Kimenettel rendelkező automaták

Kiegészítő anyag a Nyelvek és automaták jegyzethez

Friedl Katalin
BME SZIT
friedl@cs.bme.hu

2019. december 11.

A különböző automatatípusok (véges automata, veremautomata, Turing-gép) alap modellje nyelvek elfogadására való, az a kérdés, hogy a bemeneten végzett számítás után elfogadó vagy nem elfogadó állapotban áll meg az automata (vagy, hogy Turing-gép esetén megáll-e egyáltalán).

Itt most olyan változatokkal ismerkedünk meg, amelyek egy bit információn kívül (elfogad/nem fogad el) több információt adnak. Ezt általában úgy lehet elképzelni, hogy a szokásos bemeneten (bemeneti szalagon) túl egy kimeneti szalaggal is ellátjuk az automatát, amire a számítás során lépésenként kiír valamit. A számítás végén ennek a kimeneti szalagnak a tartalma a számítás eredménye. A pontos szabályok, megengedett lépések az egyes modellekben különbözők.

1. Véges automaták

A véges automatákhoz két természetes módon is rendelhetünk kimenetet.

1.1. Moore-automata

A Moore-automata esetében egy *determinisztikus, teljes* véges automatát egészítünk ki azzal, hogy minden állapotához tartozzon egy kimenet.

1. Definíció. *A* Moore-automata $M = (Q, \Sigma, \Delta, \delta, \mu, q_0)$ *ahol:*

- Q *egy véges, nem üres halmaz. Ez az automata* állapotainak *halmaza.*
- Σ *egy véges, nem üres halmaz. Ez az automata* bemeneti ábécé*je.*
- Δ *egy véges, nem üres halmaz. Ez az automata* kimeneti ábécé*je.*
- $\delta : Q \times \Sigma \to Q$, *az automata* állapotátmeneti függvénye.
- $\mu : Q \to \Delta$ *a kimeneti függvény.*

- $q_0 \in Q$ *a* kezdő állapot.

A Moore-automata *működése* egy adott $w \in \Sigma^*$ szón a következőképpen írható le. Az automatát a q_0 állapotból indítjuk. Ha w az n hosszú $a_1 a_2 \cdots a_n$ karaktersorozat ($a_i \in \Sigma$), akkor először az $r_1 = \delta(q_0, a_1)$ állapotba lép és kiírja a $\mu(r_1)$ karaktert. A második lépésben az $r_2 = \delta(r_1, a_2)$ állapotba kerül, és kiírja a $\mu(r_2)$ karaktert, és így tovább, amíg az a_n karaktert is feldolgozza és kiírja az utolsó r_n állapotnak megfelelő $\mu(r_n)$ karaktert. Ekkor a számítás véget ér. A $w = a_1 a_2 \cdots a_n$ bemeneten a teljes kimenet a $\mu(r_1)\mu(r_2)\cdots\mu(r_n) \in \Delta^*$ szó.

Ha a kimenetre nem figyelünk, azaz eltekintünk Δ-tól és a μ-től, akkor ez egy determinisztikus, teljes véges automata, aminél nincs megadva az elfogadó állapotok halmaza.

Az M automata alapján a μ függvény kiterjeszthető Σ^* elemeire is,

2. Definíció. *A $\mu : Q \to \Delta$ kimeneti függvény $\overline{\mu} : \Sigma^* \to \Delta^*$ kiterjesztése a következő*

$$\overline{\mu}(w) = \begin{cases} \varepsilon & \textit{ha } w = \varepsilon \\ \mu(r_1)\mu(r_2)\cdots\mu(r_n) & \textit{ha } w \neq \varepsilon \textit{ és } q_0, r_1, \cdots, r_n \textit{ az állapotok} \\ & \textit{sorozata } w \textit{ bemenetnél} \end{cases}$$

1. Megjegyzés. *Gyakran a kiterjesztést is csak μ-vel jelölik.*

1. Tény. *Vegyük észre, hogy $\overline{\mu} : \Sigma^n \to \Delta^n$ teljesül minden $n \geq 0$ egész számra.*

1. Következmény. *Van olyan $f : \Sigma^* \to \Delta^*$ függvény, amihez nincs $M = (Q, \Sigma, \Delta, \delta, \mu, q_0)$ Moore-automata, hogy $f = \overline{\mu}$.*

Bizonyítás: Ha f nem őrzi meg a szavak hosszát, akkor biztos nincs megfelelő Moore-automata. □

1. Példa. *Ha egy tetszőleges teljes DVA-nál $\Delta = Q$ és $\mu(q) = q$, akkor minden w szóra a $\overline{\mu}(w)$ kimenet a számítás során érintett állapotok sorozata lesz (a kezdő q_0 kivételével).*

2. Példa. *Egy tetszőleges M teljes DVA-ra legyen $\mu(q) = \mathtt{1}$, ha q elfogadó és $\mathtt{0}$ különben, $\Delta = \{\mathtt{0}, \mathtt{1}\}$. Ekkor $\overline{\mu}(w)$ utolsó karaktere megmutatja, hogy w-t elfogadja-e M, sőt a teljes kimenetből azt is látjuk, hogy w mely kezdőszeleteit fogadja el M.*

1. Feladat. *Adjunk meg egy olyan Moore-automatát, amelyiknél $\Sigma = \Delta = \{\mathtt{0}, \mathtt{1}\}$ és a kimeneti $\overline{\mu}(a_1 a_2 \ldots a_n) = b_1 b_2 \ldots b_n$ függvényre $n \geq 1$ esetén $b_i = 1 - a_i$ teljesül ($1 \leq i \leq n$)!*

Megoldás: Legyen $Q = \{A, B\}$ és A a kezdőállapot. Az átmeneti függvény

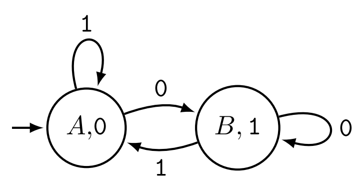
$$\begin{aligned} \delta(A, \mathtt{0}) &= B \\ \delta(A, \mathtt{1}) &= A \\ \delta(B, \mathtt{0}) &= B \\ \delta(B, \mathtt{1}) &= A, \end{aligned}$$

a kimeneti függvény pedig

$$\mu(A) = \mathtt{0}$$
$$\mu(B) = \mathtt{1}.$$

Ez az automata a `0` hatására mindig B-be lép és akkor egy `1`-t ír ki, míg egy `1`-re A-ba lép és ezért egy `0`-t ír ki. □

Most is lehet az automatáknál már megszokott irányított gráfos ábrázolást használni (de most nem lesz szükség az elfogadó állapotokat jelző dupla körre). Ekkor a kimeneti karaktert az állapotba írjuk bele. Az előző automata ebben a formában:



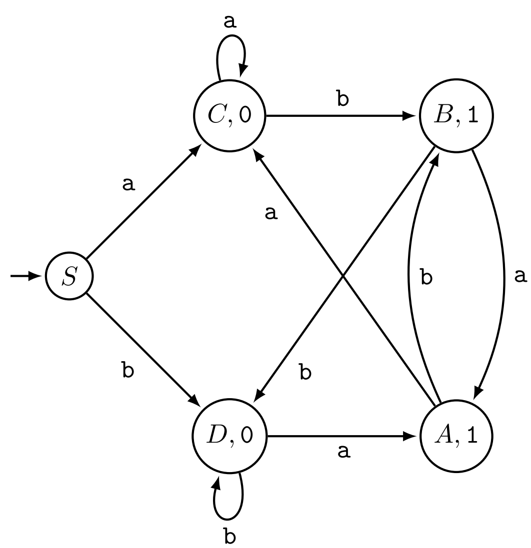
Sok esetben a megvalósítani kívánt f függvény nincs előírva minden lehetséges bemeneten, csak egy L nyelven. Ilyenkor a feladat olyan Moore-automatát készíteni, ami ezeknek az előírásoknak eleget tesz, de az L-be nem tartozó szavakon tetszőlegesen viselkedhet.

Érdekes kérdés, hogy mely f függvényekhez van olyan Moore-automata, hogy $f = \overline{\mu}$. Azt már tudjuk, hogy f-nek hossztartónak kell lenni, de ez nem elég. (Miért nem?)

Más szempontból szokás azt vizsgálni, hogy egy adott Moore-automatának mi az értékkészlete, azaz mi lesz a $\overline{\mu}(\Sigma^*) = \{f(w) : w \in \Sigma^*\} \subseteq \Delta^*$ nyelv. Vagy még általánosabban: mi lesz egy adott L nyelv $\overline{\mu}(L) = \{f(w) : w \in L\} \subseteq \Delta^*$ képe.

2. Feladat. *(a) Adjunk olyan Moore-automatát, amire* $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, $\Delta = \{\mathtt{0}, \mathtt{1}\}$ *és a számítás során mindig ott ír ki* `1`*-t, amikor a bemeneten* `a` *után* `b`*, vagy* `b` *után* `a` *következik! Egy hosszú bemenet esetén legyen* $\mu(\mathtt{a}) = \mu(\mathtt{b}) = \mathtt{0}$.
(b) Mi lesz a $\overline{\mu}(\Sigma^*)$ *nyelv?*
(c) Mi lesz a $\overline{\mu}(\mathtt{a}^*\mathtt{b}^*)$ *nyelv?*
(d) Mi lesz a $\overline{\mu}(\{\mathtt{a}^n\mathtt{b}^n : n \geq 1\})$ *nyelv?*

Megoldás: Egy lehetséges megoldás, hogy lesz egy-egy állapot az `a`-k utáni első `b`, és a `b`-k utáni első `a` betű feldolgozására (A, B)– ilyenkor `1` a kimenet, továbbá egy-egy `0`-t kiíró állapot a többi `a` és többi `b` kezelésére (C, D).



(b) A kimenet, ha nem az üres szó, akkor biztos, hogy 0-val kezdődik, de könnyű látni, hogy utána tetszőleges szó állhat, azaz $\overline{\mu}(\Sigma^*) = \{\varepsilon\} \cup 0\Sigma^*$.

(c) Az üres szó kivételével ezekben a szavakban legfeljebb egy karakterváltás van és 0-val kezdődnek, azaz a nyelv: $\{\varepsilon\} \cup 00^*10^*$.

(d) Itt pontosan egy karakterváltás lesz, az is középen, $\{0^n10^{n-1} : n \geq 1\}$.
□

1.2. Mealy-automata

A Mealy-automata esetében a kimeneti karakter nem az állapothoz, hanem az átmenethez tartozik.

3. Definíció. *A* Mealy-automata $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ *ahol:*

- Q *egy véges, nem üres halmaz. Ez az automata* állapotainak *halmaza.*
- Σ *egy véges, nem üres halmaz. Ez az automata* bemeneti ábécé*je.*
- Δ *egy véges, nem üres halmaz. Ez az automata* kimeneti ábécé*je.*
- $\delta : Q \times \Sigma \to Q$*, az automata* állapotátmeneti függvénye.
- $\lambda : Q \times \Sigma \to \Delta$ *a kimeneti függvény.*
- $q_0 \in Q$ *a* kezdő állapot.

A Mealy-automata *működése* egy adott $w \in \Sigma^*$ szón a következőképpen írható le. Az automatát a q_0 állapotból indítjuk. Ha w az n hosszú $a_1a_2 \cdots a_n$ karaktersorozat ($a_i \in \Sigma$), akkor először az $r_1 = \delta(q_0, a_1)$ állapotba lép és kiírja a $\lambda(q_0, a_1)$ karaktert. A második lépésben az $r_2 = \delta(r_1, a_2)$ állapotba kerül, és kiírja a $\lambda(r_1, a_2)$ karaktert, és így tovább, amíg az a_n karaktert is feldolgozza és kiírja az utolsó, $\delta(r_{n-1}, a_n)$ lépésnek megfelelő $\lambda(q_{n-1}, a_n)$ karaktert. Ekkor a számítás véget ér. A $w = a_1 a_2 \cdots a_n$ bemeneten a teljes kimenet a $\lambda(q_0, a_1)\lambda(r_1, a_2) \cdots \lambda(r_{n-1}, a_n) \in \Delta^*$ szó.

A kimeneti függvény most is kiterjeszthető Σ^* elemeire,

4. Definíció. *A $\lambda : Q \times \Sigma \to \Delta$ kimeneti függvény $\overline{\lambda} : \Sigma^* \to \Delta^*$ kiterjesztése a következő*

$$\overline{\lambda}(w) = \begin{cases} \varepsilon & \text{ha } w = \varepsilon \\ \overline{\lambda}(w')\lambda(r, a) & \text{ha } w = w'a,\ a \in \Sigma \text{ és } M \text{ az } r \text{ állapotban} \\ & \text{van } w' \text{ után} \end{cases}$$

2. Megjegyzés. *Gyakran a kiterjesztést is csak λ-val jelölik.*

2. Tény. *Vegyük észre, hogy $\overline{\lambda} : \Sigma^n \to \Delta^n$ teljesül minden $n \geq 0$ egész számra.*

3. Feladat. *Az 1. feladatra most készítsünk egy Mealy-automatát.*

Megoldás: A véges automata legyen ugyanaz, de az állapotokhoz tartozó μ-értékek helyett legyen most

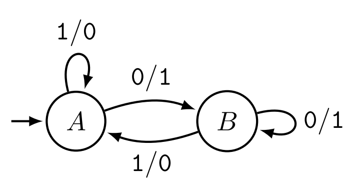
$$\begin{aligned} \lambda(A, \mathtt{0}) &= \mathtt{1} \\ \lambda(A, \mathtt{1}) &= \mathtt{0} \\ \lambda(B, \mathtt{0}) &= \mathtt{1} \\ \lambda(B, \mathtt{1}) &= \mathtt{0}, \end{aligned}$$

így minden olvasott karakterre, állapottól függetlenül, a karakter ellentétét írja ki.

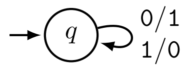
Mivel most a kimeneti függvény értelmezési tartománya megegyezik az átmeneti függvény értelmezési tartományával, szokás a két függvényt egyetlen táblázatban megadni,

	0	1
A	B,1	A,0
B	B,1	A,0

és természetesen a rajzos változat is használható, de most egy átmenetet jelképező nyíl címkéje két részből áll, a formája: olvasott/kiírt karakter.



Ugyanerre a feladatra egy másik megoldás:



□

Az, hogy erre a feladatra Moore- és Mealy-automatát is tudtunk adni, nem véletlen. A következőkben megmutatjuk, hogy a két típus ekvivalens.

1. Tétel. *Minden $M_o = (Q_o, \Sigma, \Delta, \delta_o, \mu, q_o)$ Moore-automatához van olyan $M_e = (Q_e, \Sigma, \Delta, \delta_e, \lambda, q_e)$ Mealy-automata, hogy $\overline{\mu} = \overline{\lambda}$.*

Bizonyítás: A definíciók szerint $\overline{\mu}(\varepsilon) = \overline{\lambda}(\varepsilon) = \varepsilon$, tehát csak a nem üres szavakkal kell foglalkoznunk. A bizonyítás ötlete az, hogy az állapothoz tartozó kimenetet az állapotba beérkező átmenetekre „toljuk át”.

Legyen $Q_e = Q_o$, $\delta_e = \delta_o$, $q_o = q_e$, tehát az alapot adó két véges automata megegyezik. Az új kimeneti függvény minden q állapotra és $a \in \Sigma$ karakterre legyen $\lambda(q, a) = \mu(\delta_o(q, a))$.

A konstrukció helyességének belátásához tekintsünk egy tetszőleges $w = a_1 a_2 \cdots a_n \in \Sigma^*$ szót, és legyen ezen a bemeneten az automata állapotsorozata $q_o, r_1, \ldots, r_n$. Ekkor a Moore-automata kimenete a definíció szerint $\overline{\mu}(w) = \mu(r_1)\mu(r_2)\cdots\mu(r_n)$. A konstruált Mealy-automata kimenete pedig

$$\begin{aligned}\overline{\lambda}(w) &= \lambda(q_0, a_1)\lambda(r_1, a_2)\cdots\lambda(r_{n-1}, a_n)\\ &= \mu(\delta_o(q_0, a_1))\mu(\delta_o(r_1, a_2))\cdots\mu(\delta_o(r_{n-1}, a_n))\\ &= \mu(r_1)\mu(r_2)\cdots\mu(r_n) = \overline{\mu}(w).\end{aligned}$$

□

Az előző konstrukció, bár egy kicsit bonyolultabban, de megfordítható.

2. Tétel. *Minden $M_e = (Q_e, \Sigma, \Delta, \delta_e, \lambda, q_e)$ Mealy-automatához van olyan $M_o = (Q_o, \Sigma, \Delta, \delta_o, \mu, q_o)$ Moore-automata, hogy $\overline{\lambda} = \overline{\mu}$.*

Bizonyítás: Most az átmenetekről kell az állapotokba tolni a kimenetet, aminél gondot jelent, hogy egy állapotba vivő átmenetekhez többféle kimenet is tartozhat. Ezért az állapotokat megsokszorozzuk, minden M_e-beli q állapotból $|\Delta|$ példányt csinálunk, minden lehetséges kimenethez egyet. Tehát legyen $Q_o = Q_e \times \Delta$. A q_e kezdőállapot egy tetszőlegesen választott példánya (q_e, b) lesz a kezdő állapot ($b \in \Delta$).

Egy $(q, c) \in Q_e \times \Delta$ állapothoz tartozó kimenet legyen $\mu(q, c) = c$.

A Moore-automata átmeneti függvénye $\delta_o((q, c), a) = (\delta_e(q, a), \lambda(q, a))$.

Annak megmutatásához, hogy ez így jó, tekintsünk megint egy tetszőleges $w = a_1 a_2 \cdots a_n \in \Sigma^*$ szót, és legyen ezen a bemeneten a Mealy-automata állapotsorozata $q_e, r_1, \ldots, r_n$. Ekkor $\overline{\lambda}(w) = \lambda(q_e, a_1)\lambda(r_1, a_2)\cdots\lambda(r_{n-1}, a_n)$. A Moore-automata a $q_o = (q_e, b)$ kezdőállapotból először a

$$\delta_o((q_e, b), a_1) = (\delta_e(q_e, a_1), \lambda(q_e, a_1)) = (r_1, \lambda(q_e, a_1))$$

állapotba lép és itt az állapot második részét, tehát a $\lambda(q_e, a_1)$ karaktert írja ki. A következő lépése a

$$\delta_o((r_1, \lambda(q_e, a_1)), a_2) = (\delta_e(r_1, a_2), \lambda(r_1, a_2)) = (r_2, \lambda(r_1, a_2))$$

és ekkor a $\lambda(r_1, a_2)$ karaktert írja ki, stb. Látható, hogy a kimenet $\overline{\mu}(w) = \overline{\lambda}(w)$ teljesülni fog.

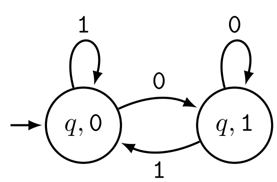
Az üres szóra pedig a kimeneti függvények definíciója szerint $\overline{\mu}(\varepsilon) = \overline{\lambda}(\varepsilon) = \varepsilon$. □

4. Feladat. *Készítsük el az 1. feladatban megadott Moore-automatából a megfelelő Mealy-automatát!*

Megoldás: A konstrukció szerint az automata ugyanaz lesz, csak a kimenetek kerülnek rá az átmenetekre. Ha ezt megvalósítjuk, épp a 3. feladatra elsőként kapott Mealy-automatát kapjuk. □

5. Feladat. *Készítsük el a 3. feladat második megoldásában megadott 1 állapotú Mealy-automatából a megfelelő Moore-automatát!*

Megoldás: A konstrukció szerint az új automatának $|Q| \cdot |\Delta| = 2$ állapota lesz, egy a `0`, egy az `1` kimenethez.



□

2. Következmény. *Egy L nyelvhez akkor és csak akkor van olyan Moore-automata, melyre $\overline{\mu}(\Sigma^*) = L$, ha van Mealy-automata, melyre $\overline{\lambda}(\Sigma^*) = L$.*

3. Következmény. *Adott L_1 és L_2 nyelvekhez akkor és csak akkor van olyan Moore-automata, melyre $\overline{\mu}(L_1) = L_2$, ha van Mealy-automata, melyre $\overline{\lambda}(L_1) = L_2$.*

3. Tétel. *Ha az $L \subseteq \Delta^*$ nyelv előáll, mint egy Moore- vagy Mealy-automata értékkészlete, akkor L egy reguláris nyelv.*

Bizonyítás: Legyen $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ egy Mealy-automata, melyre $\lambda(\Sigma^*) = L$. Cseréljük le a δ átmeneti függvényt λ-ra és tekintsük az $M' = (Q, \Delta, \lambda, q_0, Q)$ véges automatát.

Vegyük észre, hogy M' elfogadja az üres szót, ami azért helyes, mert a Mealy-automata definíciója szerint $\varepsilon = \lambda(\varepsilon) \in L$.

Ha M egy $w \in \Sigma^*$ szón a $q_0, r_1, r_2, \ldots, r_n$ állapotokon ment végig, miközben a $w' \in \Delta^*$ szót írta ki, akkor az új M' automata a w' szón ugyanezeken az állapotokon megy végig, és mivel M'-ben minden állapot elfogadó, el is fogadja a w' szót. Igazából M' pontosan azokat a szavakat fogadja el, amelyeken a számítása nem akad el, ezek viszont épp azok, amik előállnak az M automata kimeneteként. $\square$

2. Fordító automaták

A Moore- és Mealy-automaták, mint láttuk, csak olyan $f : \Sigma^* \to \Delta^*$ függvényeket képesek előállítani, amelyek n hosszú szavakat n hosszú szavakba képeznek. Ha azt akarjuk, hogy legyen lehetőség a hossz megváltoztatására is, akkor meg kell engedni, hogy egy olvasott karakterhez ne csak egy kimeneti karakter tartozhasson.

Azt is célszerű megengedni, hogy egy szónak több fordítása legyen, ahogy ez a természetes nyelveknél is van. Ehhez már nemdeterminisztikus automatákra lesz szükség.

A *fordítás* általános értelemben nem egy függvény, hanem egy reláció, a $\Sigma^* \times \Delta^*$ egy F részhalmaza. Egy $x \in \Sigma^*$ szóhoz akár több $y \in \Delta^*$ is tartozhat, amire $(x, y) \in F$. Tekinthetjük úgy, hogy az F reláció egy szótár, ami megadja az x szó lehetséges fordításait.

Ez megfelel annak, hogy egy nemdetermnisztikus számításnál több számítási ág is lehet, amik akár különböző kimenetet is produkálhatnak.

5. Definíció. *Legyen Σ és Δ két ábécé. Általános értelemben az F* fordítás *egy $F \subseteq \Sigma^* \times \Delta^*$ reláció.*

Egy $x \in \Sigma^*$ szó *fordításai* $F(x) = \{y \in \Delta^* : (x, y) \in F\}$ azok az $y \in \Delta^*$ szavak, melyekre $(x, y) \in F$. Az is előfordulhat, hogy egy x szónak nincs fordítása, $F(x) = \emptyset$.

Egy $L \subseteq \Sigma^*$ nyelv *fordítása* az $F(L) = \{y \in \Delta^* : \exists x \in L, (x, y) \in F\} \subseteq \Delta^*$ nyelv.

2.1. Véges fordító

Egy M véges fordító hasonló a Mealy-automatához, de fontosak a különbségek:

- a véges automata nem kell, hogy teljes legyen,
- a véges automata lehet nemdeterminisztikus, tartalmazhat ε-mozgásokat,
- egy lépésben tetszőleges (akár 0 darab) karaktert ír ki.

Mivel egy lépésben a kimenet az átmenettől függ, ezt most beépítjük az átmeneti függvénybe. (Ezt a Mealy-automatánál is megtehettük volna, de ott egyszerűbb volt külön kezelni.)

6. Definíció. *Az M* véges fordító *egy nemdeterminisztikus véges automata, melynek átmeneti függvénye $\delta(q,a) \subseteq Q \times \Delta^*$, ahol $q \in Q$ egy állapot, $a \in \Sigma \cup \{\varepsilon\}$ a bemenetről olvasott karakter. Ha $(r,\alpha) \in \delta(q,a)$, akkor egy lehetséges lépése az automatának az, hogy r állapotba kerül és a kimenetre az $\alpha \in \Delta^*$ karaktersorozatot írja ki.*

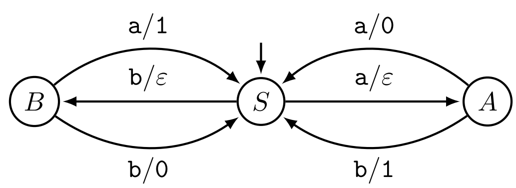
Egy x bemeneten egy *számítás kimenetét* a lépésenkénti kimenetek összefűzése adja, feltéve, hogy az automata nem akad el a számítás közben. Amennyiben x feldolgozása közben a számítás elakad, akkor a fordítás nincs értelmezve.

Az M-hez tartozó fordítást $F_M \subseteq \Sigma^* \times \Delta^*$ jelöli.

Egy *x szó fordítása* a lehetséges kimenetek az x-en létrejövő kimenetek $F_M(x)$ halmaza. $F_M(\varepsilon) = \{y : (x,y) \in F\} \cup \{\varepsilon\}$.

3. Tény. *Egy M véges fordítónál egy x szó $F_M(x)$ fordításai között lehet hosszabb és lehet rövidebb is mint x. Például, ha minden $(r,\alpha) \in \delta(q,a)$ átmenetben $|\alpha| > 1$,akkor minden nem üres szó hossza nőni fog a fordításkor. Másrészt mivel $\alpha = \varepsilon$ is lehetséges, ezért a fordításkor rövidülhetnek is a szavak.*

6. Feladat. *Tekintsük az alábbi véges fordítót. Az egyes átmenetek címkéje, a Mealy-automatához hasonlóan, a bemenet/kimenet párt adja meg.*



(a) Mi lesz egy $w_1 w_2 \cdots w_n$ szó $w_i \in \{\texttt{a},\texttt{b}\}$ fordítása?
(b) Mi $F_M(L)$, ha $L = \{\texttt{a},\texttt{b}\}^*$?
(c) Mi $F_M(L)$, ha $L = \{\texttt{a}^n\texttt{b}^k : n,k \geq 0\}$?
(d) Mi $F_M(L)$, ha $L = \{\texttt{a}^n\texttt{b}^n : n \geq 0\}$?

Megoldás: (a) Ez a fordító úgy viselkedik, hogy minden második karakternél ír ki egy karaktert, ami `0`, ha $w_{2i-1} = w_{2i}$, és `1`, ha $w_{2i-1} \neq w_{2i}$. A kimenet hossza $\lfloor n/2 \rfloor$.

(b) Tetszőleges $x_1 x_2 \cdots x_k$ 0-1 sorozathoz vegyük azt a szót, melyben $a_{2i-1} = \texttt{a}$ és a_{2i} `a` vagy `b`, attól függően, hogy x_i nulla vagy egy ($1 \leq i \leq k$). Ez a pár a fordításnak egy eleme, ezért $F_M(L) = \{\texttt{0},\texttt{1}\}^*$

(c) Ezekben a szavakban legfeljebb egy olyan pár van, ahol a két betű nem egyezik meg, ezért $F_M(L)$ azokból a $\{\texttt{0},\texttt{1}\}^*$-beli szavakból áll, melyekben legfeljebb egy darab `1` van. (Miért kaphatunk meg minden ilyen szót?)

(d) Ha n páros, akkor a fordítás eredménye $\texttt{0}^n$. Páratlan n esetén kapunk egy egyest, előtte és utána ugyanannyi ($(n-1)/2$ darab) nullával, a nyelv tehát $F_M(L) = \{\texttt{0}^k : k \text{ páros }\} \cup \{\texttt{0}^k\texttt{1}\texttt{0}^k : k \geq 0\}$. □

4. Tény. *Minden véges fordító átalakítható olyanná, amelyik egy lépésben vagy csak ír, vagy csak olvas. Ehhez annyit kell tenni. hogy minden* $(r, \alpha) \in \delta(q, a)$ *átmenetet átmenetek egy sorozatával oldunk meg: először csak olvasunk, utána minden lépésben egy-egy karaktert írunk ki. Azaz ha* $\alpha = \mathtt{b}_q\mathtt{b}_2 \cdots \mathtt{b}_k$*, akkor felveszünk* k *új állapotot, legyenek ezek* $t_1, t_2, \ldots, t_k$*, és az alábbi átmeneteket:* $(t_1, \varepsilon) \in \delta(q, a)$, $(t_2, \mathtt{b}_1) = \delta(t_1, \varepsilon)$, $(t_3, \mathtt{b}_2) = \delta(t_2, \varepsilon)$, ..., $(t_k, \mathtt{b}_{k-1}) = \delta(t_{k-1}, \varepsilon)$, $(r, \mathtt{b}_k) = \delta(t_k, \varepsilon)$.

Ahhoz, hogy a fordítás ne változzon, azt is fel kell tenni, hogy az újonnan bevezetett állapotokban nem állhat meg a számítás. Ez arra hasonlít, mintha az eredeti állapotok elfogadó állapotok lennének, az újak nem, és csak azt tekintjük érvényes fordításnak, ami elfogadó állapotban ér véget.

4. Tétel. *Legyen* $L \subseteq \Sigma^*$ *egy reguláris nyelv és* M_f *egy véges fordító. Ekkor az* $F_{M_f}(L) \subseteq \Delta^*$ *nyelv is reguláris.*

Bizonyítás vázlat: Mivel L reguláris, van hozzá determinisztikus teljes véges automata, legyen $M = (Q, \Sigma, \delta, q_0, F)$, melyre $L(M) = L$.

Legyen $M_f = (R, \Sigma, \Delta, \delta_f, r_f)$ a véges fordító, amelyről az egyszerűség kedvéért feltehetjük, hogy az előző megjegyzés értelmében minden lépésben vagy csak ír vagy csak olvas.

Ezekből kell egy M' véges automatát készíteni az $f_{M_f}(L)$ nyelvhez.

Az ötlet, hogy a fordító olvasó lépéseit az M lépéseivel hangoljuk össze, a tényleges lépések a fordító kiíró lépéseiből származnak.

Ehhez legyen M' állapotainak halmaza $Q \times R$, a kezdőállapot (q_0, r_f). Az M' átmeneti függvényét jelölje δ'. Ezt így adhatjuk meg: Egy $(q, r) \in Q \times R$ állapotból, ha

M_f **olvasna:** $\delta(q, a) = q'$ és $(r', \varepsilon) \in \delta_f(r, a)$, akkor legyen az új állapot (q', r')

M_f **írna:** $(r', b) \in \delta_f(r, \varepsilon)$, akkor legyen az új állapot (q, r').

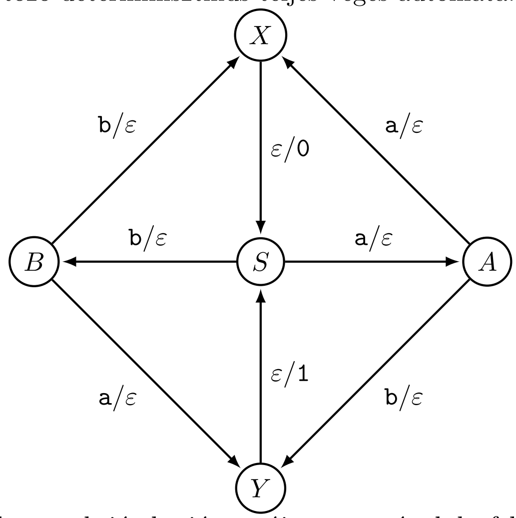
Ha elfogadó állapot lesz minden olyan (q, r) pár, ahol $q \in F$ és az r az eredeti fordítónak is állapota, nem a fordítónak a korábban vázolt átalakításakor az olvasás–írás szétválasztásakor hoztuk létre, akkor az M' valóban a kívánt nyelvet fogadja el, azaz azokat a szavakat, amik előállnak fordításként.

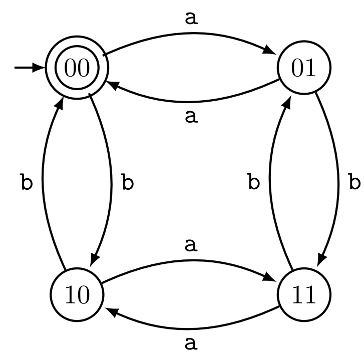
□

3. Megjegyzés. *A konstrukció egy nemdeterminisztikus,* ε*-átmenetes véges automatát ad. Ebből azután a szokott módon kiküszöbölhetjük az* ε*-mozgásokat, és determinizálhatjuk is az automatát.*

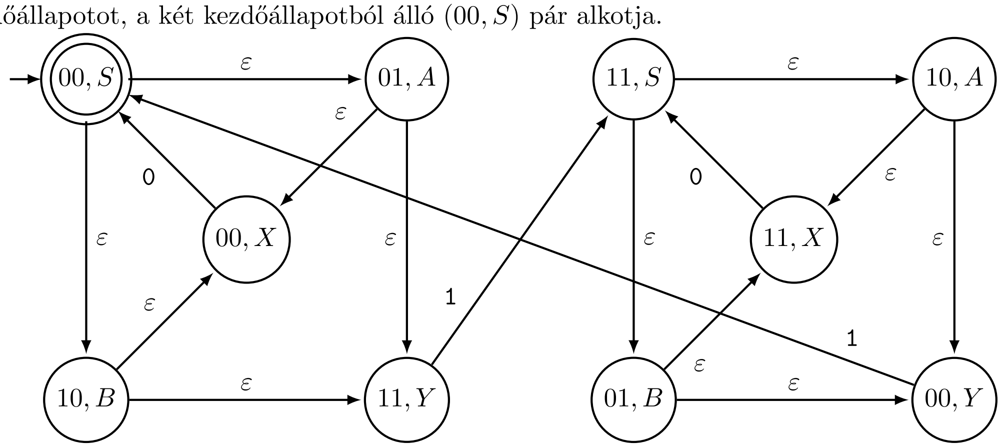
7. Feladat. *Álljon az* $L \subset \{\mathtt{a}, \mathtt{b}\}^*$ *azokból a szavakból, melyekben az* `a` *és a* `b` *betűk száma is páros. A fordító pedig legyen az, ami a 6. feladatban szerepelt. A bizonyítás alapján konstruáljuk meg a véges automatát a fordítás eredményeként adódó nyelvhez.*

Megoldás: Először a fordítót átalakítjuk, hogy ne legyen egy lépésben írás és olvasás is. Az adott automata esetében, hogy kevesebb állapot legyen, az újonnan bevezetett, állapotok közül az ekvivalenseket rögtön összevonjuk. Most is S a kezdőállapot. Az így kapott módosított fordító és egy, az L nyelvhez tartozó determinisztikus teljes véges automata:

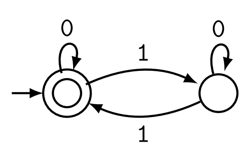




A konstrukció alapján az új automatának legfeljebb $4 \cdot 5 = 20$ állapota lehet. A kezdőállapotot, a két kezdőállapotból álló $(00, S)$ pár alkotja.



A szokásos determinizáló eljárást végrehajtva a kapott automata ez lesz:



Belegondolva, ez valóban jó, hiszen ha a fordítás kimenetén megjelenik egy `1`, akkor, mivel az L nyelvben `a` és `b` betűből is páros sok van, kell legyen az `1`-nek egy párja is. □

Az előző tételhez hasonló igaz a CF nyelvekre is.

5. Tétel. *Legyen $L \subseteq \Sigma^*$ egy környezetfüggetlen nyelv és M egy véges fordító. Ekkor az $F_M(L) \subseteq \Delta^*$ nyelv is környezetfüggetlen.*

Bizonyítás vázlat: Itt a bizonyítás egyszerűbb, mint a reguláris nyelveknél, mert felhasználhatjuk azt, hogy egy CF nyelvhez van egy állapotú üres veremmel elfogadó veremautomata. Ezért itt az állapotok a véges fordító állapotainak felelnek meg.

A konstrukció annak a másolása, ahogy CF nyelvtanból egy üres veremmel elfogadó, egyetlen állapotból álló veremautomatát készítünk.

Az egyszerűség kedvéért tegyük fel, hogy a bemeneti és a kimeneti ábécé diszjunkt, $\Sigma \cap \Delta = \emptyset$.

Az első lépésben berakjuk a verembe a CF nyelvtan kezdőváltozóját. Ez után egy lépésben

levezetés: ha egy A változó van a verem tetején, akkor ezt helyettesítjük az A-hoz tartozó jobb oldalak egyikével.

M_f **olvasna:** ha $a \in \Sigma$ van a verem tetején, ezt kiveszi és a véges fordító átmenete szerint állapotot vált (a bemenetről nem olvas).

M_f **írna:** ha $b \in \Delta$ betűt írna és ez a bemenet aktuális karaktere, akkor ezt elolvassa, a bemeneten továbblép, a véges fordító átmenete szerint állapotot vált (a verem nem változik).

$\square$

2.2. Veremfordító

A veremfordító alapgondolata hasonló a véges automatához, egy veremautomatát látunk el egy kimeneti szalaggal. A kimenetet itt is az átmeneti függvényben adjuk meg.

7. Definíció. *A* veremfordító *egy olyan $M = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, \delta)$, ahol*

- *Q egy véges, nem üres halmaz, az automata* állapota*inak halmaza.*
- *Σ egy véges, nem üres halmaz, az automata bemeneti* ábécé*je.*
- *Γ egy véges, nem üres halmaz, a* verem ábécé*je vagyis a lehetséges veremszimbólumok halmaza.*
- *Δ egy véges, nem üres halmaz. Ez az automata* kimeneti ábécé*je.*
- *$q_0 \in Q$ a* kezdő állapot*.*
- *$Z_0 \in \Gamma$ a verem* kezdő szimbóluma*.*
- *δ az* állapotátmeneti függvény*, $\delta(q, a, A) \subseteq Q \times \Gamma^* \times \Delta^*$, ahol $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ és $A \in \Gamma \cup \{\varepsilon\}$.*

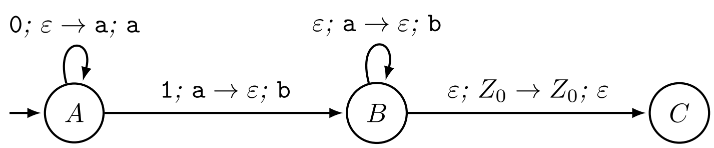
A veremfordító *működése* egy adott $w \in \Sigma^*$ bemeneten: kezdéskor a q_0 kezdő állapotban van, a veremben egyetlen szimbólum, a Z_0 található. Minden lépésben az aktuális állapot, a w soron következő karaktere és a veremben levő legfelső szimbólum alapján változik az állapot, a verem tetejére berakunk egy, a Γ karaktereiből álló véges hosszú sorozatot és a kimeneten megjelenik egy, a Δ karaktereiből álló véges hosszú sorozat. Pontosabban, ha az i-edik lépés előtt r_{i-1} állapotban van az automata, akkor a bemeneti w szóból 0 vagy 1 karaktert ($a_i \in \Sigma \cup \{\varepsilon\}$) olvas, a verem tetejéről levesz 0 vagy 1 szimbólumot ($A_{i-1} \in \Gamma \cup \{\varepsilon\}$). Az $(r_i, t_i, \beta_i) \in \delta(r_{i-1}, a_i, A_{i-1})$ átmenetnél az automata az r_i állapotba kerül, a verem tetejére (A_{i-1} helyére) kerül a $t_i \in \Gamma^*$ szimbólumsorozat, és $\beta_i \in \Delta^*$ kerül a kimeneti szalagra. Kezdéskor $r_0 = q_0$, $A_0 = Z_0$ vagy $A_0 = \varepsilon$.

A fordítás akkor érhet *véget*, ha az automata a bemenetet végigolvasta.

Egy számítás kimenetét ugyanúgy definiáljuk, mint a véges fordítónál, A kimenet az üres szóra ε, nem üres szavaknál a számítás egyes lépései során keletkezett kimenetek összefűzve, feltéve, hogy a fordítás véget ért. Egyébként a fordítás nem definiált.

A veremfordítót is megadhatjuk gráffal, de most az átmenetet jelentő nyílon nem csak a bemeneti karaktert és a verem változását, hanem a kimenetet is fel kell tüntetni.

3. Példa. *Tekintsük az alábbi veremfordítót!*



Itt $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, $\Gamma = \{Z_0, \mathtt{a}, \mathtt{b}\}$, $\Delta = \{\mathtt{a}, \mathtt{b}\}$. *A fordító az A állapotban marad, amíg* 0*-kat olvas, és ilyenkor mindegyikre berak egy* a *betűt a verembe, továbbá a kimenetre is ír egy* a *betűt. Egy* 1 *karaktert látva átkerül a B állapotba, ahonnan kezdve többet már nem olvas a bemenetről, a veremből egyenként kiszedi a berakott* a *karaktereket, mindegyiknél egy-egy* b *karaktert ír a kimenetre. Ha a verem kiürült, akkor a C állapotba lép, ahol a fordítás véget ér. Ez a fordítás csak a $\mathtt{0}^n$ és a $\mathtt{0}^n\mathtt{1}$ alakú szavakon ér véget, különben elakad, tehát csak az ilyen alakú szavaknak van fordítása, $F_M(\mathtt{0}^n) = \mathtt{a}^n$ ha $n \geq 0$, $F_M(\mathtt{0}^n\mathtt{1}) = \mathtt{a}^n\mathtt{b}^n$, ha $n \geq 1$.*

1. Állítás. *Egy veremfordító reguláris nyelvből készíthet nem reguláris nyelvet.*

Bizonyítás: Az előző példában szereplő veremfordító a $\mathtt{0}^n\mathtt{1}$ szót az $\mathtt{a}^n\mathtt{b}^n$ szóra fordítja, tehát az $L = \{\mathtt{0}^n\mathtt{1} : n \geq 1\}$ reguláris nyelv fordítása az $F_M(L) = \{\mathtt{a}^n\mathtt{b}^n : n \geq 1\}$ nem reguláris nyelv. $\square$

4. Megjegyzés. *Az is igaz, hogy egy veremfordítónál egy környezetfüggetlen L nyelv $F_M(L)$ fordítása lehet nem környezetfüggetlen. Például az előző példa módosításával lehet ilyet csinálni. (Hogyan?)*

2.3. Fordítás nyelvtannal

Egy fordítást nem csak automatával, de nyelvtannal is le lehet írni. Erre itt két lehetőséget mutatunk, melyek szoros kapcsolatban vannak az automatával történő fordítással.

Az első módszer az eredeti szó és fordításának egyfajta „párhuzamos" generálását teszi lehetővé.

8. Definíció. *Egy* egyszerű szintakszis-vezérelt fordítási séma *egy* $G = (V, \Sigma, \Delta, S, P)$ *helyettesítési szabályrendszer, amiben* V *a változók halmaza,* Σ*,* Δ *két (nem feltétlenül különböző) ábécé,* S *a kezdőváltozó,* P *a szabályok halmaza. Minden szabály bal oldala egy változó, a jobb oldala két részből áll, azaz* $A \to \alpha_1, \alpha_2$ *alakú, ahol*

- α_1*-ben a változókon kívül a* Σ *ábécé elemei,* α_2*-ben a* Δ *elemei fordulhatnak elő.*
- α_1*-ben és* α_2*-ben ugyanazok a változók szerepelnek, a sorrendjük is azonos. azaz*

$$A \to x_0 A_1 \cdots x_{k-1} A_k x_k, \quad y_0 A_1 \cdots y_{k-1} A_k y_k \quad x_i \in \Sigma^*, y_i \in \Delta^*$$

5. Megjegyzés. *Ha minden szabály jobb oldalából csak az első részt tekintjük* ($A \to \alpha_1$)*, akkor lényegében egy CF nyelvtant kapunk. Ugyanez igaz akkor is, ha minden szabály jobb oldalából csak a második részt* ($A \to \alpha_2$) *tekintjük.*

Egy ilyen rendszernél az első szabály bal oldalán levő kezdőváltozóval indul egy levezetés. Minden lépésben kiválasztunk egy A_i változót, egy ahhoz tartozó $A_i \to \alpha_1, \alpha_2$ szabályt. A jobb oldal első részében az A_i változót α_1-gyel, a második részben pedig α_2-vel helyettesítjük.

A levezetésnek akkor van vége, amikor a karaktersorozatból elfogytak a változók. Ekkor egy szópárt kapunk, ez az F fordítás egy eleme. (A két részben egyszerre fogynak el a változók. Miért?)

Az ilyen módon együtt levezetett szópárok halmaza alkotja a G-hez tartozó $F_G \subseteq \Sigma^* \times \Delta^*$ fordítást.

6. Megjegyzés. *Az általánosabb, szintaxis-vezérelt fordítási sémákban a két részben szereplő változók ugyan megfelelnek egymásnak, de a sorrendjük lehet különböző is, azaz pl.* $A \to \mathtt{a}AB; \mathtt{a}BA$ *is lehet érvényes szabály. De így a séma már nem egyszerű.*

4. Példa. *Ez a séma a szokásos és a lengyel posztfix jelölésű aritmetikai formulák levezetését teszi lehetővé.*

$$E \to E + T,\ ET+ \mid T,\ T \qquad T \to T * F,\ TF* \mid F,\ F \qquad F \to (E),\ E \mid \mathtt{a},\ \mathtt{a}$$

Lássunk példát egy levezetésre! Minden lépésben az első változót helyettesítjük ugyanaz a szabály alapján mindkét részben (tehát ez egy bal levezetés).

$$\begin{aligned}E &\Rightarrow E+T,\ ET+ \Rightarrow T+T,\ TT+ \Rightarrow F+T,\ FT+ \Rightarrow (E)+T,\ ET+\\ &\Rightarrow (E+T)+T,\ ET+T+ \Rightarrow (T+T)+T,\ TT+T+\\ &\Rightarrow (F+T)+T,\ FT+T+ \Rightarrow (\mathtt{a}+T)+T,\ \mathtt{a}T+T+\\ &\Rightarrow (\mathtt{a}+F)+T,\ \mathtt{a}F+T+ \Rightarrow (\mathtt{a}+\mathtt{a})+T,\ \mathtt{aa}+T+\\ &\Rightarrow (\mathtt{a}+\mathtt{a})+F,\ \mathtt{aa}+F+ \Rightarrow (\mathtt{a}+\mathtt{a})+\mathtt{a},\ \mathtt{aa}+\mathtt{a}+\end{aligned}$$

Vagy például a levezetés során alkalmazott szabályok sorozatát is le lehet így gyártani. Ehhez a következő szintakszis-vezérelt fordítási séma használható:

5. Példa.

$$E \to E+T,\ 1ET \mid T,\ 2T \quad T \to T*F,\ 3TF \mid F,\ 4F \quad F \to (E),\ 5E \mid \mathtt{a},\ 6$$

Az előzőnek megfelelő levezetés

$$\begin{aligned}E &\Rightarrow E+T,\ 1ET \Rightarrow T+T,\ 12TT \Rightarrow F+T,\ 124FT \Rightarrow (E)+T,\ 1245ET\\ &\Rightarrow (E+T)+T,\ 12451ETT \Rightarrow (T+T)+T,\ 124512TTT\\ &\Rightarrow (F+T)+T,\ 1245124FTT \Rightarrow (\mathtt{a}+T)+T,\ 12451246TT\\ &\Rightarrow (\mathtt{a}+F)+T,\ 124512464FT \Rightarrow (\mathtt{a}+\mathtt{a})+T,\ 1245124646T\\ &\Rightarrow (\mathtt{a}+\mathtt{a})+F,\ 12451246464F \Rightarrow (\mathtt{a}+\mathtt{a})+\mathtt{a},\ 124512464646\end{aligned}$$

Az első részben megkaptuk a szót, a másodikban pedig az alkalmazott szabályok sorszámait, az alkalmazás sorrendjében.

6. Tétel. *Ha $F_G \subseteq \Sigma^* \times \Delta^*$ a G egyszerű szintakszis-vezérelt fordítási séma által generált fordítás, akkor van olyan M veremfordító, amire $F_G = F_M$.*

Bizonyítás: A bizonyítás ahhoz hasonlít, ahogy egy CF nyelvtanból üres veremmel elfogadó veremautomatát készítettünk.

Először tegyük fel, hogy a bemeneti és a kimeneti ábécé diszjunkt, $\Sigma \cap \Delta = \emptyset$. A veremfordítónak egy állapota (q) lesz, bemeneti ábécéje Σ, a kimeneti Δ, a veremábécé legyen $\Gamma = \Sigma \cup \Delta$. Az átmeneti szabályok

- $\delta(q, \varepsilon, Z_0) = \{(q, SZ_0, \varepsilon)$.
- Minden A változóhoz $\delta(q, \varepsilon, A) = \{(q, x_0y_0A_1x_1y_1A_2 \cdots A_kx_ky_k, \varepsilon) : A \to x_0A_1 \cdots A_kx_k,\ y_0A_1 \cdots A_ky_k$ egy szabály $\}$.
- Minden $\mathtt{a} \in \Sigma$ karakterhez $\delta(q, \mathtt{a}, \mathtt{a}) = \{(q, \varepsilon, \varepsilon)\}$.
- Minden $\mathtt{b} \in \Delta$ karakterhez $\delta(q, \varepsilon, \mathtt{b}) = \{(q, \varepsilon, \mathtt{b})\}$.

Az első szabállyal berakjuk a kezdőváltozót a verembe. A második segítségével szimuláljuk a levezetést a veremben. Amikor egy változó van a verem tetején, akkor egy hozzá tartozó szabály két jobb oldalát a fentiek szerint összefésülve rakjuk a verembe. Ha egy Σ-beli karakter van a verem tetején, akkor ez egy szabály első részéhez, azaz a bemenethez tartozik. Amennyiben ez a karakter következik a bemeneten is, akkor eddig a sémával az első részben generált szó megfelel a bemenetnek, a bemeneten tovább lépünk. Ha pedig egy Δ-beli elem van a verem tetején, akkor ez egy szabály második részéhez tartozik, kiírjuk kimenetre. Ha ezzel az eljárással eljutunk a bemenet végére, akkor az első részben a bemeneten levő x szót generáltuk. Közben kiírjuk a második részben ezalatt generált szót, azaz egy megfelelő kimenetet.

A veremfordító számítása elakad, ha egy levezetés elakad, ilyenkor nem kapjuk meg a bemenet fordítását.

Amennyiben a Σ és Δ ábécéknek van közös eleme, akkor a fenti konstrukcióban nem tudhatjuk, hogy a verem tetején levő karaktert a bemenet vagy a kimenet részként kell tekinteni. Ilyenkor vegyünk egy $\tau : \Delta \to \Delta'$ kölcsönösen egyértelmű megfeleltetést egy olyan Δ' ábécével, aminek már nincs közös eleme a Σ-val. Ilyenkor ne az y_i-k karaktereit, hanem azok τ-nál vett képeit tegyük a verembe, a kiíráskor pedig ne ezeket, hanem az eredeti Δ-beli karaktereket írjuk ki. $\square$

7. Tétel. *Minden M veremfordítóhoz van olyan G egyszerű szintakszis-vezérelt fordítási séma, mely által generált párok halmazára $F_G = F_M$ teljesül.*

A tétel bizonyítása nem túl egyszerű, annak egy variánsa, ahogy veremautomatához CF nyelvtant lehet készíteni.

2.4. Fordítás jellemző nyelvtana

Most a fordítóknak egy másik nyelvtanos leírása következik. Ehhez kell egy kis előkészület. Itt nem egyszerre generáljuk a szópár két tagját, hanem mindkettőt egy közös szóból származtatjuk.

Legyen Γ, Σ, Δ három, nem feltétlenül különböző ábécé, $L \subseteq \Gamma^*$ egy nyelv.

Tetszőleges $h_1 : \Gamma \to \Sigma \cup \{\varepsilon\}$ és $h_2 : \Gamma \to \Delta \cup \{\varepsilon\}$ függvényekkel is definiálhatunk egy fordítást, ha előbb természetes módon kiterjesztjük ezeket a függvényeket a szavakra. Legyen $h_i(\varepsilon) = \varepsilon$ és egy $n > 0$ karakterből álló szóra $h_i(a_1 a_2 \cdots a_n) = h_i(a_1)h_i(a_2)\cdots h_i(a_n)$. Igy minden $x \in \Gamma^*$ szóból tudunk egy $h_1(x) \in \Sigma^*$ és egy $h_2(x) \in \Delta^*$ szót kapni. Ezeket is tekinthetjük úgy, hogy az egyik a másik fordítása.

9. Definíció. *Az $F \subseteq \Sigma^* \times \Delta^*$ fordításnak a $G = (V, \Gamma, S, P)$ nyelvtan egy* jellemző nyelvtana*, ha van olyan $h_1 : \Gamma \to \Sigma \cup \{\varepsilon\}$ és $h_2 : \Gamma \to \Delta \cup \{\varepsilon\}$ függvény, amire $F = \{(h_1(x), h_2(x)) : x \in \Gamma^*\}$.*

6. Példa. *Az $S \to \mathtt{a}S \mid \mathtt{b}S \mid \mathtt{a} \mid \mathtt{b}$ nyelvtanhoz legyen $h_1(\mathtt{a}) = \mathtt{0}$, $h_1(\mathtt{b}) = \mathtt{1}$ és $h_2(\mathtt{a}) = \mathtt{1}$, $h_2(\mathtt{b}) = \mathtt{0}$.*

A nyelvtanból $\{\mathtt{a}, \mathtt{b}\}^*$ minden eleme levezethető. Egy tetszőleges szóban h_1 az a*-kat* 0*-val, a* b*-ket* 1*-gyel helyettesíti, a h_2 meg épp fordítva működik. Tehát az ehhez tartozó fordítás $F = \{(w, \overline{w}) :$ $\overline{w}$-ben ott van* 0*, ahol w-ben* 1*, és fordítva $\}$. Mivel itt a szavaknak nincs több fordítása, használhatjuk az $f(w) = \overline{w}$ jelölést is.*

Belátható a következő kapcsolat az automatával történő fordítással.

8. Tétel. *Az F fordításhoz akkor és csak akkor van*
(a) véges fordító, ha van az F-et jellemző reguláris nyelvtan.
(b) veremfordító, ha van az F-et jellemző környezetfüggetlen nyelvtan.

Érdemes megjegyezni, hogy ezek a tételek a nemdeterminisztikus automatára épülő fordítókra igaz, a determinisztikus fordítók a fordításoknak egy szűkebb részhalmazát tudják előállítani.

7. Megjegyzés. *A különbséget mutatják a következő (többnyire nem túlságosan nehéz) tulajdonságok is:*
A véges fordítások zártak az unióra, de nem zártak a metszetre, komplementerre, míg a determinisztikus véges fordítók az unióra sem zártak.
Ugyanez igaz a veremfordításokra is.

Az, hogy egy x szónak az y egy fordítása-e, mind a négy típusnál (determinisztikus/nemdeterminisztikus véges fordító/veremfordító) algoritmikusan eldönthető.

De pl. az, hogy adott $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Delta^*$ nyelvekre $F_M(L_1) \subseteq L_2$ teljesül-e a (determinisztikus) véges fordítókra algoritmikusan eldönthető, de veremfordítókra nem.

Az pedig, hogy két fordítás értékkészlete ugyanaz-e, determinisztikus fordítóra (véges vagy verem-) eldönthető, de nemdeterminisztikusokra nem.

3. Turing-gép

A kimenettel rendelkező Turing-gépekről a tudnivalók a NyAu jegyzet Számolós Turing-gépek fejezetében megtalálhatók.

. . . .