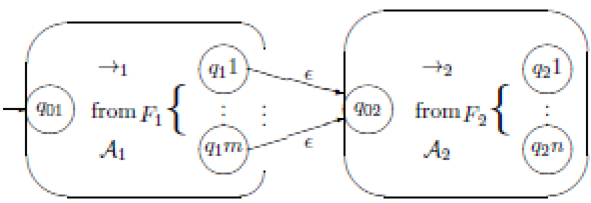
Duket qartë që $L(B) = L_1 \cup L_2$ qendron.

2. $\overline{L_1}$: Konsiderojmë AFD $A = (\Sigma, Q_1, \rightarrow_1, q_{01}, Q_1 \setminus F_1)$. Atëherë për të gjitha $w \in \Sigma^*$ kemi:

$$\begin{aligned} w \in L(A) &\Leftrightarrow \exists\, q \in Q_1 \setminus F_1 : q_{01} \xrightarrow{w}_1 q \\ &\{\rightarrow_1 \text{ determ.}\} \\ &\Leftrightarrow \neg \exists\, q \in F_1 : q_{01} \xrightarrow{w}_1 q \\ &\Leftrightarrow w \notin L(A_1) \\ &\Leftrightarrow w \notin L_1 \end{aligned}$$

Pra kemi $L(A) = \overline{L_1}$.

3. $L_1 \cap L_2$: Rrjedh prej $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$.
4. $L_1 \setminus L_2$: Rrjedh prej $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$.
5. $L_1 \cdot L_2$: Ndërtojmë ε-AFjD $B = (\Sigma, Q_1 \cup Q_2, \rightarrow, q_{01}, F_2)$ me
$$\rightarrow = \rightarrow_1 \cup \{(q, \varepsilon, q_{02}) \mid q \in F_1)\} \cup \rightarrow_2 .$$
B paraqitet grafikisht më poshtë:

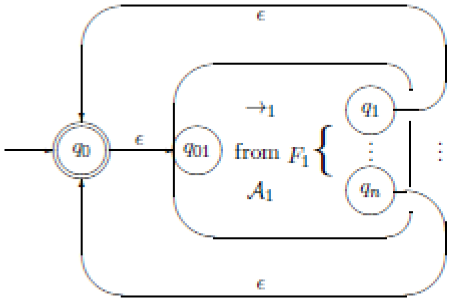


$L(B) = L_1 \cdot L_2$ mund të tregohet lehtësisht.

6. L_1^*: Për iteracionin ndërtojmë ε-AFjD
   $B = (\Sigma, \{q_0\} \cup Q_1, \rightarrow, q_0, \{q_0\})$, ku $q_0 \notin Q_1$ dhe

   $$\rightarrow = \{(q_0, \varepsilon, q_{01})\} \cup \rightarrow_1 \cup \{(q, \varepsilon, q_0) \mid q \in F_1\}$$

   B paraqitet grafikisht më poshtë:



   Lehtësisht mund të tregojme që $L(B) \mathbin{\overline{\top}} L^*$.

Shënim : AFD pranuese për bashkimin dhe prerjen mund të ndërtohen dhe në mënyrën e paraqitur më poshtë. Kemi A_1 dhe A_2 si më sipër. Shikojmë relacionin $\rightarrow \subseteq Q \times \Sigma \times Q$ në prodhimin kartezian $Q = Q_1 \times Q_2$:

Për të gjitha $q_1, q' \in Q_1$ dhe q_2, q'' ,Q_2 dhe $a \in \Sigma$ kemi

$(q_1, q_2) \xrightarrow{a} (q_1', q_2')$ if and only if $q_1 \xrightarrow{a}_1 q_1'$ and $q_2 \xrightarrow{a}_2 q_2'$.

Relacioni $\xrightarrow{a}$ tregon progresin e njëkohshëm në paralel të automateve A_1 dhe A_2 ndërsa lexojnë karakterin a.

Kemi AFD

$$A_\cup = (\Sigma, Q, \rightarrow, (q_{01}, q_{02}), F_\cup),$$
$$A_\cap = (\Sigma, Q, \rightarrow, (q_{01}, q_{02}), F_\cap)$$

me

$$F_\cup = \{(q_1, q_2) \in Q \mid q_1 \in F_1 \text{ ose } q_2 \in F_2\},$$
$$F_\cap = \{(q_1, q_2) \in Q \mid q_1 \in F_1 \text{ dhe } q_2 \in F_2\}.$$

Atëherë do kemi që $L(A_\cup) = L_1 \cup L_2$ dhe $L(A_\cap) = L_1 \cap L_2$.

Vërtetim : Tregojmë që pohimi për $A_\cap$ qëndron. Për çdo $w \in \Sigma^*$ kemi:

$$\begin{aligned} w \in L(A_\cap) &\Leftrightarrow \exists\ (q_1, q_2) \in F_\cap :\ (q_{01}, q_{02}) \xrightarrow{w} (q_1, q_2) \\ &\Leftrightarrow \exists\ q_1 \in F_1, q_2 \in F_2 :\ q_{01} \xrightarrow{w}_1 q_1 \text{ dhe } q_{02} \xrightarrow{w}_2 q_2 \\ &\Leftrightarrow w \in L(A_1) \text{ dhe } w \in L(A_2) \\ &\Leftrightarrow w \in L_1 \cap L_2 \end{aligned}$$

D

§3 Shprehjet e rregullta

Gjuhët e pranuara në mënyrë të fundme mund të përshkruhen induktivisht me anë të shprehjeve të rregullta. Për këtë qëllim përdorim alfabetin Σ e dhënë më sipër.

3.1 **Përcaktime (shprehjet e rregullta dhe gjuhët)**:

1. Sintaksa e shprehjeve të rregullta mbi Σ jepet si më poshtë:
   - $\emptyset$ dhe ε janë shprehje të rregullta mbi Σ.
   - a është shprehje e rregullt mbi Σ për çdo $a \in \Sigma$.
   - Nqs re, re_1, re_2 janë shprehje të rregullta mbi Σ, atëherë $(re_1 + re_2)$, $(re_1 \cdot re_2)$, re^* janë gjithashtu shprehje të rregullta mbi Σ.
2. *Semantka e shprehjes së rregulla re mbi* Σësht[gjuha $L(re) \subseteq \Sigma^*$, e cila përcaktohet induktivisht si më poshtë:
   - $L(\emptyset) = \emptyset$
   - $L(\varepsilon) = \{\varepsilon\}$
   - $L(a) = \{a\}$ për $a \in \Sigma$
   - $L((re_1 + re_2)) = L(re_1) \cup L(re_2)$
   - $L((re_1 \cdot re_2)) = L(re_1) \cdot L(re_2)$
   - $L(re^*) = L(re)^*$
3. Një gjuhë $L \subseteq \Sigma^*$ quhet e rregullt, nqs kemi një shprehje të rregullt re mbi Σ ku $L = L(re)$.

Më poshtë është fortësia e operatorëve:

$$* \text{ më i fortë se } \cdot \text{ dhe } \cdot \text{ më i fortë se } +.$$

Shembull : Do përdorim shprehjet e rregullta për të përshkruar disa gjuhë të diskutuar më parë.

1. Gjuha e identifikuesve të konsideruar nga analiza leksikore, mund të përshkruhet nga një shprehje e rregullt
$$re_1 = (a + \ldots + Z)(a + \ldots + Z + 0 + \ldots + 9)^*.$$
2. Gjuha mbi $\{b_1, e_1, b_2, e_2\}$ e përdorur për sinkronizimin e dy programeve, që ndajnë një printer, përshkruhet nga shprehja e rregullt:
$$re_2 = (b_1e_1 + b_2e_2)^*(\varepsilon + b_1 + b_2).$$
3. Le të jetë $\Sigma = \{a_1, \ldots\ a_n, b_1, \ldots, b_m\}$ dhe $v = a_1 \ldots a_n$. Atëherë gjuha $L_v = \{wv \mid w \in \Sigma^*\}$ e stringjeve me prapashtesë v përshkruhet nga shprehja e rregullt

$$re_3 = (a_1 + \ldots a_n + b_1 + \ldots + b_m)^* a_1 \ldots a_n.$$

Vini re: $L(re_3) = L_v$.

3.2 **Teoremë (**Kleene**)**: Një gjuhë është e rregullt nqs dhe vetëm nqs është e pranuar në mënyrë të fundme.

Vërtetim : Marrim një gjuhë $L \subseteq \Sigma^*$.

"⇒": Për shprehjen e rregulllt re mbi Σ kemi që $L = L(re)$. Tregojmë me anë të induksionit që $L(re)$ është e pranuar në mënyrë të fundme.

Baza: $L(\emptyset)$, $L(\varepsilon)$ dhe $L(a)$ janë dukshëm të pranuara në mënyrë të fundme për $a \in \Sigma$. Pse?

Hapi i induksionit: Le të jetë $L(re)$, $L(re_1)$ dhe $L(re_2)$ të pranuara në mënyrë të fundme. Atëherë $L(re_1+re_2)$, $L(re_1 \cdot re_2)$ dhe $L(re^*)$ janë gjithashtu të pranuara në mënyrë të fundme, sepse klasa e gjuhëve të pranuara në mënyrë të fundme është e mbyllur në lidhje me veprimet e bashkimit, ngjitjes dhe iteracionit.

"⇐": Kemi $L = L(A)$ për një AFD A me n gjendje. Kemi $A = (\Sigma, Q, \rightarrow, 1, F)$, ku $Q = \{1, \ldots, n\}$. Për $i, j \in \{1, \ldots, n\}$ dhe $k \in \{0, 1, \ldots, n\}$ përcaktojmë

$L^k_{ij} = \{w \in \Sigma$ | i $\xrightarrow{w} j$ and $\forall\ u \in \Sigma^*$, $\forall\ l \in Q$:

nga ($\exists\ v : v \neq \varepsilon,\ v \neq w$ dhe $uv = w$) dhe i $\xrightarrow{u}$ l rrjedh $l \leq k\}$.

$L^k_{i,j}$ përbëhet nga të gjithë stringjet w me të cilët automati A lëviz nga gjendja i në gjendjen j, ndërkohë që lexon stringun w. Për më tepër, gjatë leximit të stringut w, kalohet vetëm në gjendjet me vlerë më të vogël ose të barabartë me k.

Do tregojmë me induksion mbi k që gjuhët L^k_{ij} janë të gjitha të rregullta.

k=0 Te stringjet L^0_{ij} nuk kemi kalim në gjendje të ndërmjetme. Prandaj

$$L^0_{ij} = \begin{cases} \{a \in \Sigma \mid i \xrightarrow{a} j\},\ i \neq j \\ \{\varepsilon\} \cup \{a \in \Sigma \mid i \xrightarrow{a} j\},\ i = j \end{cases}$$

Pra L^0_{ij} është e rregullt si një nënbashkësi e fundme e $\{\varepsilon\} \cup \Sigma$

k → k+1 Supozojmë që L^k_{ij} janë të rregullta për çdo $i,j \in \{1,\ldots,n\}$. Pra për çdo $i,j \in \{1,\ldots,n\}$ do të kemi

$$L^{k+1}_{i,j} = L^k_{i,j} \cup L^k_{i,k+1}(L^k_{k+1,k+1})^* L^k_{k+1,j}$$

Për të vajtur nga gjendja i në gjendjen j kemi dy mundësi:

a) gjendja e ndërmjetme $k+1$ është ose e panevojshme, ku mjaftohemi me $L^k_{i,j}$
b) Gjendja k+1 përdoret si gjendje e ndërmjetme të paktën një herë, ku përdorim $L^k_{i,k+1}(L^k_{k+1,k+1})^* L^k_{k+1,j}$

Me induksion mbi k nxjerrim që $L^{k+1}_{i,j}$ është e rregullt.

Nga rregullsia e $L^k_{i,j}$, nxjerrim që vetë gjuha L është e rregullt, pasi

$$L = L(A) = \bigcup_{j \in F} L^n_{1,j}$$

§4 Veçoritë strukturore të gjuhëve të rregullta

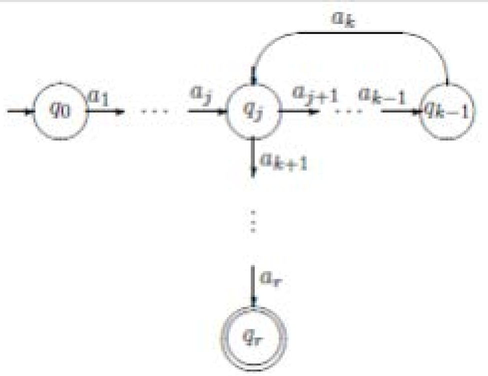
Le të jetë Σ një alfabet i çfarëdoshëm.

4.1 **Teoremë (lema kërcyese e gjuhëve të rregullta)**: Për çdo gjuhë të rregullt $L \subseteq \Sigma^*$ ekziston një numër $n \in$ N, i tillë që për të gjithë stringjet $z \in L$ me $|z| \geq n$ do kemi nje dekompozim të $z = uvw$ me $v \neq \varepsilon$ dhe $|uv| \leq n$ dhe për të gjithë $i \in$ N rrjedh $uv^iw \in L$,
p. sh., mund të përsërisim nënstringun *v* dhe stringu i përftuar do jetë fjalë e gjuhës L.
$\forall\ L \subseteq \Sigma^*$ të rregullt $\exists\ n \in$ N $\forall\ z \in L$ me $|z| \geq n$
$\exists\ u, v, w \in \Sigma^*$: $z = uvw$ and $v \neq \varepsilon$ and $|uv| \leq n$ and $\forall\ i \in$ N : $uv^iw \in L$

Vërtetim: Le të jetë $L \subseteq \Sigma^*$ e rregullt.

Sipas teoremës së Kleene ekziston një AFD $A = (\Sigma, Q, \rightarrow, q_0, F)$ me $L = L(A)$. Marrim $n = |Q|$ dhe shqyrtojmë një string $z \in L$ me $|z| \geq n$. Vihet re se A do të kalojë dy herë te e njëjta gjendje të paktën një herë gjatë leximit z. Më saktësisht:

Marrim $z = a_1 \ldots a_r$ me $r = |z| \geq n$ dhe $a_i \in \Sigma$ për $i = 1, \ldots, r$ dhe le të jenë $q_1, \ldots, q_r \in Q$ përcaktohen $q_{i-1} \xrightarrow{a_i} q_i$ *për i=1,...,r. Pra do kemi j, k* $\in \{1, \ldots, n\}$ *me* $0 \leq j < k \leq n \leq r$, *e tillë që* $q_j = q_k$



Le të jetë : $u = a_1 \ldots a_j$, $v = a_{j+1} \ldots a_k$, $w = a_{k+1} \ldots a_r$. Bazuar te veçoritë e j dhe k, nxjerrim që $v \neq \varepsilon$ and $|uv| \leq n$. Gjithashtu duket qartë që automati A, mund të bëjë cikle q_j një numër të pacaktuar herësh, p. sh., për gjithë $i \in$ N kemi që: $uv^iw \in L$.

Një aplikim praktik i lemës kërcyese lidhet me tregimin që një gjuhë nuk është e rregullt.

Shembull: Gjuha $L = \{a^nb^n \mid n \in \text{N}\}$ nuk është e rregullt. E vërtetojmë me metodën e supozimit nga e kundërta.

Hipoteza: L e rregullt. Sipas lemës kërcyese, do kemi $n \in$ N me veçoritë e treguara më sipër. Marrim $z = a^nb^n$. Mqs $|z| \geq n$, ndajmë z *në* $z = uvw$ ku $v \neq \varepsilon$ dhe $|uv| \leq n$, e tillë që për çdo $i \in$ N kemi: $uv^iw \in L$. Por v përbëhet vetëm nga karaktere a, prandaj kjo sjell që $uw = a^{n-|v|}b^n \in L$ duhet të jetë e vërtetë. Kontradiksion

Shembulli i mësipërm tregon që automatet e fundëm nuk mund të numërojnë pafundësisht.

Relacioni i Nerode

Relacioni i Nerode është i rëndësishëm për përcaktimin e rregullsisë së gjuhëve $L \subseteq \Sigma^*$.

4.2 **Përcaktime**:

Kemi $L \subseteq \Sigma^*$. Relacioni i Nerode mbi L është një relacion dysh $\equiv_L$ mbi Σ^*,

pra $\equiv_L \subseteq \Sigma^* \times \Sigma^*$, dhe për $u, v \in \Sigma^*$ përcaktohet si më poshtë:

$u \equiv_L v$ nqs dhe vetëm nqs për të gjithë $w \in \Sigma^*$ kemi që: $uw \in L \Leftrightarrow vw \in L$.

Pra do themi që $u \equiv_L v$, atëherë kur u dhe v mund të kthehen njëkohësisht në fjalë të L. Në veçanti, (kur $w = \varepsilon$) $u \in L \Leftrightarrow v \in L$.

Shësnim : Relacioni i Nerode është kongruent nga e djathta:

1. $\equiv_L$ është relacion ekuivalence mbi Σ^*, pra refleksiv, simetrik dhe kalimtar.
2. $\boldsymbol{u \equiv_L v \Rightarrow uw \equiv_L vw}$ *për të gjithë* $w \in \Sigma^*$

Mqs $\equiv_L$ është relacion ekuivalence, mund të përcaktojmë numrin e klasave të ekuivalencës së $\equiv_L$ dhe e quajmë index($\equiv_L$).

4.3 **Teorem (**Myhill dhe Nerode**)**:

Gjuha $L \subseteq \Sigma^*$ është e rregullt atëherë dhe vetëm atëherë kur $\equiv_L \subseteq \Sigma^* \times \Sigma^*$ ka indeks të fundem.

Vërtetim : "$\Rightarrow$": Marrim L të rregullt, pra do kemi $L = L(A)$ për AFD $A = (\Sigma, Q, \rightarrow, q_0, F)$.
Krijojmë relacionin dysh $\equiv_A$ në Σ^*. Për $u, v \in \Sigma^*$ kemi që:

$u \equiv_A v$ atëherë dhe vetëm atëherë $\exists q \in Q: q_0 \xrightarrow{u} q \wedge q_0 \xrightarrow{v} q$

pra automati A lëviz nga q_0 te e njëjta gjendje q pasi të ketë lexuar inputet u dhe v. Vini re që q përcaktohet në në mënyrë të vetme pasi A është determinist. Relacioni $\equiv_A$ është relacion ekuivalence mbi Σ^*.

Do tregojmë që $\equiv_A$ është një pasurim i $\equiv_L$, pra për gjithë $u, v \in \Sigma^*$ kemi që

$$u \equiv_A v \Rightarrow u \equiv_L v.$$

Kemi $u \equiv_A v$ dhe $w \in \Sigma^*$. Mund të themi që:

$$\begin{aligned} uw \in L &\Leftrightarrow \exists\, q \in Q\ \exists\, q' \in F : q_0 \xrightarrow{u} q \xrightarrow{w} q' \\ &\Leftrightarrow \{u \equiv_A v\}\ \exists\, q \in Q, q' \in F : q_0 \xrightarrow{v} q \xrightarrow{w} q' \\ &\Leftrightarrow vw \in L \end{aligned}$$

Pra numri i klasave të ekuivalencës të $\equiv_A$ janë të paktën sa indeks($\equiv_L$). Si rrjedhojë

$$\begin{aligned} & Index(\equiv_L) \\ \le\ & Index(\equiv_A) \\ =\ & \text{numri i gjendjeve të kapshme nga } q_0 \\ \le\ & |Q|, \end{aligned}$$

pra $\equiv_L$ ka indeks të fundëm.

"$\Leftarrow$": Kemi $L \subseteq \Sigma^*$ dhe $k \in \mathbb{N}$ është index($\equiv_L$). Zgjedhim k stringje $u_1, \dots, u_k \in \Sigma^*$, ku $u_1 = \varepsilon$, si përfaqësues të klasave të ekuivalencës të $\equiv_L$. Kësisoj Σ^* mund të paraqitet si një bashkim i i klasave të ekuivalencës, të cilat nuk kanë asnjë prerje me njëra tjetrën:

$$\Sigma^* = [u_1] \dot\cup \dots \dot\cup [u_k].$$

Në veçanti, $\forall$ string $u \in \Sigma^*$ do të kemi një $i \in \{1, \dots, k\}$ ku $[u] = [u_i]$.

Ndërtojmë automatin e klasave të ekuivalencës si më poshtë

$A_L = (\Sigma, Q_L, \to_L, q_L, F_L)$:

$$\begin{aligned} Q_L &= \{[u_1], \dots, [u_k]\}, \\ q_L &= [u_1] = [\varepsilon] \\ F_L &= \{[u_j] \mid u_j \in L\}, \end{aligned}$$

and for $i, j \in \{1, \dots, k\}$ and $a \in \Sigma$ let

$$[u_i] \xrightarrow{a}_L [u_j] \Leftrightarrow [u_j] = [u_i a]$$

A_L është AFD dhe për të gjithë stringjet $\in \Sigma^*$ kemi:

$$[\varepsilon] \xrightarrow{w}_L [u_j] \Leftrightarrow [u_j] = [w]$$

dhe më saktë për $w = a_1 \dots a_n$

$$[\varepsilon] \xrightarrow{w}_L [u_j] \Leftrightarrow [\varepsilon] \xrightarrow{a_1}_L [a_1] \dots \xrightarrow{a_n}_L [a_1 \dots a_n] = [u_j]$$

prandaj

$$\begin{aligned} w \in L(A_L) &\Leftrightarrow \exists\, [u_j] \in F_L : [\varepsilon] \xrightarrow{w}_L [u_j] \\ &\Leftrightarrow \exists\, u_j \in L : [u_j] = [w] \\ &\Leftrightarrow w \in L \end{aligned}$$

Pra A_L pranon gjuhën L. Pra L është e rregullt.

Për të minimizuar numrin e gjendjeve të automatit, përdorim automatin e klasave të ekuivalencës të përdorur në vërtetimin e teoremës së Myhill-Nerode, A_L.

4.4 **Shtojcë**: Kemi $L \subseteq \Sigma^*$ të rregullt dhe $k = Index(\equiv_L)$. Çdo AFD që pranon L ka të paktën k gjendje. Numri më i vogël k arrihet nga AFD A_L.

Automati A_L është një prototip i të gjithë AFD që pranojnë L me numër minimal gjendjesh k. Mund të tregojmë që çdo AFD tjetër që pranon L dhe ka k gjendje është izomorf me A_L, pra mund të përftohet nga A_L përmes riemërtimit bijektiv të gjendjeve.

4.5 **Definition** : Dy AFD ose AFjD $A_i = (\Sigma, Q_i, \to_i, q_{0i}, F_i)$, $i = 1, 2$, quhen izomorfë nqs kemi një bijeksion $\beta : Q_1 \to Q_2$ me veçoritë e mëposhtme:

- $\beta(q_{01}) = q_{02}$,
- $\beta(F_1) = \{\beta(q) \mid q \in F_1\} = F_2$, $\forall\, q, q' \in Q_1\ \forall\, a \in \Sigma : q \xrightarrow{a}_1 q' \Leftrightarrow \beta(q) \xrightarrow{a}_2 \beta(q')$.

Bijeksioni β quhet izomorfizëm nga A_1 te A_2.

Vini re që izomorfizmi është një relacion ekuivalence në automatët e fundëm.

4.6 **Teoremë** : Kemi $L \subseteq \Sigma^*$ e regullt dhe $k = Index(\equiv_L)$. Atëherë çdo AFD A që pranon L dhe ka k gjendje është izomorfe me A_L.

Vërtetim : Kemi $A = (\Sigma, Q, \rightarrow, q_1, F)$ me $L(A) = L$ and $|Q| = k$ dhe autometin e klasave të ekuivalencës prej teoremës Myhill-Nerode me $Q_L = \{[u_1], \ldots, [u_k]\}$ dhe $u_1 = \varepsilon$.

$\forall$ i $\in$ {1,...,k} përcaktojmë q_i me anë të kalimit $q_1 \xrightarrow{u_i} q_i$.

q_i përcaktohet në mënyrë të vetme, pasi kalimi është determinist $\rightarrow$.

Përcaktojmë lidhjen $\beta : Q_L \rightarrow Q$ me anë të $\beta([u_i]) = q_i$ për $i \in 1, \ldots, k$ dhe tregojmë që β është izomorfizëm nga A_L te A.

---

1. β është injektiv: Marrim $q_i = q_j$. Atëherë $q_1 \xrightarrow{u_i} q_i$ dhe $q_1 \xrightarrow{u_j} q_j$

   Prandaj $w \in \Sigma^* : u_i w \in L \Leftrightarrow u_j w \in L$. Prandaj, $u_i \equiv_L u_j$ dhe $[u_i] = [u_j]$.
2. β është surjektiv: kjo rrjedh nga (1) dhe fakti që $k = |Q|$. Pra në veçanti $Q = \{q_1, \ldots, q_k\}$.
3. $\beta([q_L]) = \beta([u_1]) = \beta([\varepsilon]) = q_1$
4. $\beta(F_L) = F$: $[u_j] \in F_L \Leftrightarrow u_j \in L \Leftrightarrow q_j \in F$
5. Për të gjithë $i, j \in \{1, \ldots, k\}$ dhe $a \in \Sigma$ kemi:

$$[u_i] \xrightarrow{a}_L [u_j] \Leftrightarrow q_i \xrightarrow{a} q_j$$

"$\Rightarrow$": Kemi $[u_i] \xrightarrow{a}_L [u_j] \Rightarrow [u_i a] = [u_j]$. $\exists$ l $\in$ $\{1,\ldots,k\}$ ku $q_i \xrightarrow{a} q_l$. Bazuar në këto kemi:

$$\begin{tikzcd} q_1 \arrow[r, "u_i"] \arrow[rr, bend right, "u_l"'] & q_i \arrow[r, "a"] & q_l \end{tikzcd}$$

Mqs stringjet $u_i a$ dhe u_l na çojnë te e njëjta gjendje $q_l \Rightarrow u_i a \equiv_L u_l$.
Pra $[u_j] = [u_i a] = [u_l]$. Bazuar te zgjedhja ~~e~~ $u_1, \ldots, u_k$ në A_L kemi që $u_j = u_l$, dhe që $j = l$. Pra $q_i \xrightarrow{a} q_j$

"⇐": Kemi $q_i \xrightarrow{a} q_j$. Si më sipër kemi figurën:

$$\begin{tikzcd} q_1 \arrow[r, "u_i"] \arrow[rr, bend right, "u_j"'] & q_i \arrow[r, "a"] & q_j \end{tikzcd}$$

Pra dalim në konkluzionin: $u_i a \equiv_L u_j$. Pra, kemi $[u_i a] = [u_j] \Rightarrow [u_i] \xrightarrow{a}_L [u_j]$

Pra A_L dhe A janë izomorfë.

Për çdo gjuhë të rregullt $L \subseteq \Sigma^*$ me k = index($\equiv_L$) kemi një AFD, e cila është unike në nivel izomorfizmi, me një numër minimal gjendjesh të barabartë me k.

4.7 **Përcaktim**: Automati minimal për gjuhën e $L \subseteq \Sigma^*$ është AFD, e cila pranon L dhe ka numër gjendjesh të barabartë me indeksin e relacionit të Nerode $\equiv_L$. Ky automat minimal është unik në nivel izomorfizmi.

Automati minimal për gjuhën e rregullt $L \subseteq \Sigma^*$ prej çdo AFD $A = (\Sigma, Q, \rightarrow, q_0, F)$ që pranon L është i llogaritshëm. Reduktimi përfshin hapat e mëposhtëm:

1. Eliminimi i gjendjeve të pakapshme.

Një gjendje $q \in Q$ quhet e kapshme nqs ka një string $w \in \Sigma^*$ me $q_0 \xrightarrow{w} q$.
Bashkësia e gjendjeve të kapshme të Q është e llogaritshme, sepse mund të konsiderojmë vetëm stringjet w me $q_0 \xrightarrow{w} q$, *të cilat kanë gjatësi* $\leq |Q|$.

2. Kombinimi i gjendjeve ekuivalente.

Për $q \in Q$, $w \in \Sigma^*$ dhe $S \subseteq Q$ ne themi $q \xrightarrow{w} S$, nqs kemi $q' \in S$ me $q \xrightarrow{w} q'$. Dy gjendje $q_1, q_2 \in Q$ quhen *ekuivalent*, $q_1 \sim q_2$, nqs për të gjithë $w \in \Sigma^*$ kemi:

$$q_1 \xrightarrow{w} F \Leftrightarrow q_2 \xrightarrow{w} F$$

pra i njëjti string çon q_1 dhe q_2 te gjendjet finale.

§5 Probleme të zgjidhshme

Kalimet e mëposhtme janë të llogaritshme algoritmikisht:

- ε-NFA → NFA → DFA
- DFA → automati minimal
- ε-AFjD për operacionet e mëposhtme në gjuhët e pranuara në mënyrë të fundme: Bashkimi, komplementi, prerja, diferenca, ngjitja dhe iteracioni
- shprehje e rregullt → AFjD → AFD → shprehje e rregullt

Më poshtë do diskutojmë problemet e mëposhtme pëe gjuhët e rregullta.

Problemi pranueshmërisë	Jepen:	AFD A dhe një string w
	Pyetja:	$w \in L(A)$?
Problemi i të qënit bosh	Jepen:	AFD A
	Pyetja:	$L(A) = \emptyset$?
Problemi i të qënit e fundme	Jepen:	AFD A
	Pyetja:	$L(A)$ e fundme?
Problemi ekuivalencës	Jepen:	AFD A_1 dhe A_2
	Pyetja:	$L(A_1) = L(A_2)$?
Problemi përfshirjes	Jepen:	AFD A_1 dhe A_2
	Pyetja:	$L(A_1) \subseteq L(A_2)$?
Problemi i prerjes	Jepen:	AFD A_1 dhe A_2
	Pyetja:	$L(A_1) \cap L(A_2) = \emptyset$?

5.1 **Teoremë**: Për gjuhët e rregullta

- problemi i pranueshmërisë,
- problemi i të qënit bosh,
- problemi i të qënit i fundëm,
- problemi i ekuivalencës,
- problemi i përfshirjes,
- problemi i prerjes

Janë të gjithë të llogaritshëm.

Vërtetim : *Problemi i pranueshmërisë*: Aplikojmë A teostringu i dhënë w dhe vendosim nëse është arritur gjendje finale në A.

Problemi i të qënit bosh: Kemi n kufirin e lemës kërcyese te gjuha e rregullt $L(A)$. Themi që:

$L(A) = \emptyset \Leftrightarrow \neg\exists\ w \in L(A) : |w| < n\ (*)$

Vërtetimi i $(*)$: "$\Rightarrow$" është i qartë. "$\Leftarrow$" me supozim nga e kundërta: Supozojmë $L(A) \neq \emptyset$.

Mqs nuk kemi një string $w \in L(A)$ me $|w| < n$, atëherë do kemi $w \in L(A)$ me $|w| \geq n$. Nqs aplikojmë lemën kërcyese me $i = 0$, do të përftojmë $w_0 \in L(A)$ me $|w_0| < n$. Pra arritëm në kundërshtim, pra supozimi fillestar është i gabuar, pra $L(A) = \emptyset$.

Për të përcaktuar nëse $L(A) = \emptyset$ bazohemi të problemi i pranueshmërisë "$w \in L(A)$?" për çdo string mbi alfabetin e A me $|w| < n$.

Problem i të qënit i fundëm: Marrim n si mësipër. Themi që:

$L(A)$ e fundme $\Leftrightarrow \neg\exists\ w \in L(A) : n \leq |w| < 2n\ \ (**)$

Vërtetimi i $(**)$: "$\Rightarrow$": Nqs ka një string $w \in L(A)$ me $|w| \geq n$, atëherë $L(A)$ do ishte e pafundme sipas lemës kërcyese.

"$\Leftarrow$" me supozim nga e kundërta: Marrim $L(A)$ të pafundme. AAtëherë do kemi stringle me gjatësi të çfarëdoshme në $L(A)$, veçanërisht një string w me $w \geq 2n$. Duke aplikuar lemën kërcyese me $i = 0$ përftojmë një string w_0 me $n \leq |w_0| < 2n$, pasi me $i = 0$ stringu u ri shkurtohet maksimalisht me n karaktere.

Problem i ekuivalencës:

M.I Ndërtojmë një AFD A si më poshtë:

$$L(A) = (L(A_1) \cap \overline{L(A_2)}) \cup (L(A_2) \cap \overline{L(A_1)})$$

Duket që $L(A_1) = L(A_2) \Leftrightarrow L(A) = \emptyset \qquad (***)$

Pra problemi i ekuivalencës për automatin reduktohet në problemin e të qënit bosh për A. Ndërtimi i A më sipër është i vështirë

M.II

Kemi $A_i = (\Sigma, Q_i, \rightarrow_i, q_{0i}, F_i), i = 1, 2$. Kemi $Q = Q_1 \times Q_2$ me relacionin e kalimit

$\rightarrow \subseteq Q \times \Sigma \times Q$ *për çdo* $q_1, q_1' \in Q_1$ *dhe* $q_2, q_2' \in Q_2$ *dhe* $a \in \Sigma$ *kemi që:*

$(q_1,q_2) \xrightarrow{a} (q_1',q_2') \Leftrightarrow q_1 \xrightarrow{a}_1 q_1'$ *dhe* $q_2 \xrightarrow{a}_2 q_2'$. *Përcaktojmë* $A=(\Sigma, Q, \rightarrow, (q_{01},q_{02}), F)$

me $F = \{(q_1, q_2) \in Q \mid q_1 \in F_1 \Leftrightarrow q_2 \notin F_2\}$.

$$\begin{aligned}
L(A_1) = L(A_2) &\Leftrightarrow \forall\ w \in \Sigma^* : (w \in L(A_1) \Leftrightarrow w \in L(A_2)) \\
&\Leftrightarrow \forall\ w \in \Sigma^* : ((\exists\ q_1 \in F_1 : q_0 \xrightarrow{w} q_1) \Leftrightarrow (\exists\ q_2 \in F_2 : q_0 \xrightarrow{w} q_2)) \\
&\Leftrightarrow \forall\ w \in \Sigma^*\ \forall\ (q_1, q_2) \in Q : (q_{01}, q_{02}) \xrightarrow{w} (q_1, q_2) \Rightarrow (q_1, q_2) \notin F \\
&\Leftrightarrow L(A) = \emptyset
\end{aligned}$$

Problemi i përfshirjes: Ndërtojmë një AFD A,

$$L(A) = L(A_1) \cap \overline{L(A_2)}$$

$$L(A_1) \subseteq L(A_2) \Leftrightarrow L(A) = \emptyset$$

Pra problemi i përfshirjes për A_1 dhe A_2 mund të reduktohet te problemi i të qënit bosh për A.

Problemi i prerjes: Ndërtojmë nje AFD A

$$L(A) = L(A_1) \cap L(A_2),$$

Pra problemi i prerjes për A_1 dhe A_2 mund të reduktohet te problemi i të qënit bosh për A.

Kapitulli III

Gjuhët pa kontekst dhe automatët me memorje

Në kapitullin më sipër pamë përdorime të ndryshme të gjuhëve të rregullta në Informatikë (p.sh. analiza leksikore apo njohja e nënstringjeve) dhe lehtësinë e përdorimit të tyre (paraqitjen nga automatët e fundëm dhe shprehjet e rregullta, veçori të mira të veprimeve të mbyllura dhe problemeve të zgjidhshëm). Sidoqoftë këto nuk janë mjaftueshëm për përshkrimin e gjuhëve të programimit.

Vështirësia qëndron në përdorimin e pakufiyuar të kllapave të ndërfutura në gjuhët e programimitThe matter is that programming languages accept bracket structures of arbitrary nesting-depth, si p.sh.

- shprehjet aritmetike të formës $3 * (4 - (x + 1))$,
- lista të formës (CAR(CONS x y)),
- pjesë kodi të formës

```
while(b1) {
   x = e1;
   while(b1) {
      y = e2;
      z = e3;
   }
}
```

Më sipër është parë që gjuha $L = \{a^n b^n \mid n \in \mathbb{N}\}$ nuk është e rregullt

Për përshkrimin e gjuhëve të programimit dp të përdorim gjuhët pa kontekst.

§1 Gramatikat pa kontekst

1.1 **Përcaktim** : Gramatikë pa kontekst do të quhet katërshja $G = (N, T, P, S)$, me vetitë si më poshtë:

(i) N është alfabeti i simboleve jofundorë,

(ii) T është alfabeti i simboleve fundorë ku $N \cap T = \emptyset$,

(iii) $S \in N$ është simboli fillestar,

(iv) $P \subseteq N \times (N \cup T)^*$ është bashkësi e fundme rregullash

Simbolika e përdorur:

- $A, B, C, \ldots$ përdoren për simbolet jofundorë,
- $a, b, c, \ldots$ përdoren për simbolet jofundorë,
- $u, v, w, \ldots$ përdoren për stringjet me simbole fundorë e jofundorë.

Rregullat shkruhen

$(A, u) \in P$ ose

$A \to u$.

Nqs disa rregulla kanë njësoj anën e majtë, si p.sh.

$$A \to u_1, A \to u_2, \ldots, A \to u_k,$$

atëherë mund të shkruajmë shkurtimisht si një metarule unik

$$A \to u_1 \,|\, u_2 \,|\, \ldots \,|\, u_k$$

ose

$$A ::= u_1 \,|\, u_2 \,|\, \ldots \,|\, u_k. \qquad (*)$$

Pra $|$ përdoret për paraqitjen e alternativave $u_1, \ldots, u_k$.

Kur rregullat e gramatikës pa kontekst paraqiten në formën $(*)$, themi se jemi në formatin *Backus-Naur* ose shkurtimisht BNF. ky format u prezantua në 1960 nga John Backus dhe Peter Naur për përcaktimin e gjuhës ALGOL 60. Paraqitja e zgjeruar e formatit BNF, e quajtur EBNF, na lejon të bëjmë shkurtime të mëtejshme. Formati EBNF mund të përkthehet 1–1 në diagramën sintaksore, e cila u paraqit në 1970 nga Niklaus Wirth për të përcaktuar gjuhën e programimit PASCAL.

Relacioni derivimit në gramatikën pa kontekst G shënohet $\vdash_G$ në $(N \cup T)^*$:

$$x \vdash_G y \text{ nqs dhe vetëm nqs } \exists A \to u \in P \; \exists w_1, w_2 \in (N \cup T)^* :$$

$$x = w_1\, A\, w_2 \text{ d} \boxed{y} = w_1\, u\, w_2. \qquad \square$$

Me $\vdash^*_G$ shënojmë mbylljen refleksive dhe kalimtare të $\vdash_G$. x $\vdash^*_G$ y lexohet "y mund të përftohet nga x".

Gjuha e gjeneruar nga G është

$$L(G) = \{w \in T^* \mid S \vdash^*_G \text{ w'}\}$$

Dy gramatika pa kontekst G_2 quhen ekuivalente kur $L(G_1) = L(G_2)$.

1.2 **Përcaktim** : Gjuha $L \subseteq T^*$ quhet pa kontekst nëse ekziston një gramatikë G pa kontekst ku $L = L(G)$.

Shembull :

(1) Gjuha $L = \{a^n b^n \mid n \in \mathbb{N}\}$ gjenerohet nga gramatika $G_1 = (\{S\}, \{a, b\}, P_1, S)$, ku P_1:

$$S \to \varepsilon \mid aSb.$$

P.sh. $a^2b^2 \in L(G_1)$, sepse

$$S \vdash_{G_1} aSb \vdash_{G_1} aaSbb \vdash_{G_1} aabb.$$

(2) Shprehjet aritmetike me variabla a, b, c dhe operatorë $+$ dhe $*$ gjenerohen nga gramatika $G_2 = (\{S\}, \{a, b, c, +, *, (,)\}, P_2, S)$ ku P_2:

$$S \to a \mid b \mid c \mid S + S \mid S * S \mid (S).$$

P.sh. $(a + b) * c \in L(G_2)$, sepse

$$S \vdash_{G_2} S * S \vdash_{G_2} (S) * S \vdash_{G_2} (S + S) * S$$
$$\vdash_{G_2} (a + S) * S \vdash_{G_2} (a + b) * S \vdash_{G_2} (a + b) * c.$$

1.3 **Përcaktim** : Derivim nga A te w në G me gjatësi $n \geq 0$ është një sekuencë hapash derivimi

$$A = z_0 \vdash_G z_1 \vdash_G \cdots \vdash_G z_n = w. \qquad (**)$$

Ky derivim quhet derivim nga e majta, nëse zëvendësojmë elementin jofundor më në të majtë në çdo hap derivimi $z_i \vdash_G z_{i+1}$, p.sh. nqs z_i dhe z_{i+1} kanë formën

$$z_i = w_1 A w_2 \text{ dhe } z_{i+1} = w_1 u w_2, \; ku \; w_1 \in T^*.$$

Derivimi nga e djathta paraqitet në të njëjtën mënyrë (ku: $w_2 \in T^*$).

Çdo derivim mund të paraqitet grafikisht si një pemë.

1.4 **Përcaktim**: Një pemë derivimi nga A te w në G ka veçoritë e mëposhtme:

(i) Çdo nyje etiketohet me një simbol nga $N \cup T \cup \{\varepsilon\}$. Nyja etiketohet me A dhe çdo nyje e brendshme etiketohet me një simbol nga N.

(ii) Nqs një nyje e brendshme e etiketuar me B ka k nyje bij, të cilët etiketohen me simbolet $\beta_1, \ldots, \beta_k$ nga e majta në të djathtë, atëhere kemi

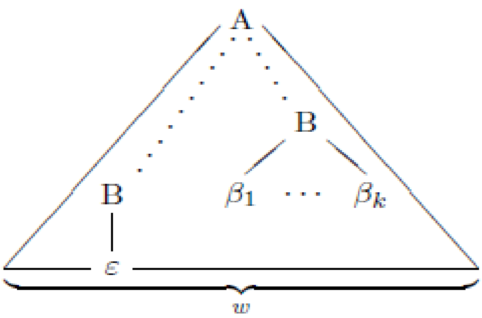
a) $k = 1$ dhe $\beta_1 = \varepsilon$ dhe $B \to \varepsilon \in P$

ose

b) $k \geq 1$ dhe $\beta_1, \ldots, \beta_k \in N \cup T$ dhe $B \to \beta_1 \ldots \beta_k \in P$.

(iii) Stringu w gjenerohet nga bashkimi i simboleve në gjethe nga e majta në të djathtë.

Ilustrim



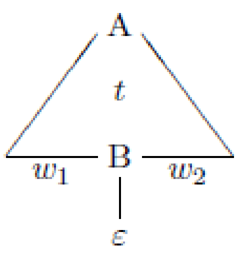
Ndërtojmë pemën e derivimit nga A te w duke u bazar te derivimi nga A te w në formën $(**)$ duke përdorur induksion mbi gjatësinë n të derivimit.

$n = 0$: Derivimi A i përket derivimit A.

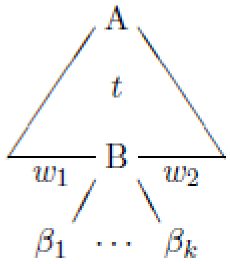
$n \to n + 1$: Kemi derivimin

$$A = z_0 \vdash_G \ldots \vdash_G z_n = w_1 B w_2 \vdash_G w_1 u w_2 = z_{n+1}.$$

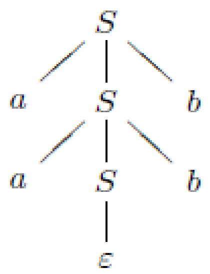
Le të jetë t pema e derivimit për derivimin $A = z_0 \vdash_G \ldots \vdash_G z_n$. Nqs $u = \varepsilon$, atëhere pema e plotë e derivimit është:



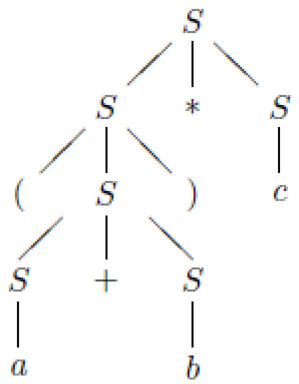
Nqs $u = \beta_1 \dots \beta_k$, ku $\beta_1, \dots, \beta_k \in N \cup T$, atëhere pema e pltë e derivimit është:



Shembull : Pemët e derivimet për shembujt e përmendur më lart



dhe



Shënim : Midis derivimeve dhe pemëve të derivimit egziston marrëdhia:

(i) $A \vdash_G^* w \quad \Leftrightarrow$ Ka pemë derivimi nga A te w në G.

(ii) Pavarësisht derivimeve të ndryshme nga A te w, vetëm një derivim nga e majta dhe vetëm një derivim nga e djathta i përkasin pemës së derivimit nga A te w.

Vërtetim:

(i) "⇒" duket qartë pasi pema ndërtohet mbi dderivimin

Për të vërtetuar "⇐" përdorin induksion mbi thellësinë e pemës.

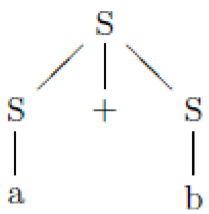
(ii) Pemët e derivimit vuajnë renditjen e zbatimit të rregullave të zevendësimit, kur simbolet jofundore veprojnë njëherësh. P.sh. të dy derivimet më poshtë

$$S \vdash_{G_2} S + S \vdash_{G_2} a + S \vdash_{G_2} a + b$$

dhe

$$S \vdash_{G_2} S + S \vdash_{G_2} S + b \vdash_{G_2} a + b$$

kanë të njëtjën pemë derivimi:



Kjo mund të shmanget nëse përdorim derivimin nga e majta ose derivimin nga e djathta.

D

Marrim rastin kur sintaksa e një gjuhe programimi GjP jepet nga nje gramatikë pa kontekst G. Kompilatori për GjP do gjenerojë për çdo program një pemë derivimi në G gjatë fazës së analizës leksikore. kompilatori gjeneron kodin e makinës për programin duke u bazuar të pema e derivimit.

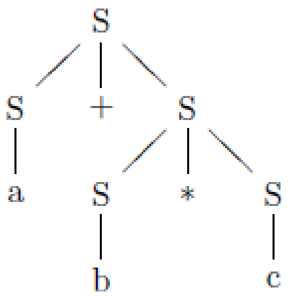
Për përdorimin e GjP është e rëndësishme që secili program i GjP të ketë një semantikë të qartë. Prandaj për çdo program të GjP duhet të kemë vetëm një pemë derivimi.

1.5 **Përcaktim**:

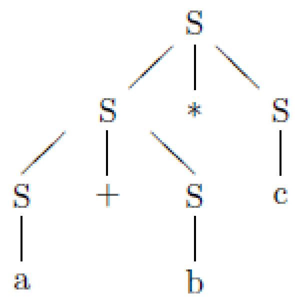
(i) Një gramatikë pa kontekst $G = (N, T, P, S)$ quhet e qartë, nqs për çdo string $w \in T^*$ kemi të shumtëm një pemë derivimi ose pemë derivimi nga e majta nga S te w në G. Përndryshe G quhet e paqartë.

(ii) Një gjuhë pa kontekst $L \subseteq T^*$ quhet e qartë, nqs egziston një gramatikë pa kontekst e qartë e tillë që $L = L(G)$. Përndryshe L quhet e paqartë.

Shembull : Gramatika G_2 për shprehjet aritmetike është e paqartë, pasi për stringun $a + b * c \in L(G_2)$ egzistojnë dy pemë derivimi:



dhe



Këto i përkasin dy shprehjeve të ndryshme nëse përdorim kllapat $a + (b * c)$ dhe $(a + b) * c$.

Për këtë arsye, në gjuhët e programimit, përdorim gramatikën G_3 për shprehjet aritmetike me rregullat si më poshtë:

- Vlerësimi bëhet ngas e majta në të djathtë. Pra $a + b + c$ vlerësohet sikur $(a + b) + c$.
- Shumëzimi * ka prioritet më të lartë se +. Prandaj, $a + b * c$ vlerësohet sikur $a + (b * c)$.

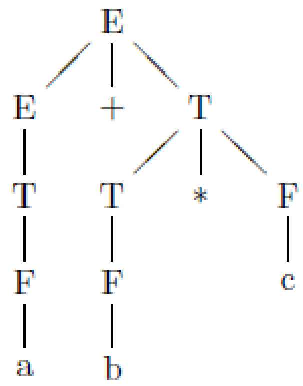
Nëse duan një sekuencë tjetër vlerësimi, duhet të përdorim kllapat (dhe).

Vini re $G_3 = (\{E, T, F\}, \{a, b, c, +, *, (,)\}, P_3, E)$ ku P_3:

$$\begin{aligned} E &\rightarrow T \mid E + T \\ T &\rightarrow F \mid T * F \\ F &\rightarrow (E) \mid a \mid b \mid c \end{aligned}$$

G_3 është e qartë. Gjithashtu $L(G_3) = L(G_2)$.

Pema e derivimit të $a + b * c$ në G_3



Shembull : (Chomsky, 1964) Një gjuhë pa kontekst e paqartë është

$$L = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } (i = j \text{ or } j = k)\}.$$

§2 Lema kërcyese

Lema kërcyese per gjuhët pa kontekst përbën një kusht të nevojshëmc që një gjuhë të jetë pa kontekst.

2.1 **Teorem (Lema kërcyese për gjuhët pa kontekst ose lema-*uvwxy*)**: Për çdo gjuhë pa kontekst $L \subseteq T^*$ egziston një numër $n \in \mathbb{N}$, i tillë që për të gjithë stringjet $z \in L$ me $|z| \geq n$ atëhere do të egzistojë një dekompozim $z = uvwxy$ me veçritë si më poshtë:

(i) $vx = \varepsilon$,

(ii) $|vwx| \leq n$,

(iii) për të gjithë $i \in \mathbb{N}$ kemi: $uv^iwx^iy \in L$.

Mund të heqim ose përsërisim nënstringjet v dhe x një numër të pakufizuar herësh, dhe stringjet e rinj të përftuar të jenë pjesë e gjuhës pa kontekst L.

Përcaktime për pemën e fundme t:

- Faktor degëzimi i t = numri më i madh i bijve që ka një nyje në t.
- Shteg me gjatësi m në t është një sekuencë nyjesh nga rrënja te gjethja e t me m nyje. Rast i veçantë $m = 0$.

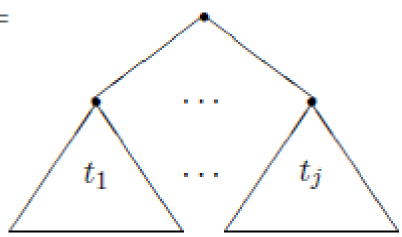
2.2 **Lemë** : Le të jetë t një pemë e fundme me faktor degëzimi $\leq k$, ku çdo shteg ka gjatësi $\leq m$. Atëhere numri i gjetheve në t është $\leq k^m$.

Vërtetim : Induksion mbi $m \in \mathbb{N}$:

$m = 0$: t përbëhet vetëm nga $k^0 = 1$ nyje.

$m \to m+1$: t ka j nënpemë $t_1, \dots, t_j$ me $j \leq k$, kë shtigjet kanë gjatësi $\leq m$:

$t =$



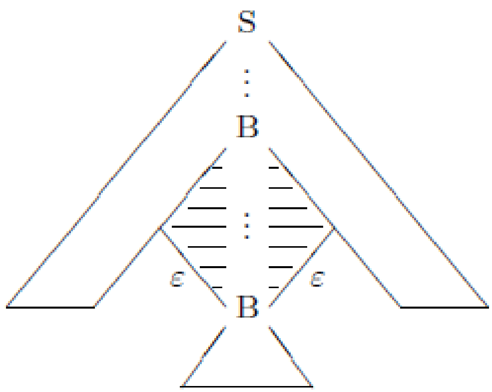
Në bazë të supozimit, numri i gjetheve në secilën prej nënpemëve $t_1, \dots, t_j$ is $\leq k^m$. Pra arrijmë në përfundimin që për t:

$$\text{numri gjetheve} \leq j \cdot k^m \leq k \cdot k^m = k^{m+1}.$$

Vërtetim i lemës kërcyese: Le të jetë $G = (N, T, P, S)$ një gramatikë pa kontekst me $L(G) = L$. Let:

- $k =$ gjatësia më e madhe në rregullat e P, të paktën 2
- $m = |N|$,
- $n = k^{m+1}$.

Marrim $z \in L$ me $|z| \geq n$. Atëhere egziston një pemë derivimi t nga S te z në G pa nënpemë që i përket derivimit $B \vdash_G {}^* B$:

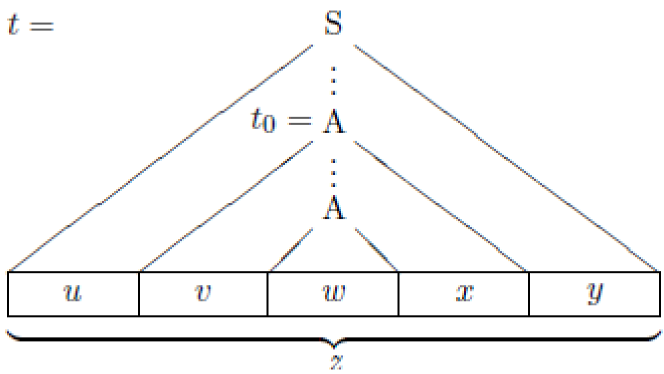


Çdo pjesë e tillë mund të hiwqet nga t pa ndryshuar stringun e derivuar z.

Për k dhe $|z|$ më sipër, vëmë re që t ka faktor degëzimi $\leq k$ dhe $\geq k^{m+1}$ gjethe. Prandaj, sipas lemës më sipër, egziston një shteg me gjatësi $\geq m + 1$ në t. Pra kemi $\geq m + 1$ nyje të brendshme, pra të paktën një simbol jofundor do të përsëritet.

Do të quajmë pemë përsëritje në t, një nënpemë të t, ku etiketa e rrënjës përsëritet në ndpnjë nyje. zgjedhim pemën përsëritëse minimale t_0 në t, pemë që nuk përmban nënpemë përsëritëse. Në t_0 çdo shteg ka gjatësi $\leq m + 1$.

Shënojmë me A rrënjën e t_0. Atëhere t ka strukturën e mëposhtme:



Nga kjo strukturë, marrim dekompozimin $z = uvwxy$ me

$$S \vdash^*_G uAy \vdash^*_G uvAxy \vdash^*_G uvwxy \qquad (*)$$

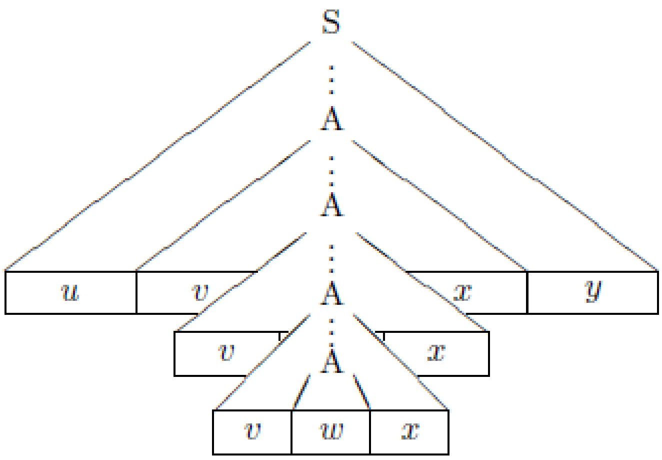
Vëmë re se ky dekompozim i z kënaq kushtet e lemës kërcyese:

(i) Nga zgjedhja e t kemi $vx = \varepsilon$.

(ii) Nga zgjedhja e t_0 dhe kemi $|vwx| \leq k^{m+1} = n$.

(iii) Nga $(*)$ rrjedh që për të gjithë $i \in$ N kemi: $uv^iwx^iy \in L(G)$.

Pema e derivimit nga S te uv^iwx^iy në G për $i = 3$ jepet më poshtë:



Lema kërcyese mund të përdoret për të vërtetuar që një gjuhë e dhënë nuk është pa kontekst.

Shembull: Gjuha $L = \{a^nb^nc^n \mid n \in \text{N}\}$ nuk është pa kontekst. E vërtetojmë duke përdorur vërtetimin nga e kundërta.

Hipoteza: L është pa kontekst. Atëhere sipas lemës kërcyese egziston $n \in$ N.

Marrim $z = a^nb^nc^n$. Mqs $|z| \geq n$, mund ta dekompozojmë z në $z = uvwxy$ me $vx = \varepsilon$ dhe $|vwx| \leq n$, e tillë që për çdo $i \in$ N kemi: $uv^iwx^iy \in L$. Mqs $|vwx| \leq n$, nuk mund të kemi edhe a edhe c në nënstringun vwx. Pra, duke kaluar te stringu i ri uv^iwx^iy, të shumtën dy nga karakteret a, b, c do të preken njëkohësisht. Kjo sjell që nënstringjet e përftuar nuk janë në L. *Kundërshtim.*

Lema kërcyese na tregon që gramatikat pa kontekst nuk mjaftojnë të japin një përshkrim të plotë të gjuhëve të avancuaara të programimit si JAVA.

Shembull : Gjuha e programimit Java nuk është pa kontekst. E vërtetojmë duke përdorur vërtetimin nga e kundërta.

Hipoteza: Supozojmë që Java është pa kontekst. Atëhere egziston kufiri $n \in \mathbb{N}$.

Vini re një klasë Java korrekte:

$$
\begin{array}{l}
\texttt{class}\ C\ \{ \\
\quad \texttt{int}\ X\underbrace{1\ldots1}_{n}; \\
\quad \texttt{void}\ m()\ \{ \\
\qquad X\underbrace{1\ldots1}_{n} = X\underbrace{1\ldots1}_{n} \\
\quad \} \\
\}
\end{array}
$$

Në çdo dekompozim *uvwxy* të programit, pjesa *vwx* prek të shumtën dy nga tre paraqitjet e variablit $X\underbrace{1\ldots1}_{n};$, pasi $|vwx| \leq n$. Pra stringjet e reja të përftuara uv^iwx^iy, do jenë të formës

$$
\begin{array}{l}
\texttt{class}\ C\ \{ \\
\quad \texttt{int}\ X\underbrace{1\ldots1}_{k}; \\
\quad \texttt{void}\ m()\ \{ \\
\qquad X\underbrace{1\ldots1}_{l} = X\underbrace{1\ldots1}_{m} \\
\quad \} \\
\}
\end{array}
$$

ku k, l, m nuk janë të treja të barabarta. Keto stringje nuk janë programe korrekte Java pasi nuk plotësojnë kushtin:

"Çdo variabël duhet të deklarohet para se të përdoret." (∗∗)

Ose $X\underbrace{1\ldots1}_{m}$ ose $X\underbrace{1\ldots1}_{l}$ nuk janë të deklaruar.