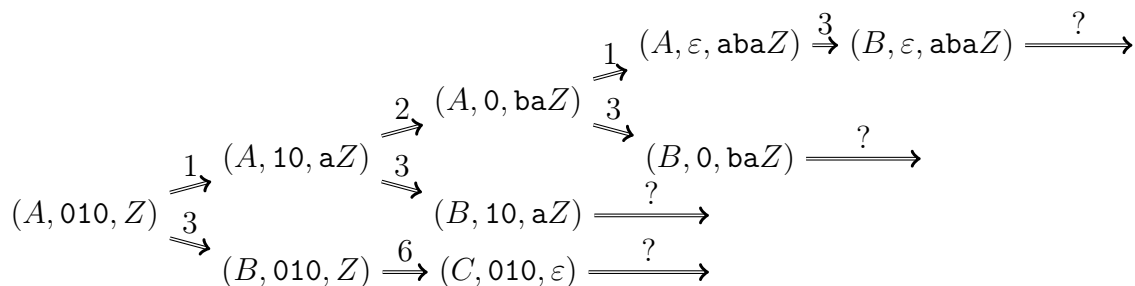


1. Legyen az ábécé $\Sigma = \{0, 1\}$, a veremautomata állapotai $Q = \{A, B, C\}$, amiből A a kezdő állapot és C az egyetlen elfogadó állapot, Z a verem kezdő szimbóluma. A veremautomata állapotátmeneti függvénye: $\delta(A, 0, \varepsilon) = \{(A, a)\}$, $\delta(A, 1, \varepsilon) = \{(A, b)\}$, $\delta(A, \varepsilon, \varepsilon) = \{(B, \varepsilon)\}$, $\delta(B, 0, a) = \{(B, \varepsilon)\}$, $\delta(B, 1, b) = \{(B, \varepsilon)\}$, $\delta(B, \varepsilon, Z) = \{(C, \varepsilon)\}$.

- Adja meg a lehetséges számításokat a 010 szó esetén!
- Elfogadja az automata a 0110 szót?
- Mi az automata által elfogadott nyelv?
- Adjon meg ehhez a nyelvhez egy környezetfüggetlen nyelvtant!

Megoldás: Az alábbi számítási fából látszik, hogy a veremautomata nem fogadja el a 010 szót.



(b) A 0110 szót elfogadja, mert ha az első két lépésben az A -ban maradunk, a veremben (felülről nézve) baZ lesz. Ha itt átmegyünk a B állapotba, akkor a 3. karakterre a b betűt, a negyedikre meg az a betűt kieszedhetjük a veremből. Amikor a bemeneti szó elfogyott, akkor a veremben csak a Z van, és ekkor átlépünk a C elfogadó állapotba.

(c) A veremautomata a páros hosszú palindromokat fogadja el. A (b) pont mintájára könnyű látni, hogy ezeket tényleg elfogadja. Másrészt, egy elfogadó számítás során amíg az A állapotban van, a bemeneti $w = a_1a_2 \dots a_n$ szó elejének megfelelően rak a és b betűket a verembe. Mivel a C -ben már nem olvas, a B állapotban kell, hogy a szó további részét megvizsgálja. Ha ide k darab karakter A -beli feldolgozása után kerül, akkor abban az esetben nem akad el, ha $a_{k+1} = a_k$, $a_{k+2} = a_{k-1}$, stb. Ahhoz, hogy a verem pont akkor ürüljön ki, amikor a szó végére ér az kell, hogy $n = 2k$ teljesüljön.

Vegyük észre, hogy ez a veremautomata az üres szót is elfogadja (ami jó, mert az is páros hosszú palindrom).

(d) Egy lehetőség: $S \rightarrow aSa \mid bSb \mid \varepsilon$

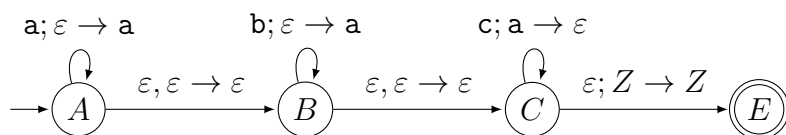
2. Készítsen veremautomatát a palindromok nyelvéhez!

Megoldás: Az előzőt annyiban kell módosítani, hogy lehessen egy tetszőleges „középső” karakter is. Ezt megoldja, ha az A állapotból a B -be egy karakter olvasásával a verem változtatása nélkül is át tudunk lépni, azaz az előző feladat átmeneti szabályainál az első kettőt így módosítjuk: $\delta(A, 0, \varepsilon) = \{(A, a), (B, \varepsilon)\}$, $\delta(A, 1, \varepsilon) = \{(A, b), (B, \varepsilon)\}$.

3. Adjon veremautomatát az alábbi nyelvekhez!

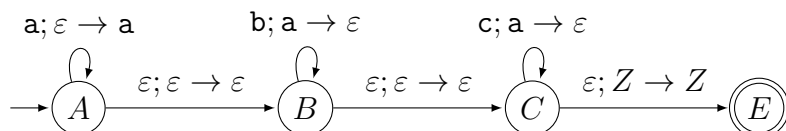
- $L_a = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ és } i + j = k\}$
- $L_b = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ és } j + k = i\}$
- $L_c = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ és } i + k = j\}$
- $L_d = \{a^i b^j c^k \mid i, j, k \geq 0; i = j \text{ vagy } i = k\}$

Megoldás: (a) Az ötlet az, hogy amikor a vagy b karakter jön, akkor mondjuk mindig egy a -t rakunk a verembe, a c -kre meg egyet kieszedünk. Akkor fogadunk el, ha a szó végén csak a Z marad a veremben. Persze arra is vigyáznunk kell, hogy b után nem jöhet a , c után meg már csak c jöhet akkor, ha elfogadjuk a szót. Lesz tehát 4 állapot, az A -ban az a -knál, a B -ben a b -knél, a C -ben a c -knél tartunk. Innen léphetünk át az E állapotba, ami az egyetlen elfogadó állapot.

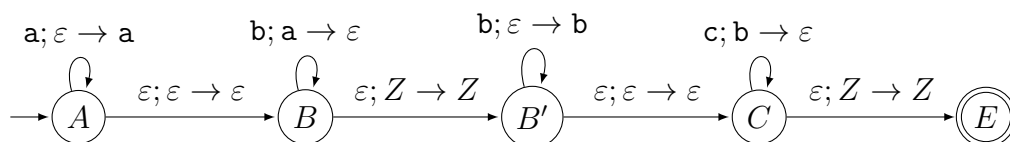


A különböző állapotok közötti $\varepsilon; \varepsilon \rightarrow \varepsilon$ átmenetek biztosítják azt is, hogy az üres szót ($i = j = k = 0$) elfogadja.

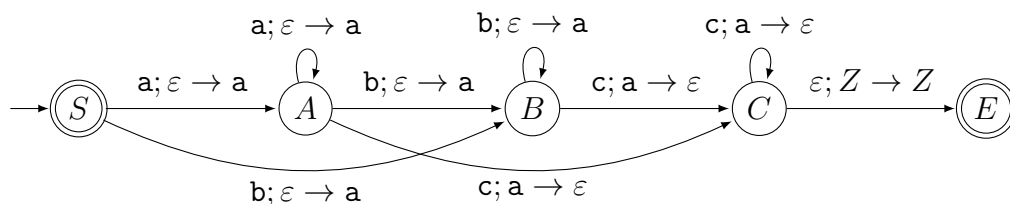
(b) Hasonló az előzőhöz, de most csak az **a**-kra rakunk be egy-egy **a**-t a verembe, **b**-re és **c**-re is kiveszünk egyet.



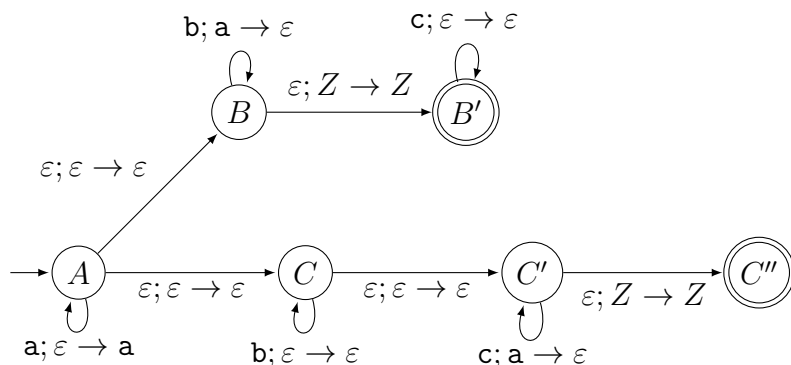
(c) Az **a**-kra megint betehetünk egy-egy **a**-t a verembe, de a **b**-knél ki kell ezeket szedni. Ha közben a verem kiürül, akkor onnantól kezdve mondjuk **b**-ket rakunk be, amiket majd a **c**-k olvasásakor szedegetünk ki.



Megjegyzés: a fent megadott veremautomaták mindegyike nemdeterminisztikus az új állapotba vezető átmenetek miatt, de ezekhez a nyelvekhez lehet determinisztikus veremautomatát is adni. Pl. az első esetben:



(d) Az **a**-kra megint betehetünk egy-egy **a**-t a verembe, de utána nemdeterminisztikusan vagy a **b**-k vagy a **c**-k számával vetjük össze a verem tartalmát. Vegyük észre, hogy azokat a szavakat is el kell fogadni, amiben nulla darab **a** és **c** van, és bármennyi **b**.



4. Adjon veremautomatát az alábbi nyelvekhez!

(a) $L_a = \{a^n b^m \mid 2n = m \geq 1\}$

(b) $L_b = \{a^n b^m \mid 2n \geq m \geq n \geq 1\}$

(c) $L_c = \{a^n b^m \mid 3n = 2m\}$

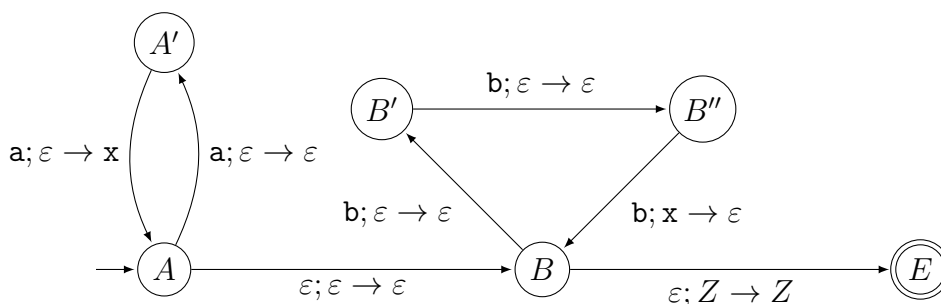
Megoldás: Itt most már csak szavakban vázoljuk a veremautomaták működését.

(a) A kezdő állapotból az első **a** olvasásakor két darab **a** betűt rak a verembe és átmegy az **A** állapotba. Itt minden további **a** betűnél megint két darab **a** betűt rak a verembe. Az **A**-ból az első **b** olvasásakor kivesz a veremből egy **a**-t és átlép a **B** állapotba, ahol minden további olvasott **b** betűnél kivesz egy **a**-t a veremből. Végül olvasás nélkül a verem alját jelző **Z**-t látva átlép a **C**

elfogadó állapotba, ahol a számítás véget ér. Világos, hogy ha $m = 2n$, akkor a szó végén az elfogadó állapotba juthatunk. Ha pedig rosszkor váltunk állapotot vagy a szó nem megfelelő, akkor vagy nem jutunk el a C -ig vagy a szó addigra nem ér véget, a számítás mindenképpen elakad, és ezért nem elfogadó.

(b) Kihaszználjuk a nemdeterminizmus adta lehetőségeket: az előző automatát úgy módosítjuk, hogy az A állapotban minden a betűre vagy egy vagy kettő a -t rakunk a verembe. Ha egy szó az L_b nyelvben van, akkor lesz hozzá elfogadó számítás (amikor $(2n - m)$ -szer rakunk be egyet és $(m - n)$ -szer kettőt). Ha az m túl kicsi, a verem nem fog kiürülni, ha meg nagy, akkor nem tudunk kivenni belőle, amikor kellene, a számítás elakad.

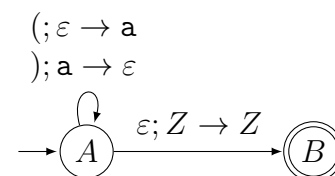
(c) Először minden második a után egy x -et teszünk a verembe, majd minden harmadik b után kivesszünk egy x -et, hiszen azt kell ellenőrizni, hogy teljesül-e $n/2 = m/3$.



5. Készítsen veremautomatát a jó zárójelezések nyelvéhez!

Megoldás:

Az ötlet hasonló az előzőekhez, a nyitó zárójelnél berakunk egy a betűt a verembe, a csukónál kivesszünk, különbség, hogy csukó zárójel után is jöhet nyitó, ezért nem kell annyi állapot. Az, hogy egy jó zárójelsorozattal van dolgunk, pontosan azt jelenti, hogy csukó zárójelnél mindig van legalább egy a a veremben de a szó végére egy sem marad.



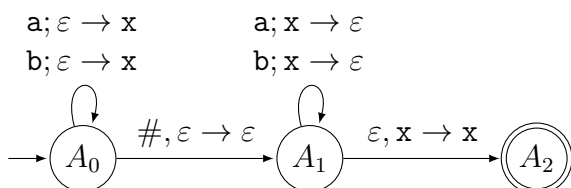
6. Adjon veremautomatát az alábbi nyelvhez!

$$\{x\#y \mid x, y \in (a + b)^*; x \neq y\}$$

Megoldás: Ez a nyelv a következő három nyelv uniója:

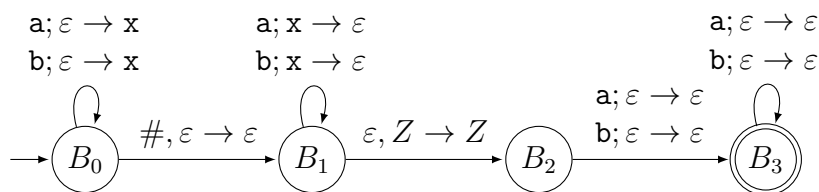
- A : olyan $x\#y$ szavak, melyben x hosszabb, mint y .
- B : olyan $x\#y$ szavak, melyben y hosszabb, mint x .
- C : olyan $x\#y$ szavak, melyben x és y bármilyen hosszúak, de valamilyen i -re $x_i \neq y_i$.

Konstruálunk mindháromhoz egy-egy veremautomatát.

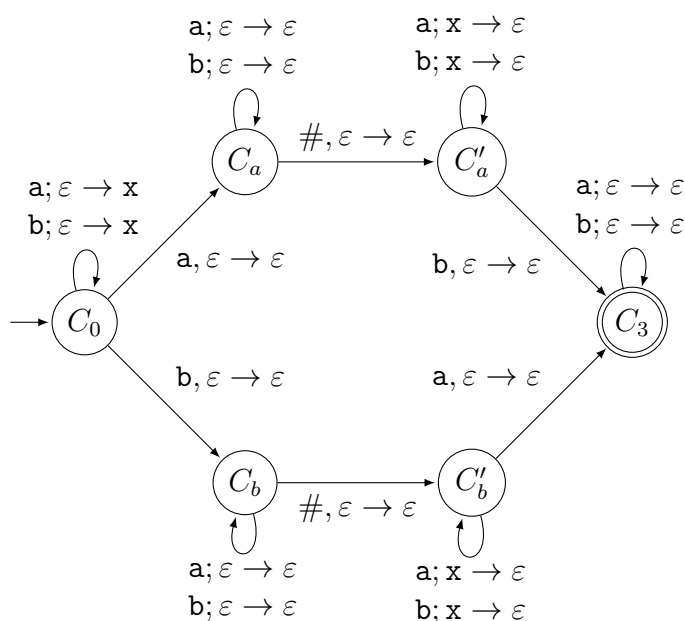


Ez az A nyelv minden szavát elfogadja: először x minden betűjére egy x -et tesz a verembe, majd a $\#$ után y minden betűjére ki akar egyet venni. De mivel y rövidebb, az utolsó karakter után A_2 állapotban lesz, ami nem elfogadó. De mivel a verem tetején x lesz, ezért átléphet A_2 állapotba.

Ha egy szót elfogad ez a veremautomata, akkor az utolsó átmenet csak A_1 -ből A_2 -be léphetett, és ekkor egy x volt a verem tetején. Az előző gondolatmenetből látszik, hogy ez azt jelenti, hogy x hosszabb, mint y .



Az előző esethez hasonlóan működik ez az automata is. De amikor kiürül a verem, akkor nem kerülhetünk rögtön elfogadó állapotba, csak úgy, ha még nem értünk az input végére, azaz olvasunk még legalább egy karaktert.



Itt a C_0 állapotban (nemdeterminisztikusan) néhány karaktert elolvasunk x -ből és mindegyikhez beteszünk egy-egy x -et a verembe. Minden lépés után van egy másik nemdeterminisztikus választás is: ha pl. a a következő karakter, akkor maradhatunk C_0 -ben is, de átléphetünk C_a -ba is. Tegyük fel, hogy amikor elmozdulunk C_0 -ból, akkor $i - 1$ darab x van a veremben. Ha C_a -ba lépünk, akkor azt jelenti, hogy $x_i = a$. Ezután elolvassuk x maradék részét (nem változtatunk a verem tartalmán), majd a $\#$ után C'_a -ba lépünk. Itt a verem kiürítésével elolvassuk y első $i - 1$ karakterét. Ha a verem kiürült és a következő karakter b , akkor $y_i = b$, ezért bármit is olvasunk ezután, elfogadjuk az inputot. Hasonlóan látható, hogy ha nincs olyan i , amire $x_i \neq y_i$, akkor a veremautomatának nincs elfogadó számítása. (Pl. $aaa\#aa$ esetén.)

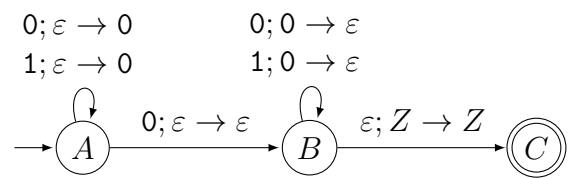
Végül a fenti veremautomatából egy újat készítünk. Az S kezdőállapotból vezessen egy-egy $\varepsilon; \varepsilon \rightarrow \varepsilon$ átmenet az A_0, B_0, C_0 állapotokba. Könnyen látható, hogy ez a három nyelv unióját fogja elfogadni.

7. Álljon L azokból a $\{0, 1\}$ feletti sorozatokból, melyekben a középső karakter 0. Igazolja, hogy L környezetfüggetlen nyelv!

Megoldás: Két lehetőségünk is van: megadhatunk egy környezetfüggetlen nyelvtant a nyelvhez vagy egy veremautomatát.

A nyelvtan ötlete, hogy ha $w \in L$ legalább 3 hosszú, akkor az első és utolsó karakterét levágva a maradék is L -ben lesz (és ez a két karakter tetszőleges). Ha 3-nál rövidebb, akkor meg csak $w = 0$ lehet. Egy ezt tükröző nyelvtan: $S \rightarrow 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 0$.

A veremautomata meg működhet úgy, hogy 0-t rak a verembe, amíg el nem éri a közepét (ezt nem kell tudni, mikor következik be, erre jó a nemdeterminizmus!), ha ez 0, akkor a továbbiakban kiszedegeti a 0-kat a veremből. Akkor fogad el, ha a szó végénél a veremből is épp kifogytak a 0 karakterek.



8. Legyen L_r egy tetszőleges reguláris nyelv és legyen L_c egy tetszőleges környezetfüggetlen nyelv.

(a) Mutasson olyan példát, amikor $L_r \cap L_c$ nem reguláris!

(b) Igazolja, hogy $L_r \cap L_c$ mindig környezetfüggetlen!

Megoldás: (a) Ha $L_r = \{a, b\}^*$ és $L_c = \{a^n b^n \mid n \geq 0\}$, akkor $L_r \cap L_c = L_c$, ami nem reguláris.

(b) Legyen $M_1 = (Q_1, \Sigma, q_1, F_1, \delta_1)$ egy DVA az L_r nyelvhez, és $M_2 = (Q_2, \Sigma, \Gamma, q_2, Z_0, F_2, \delta_2)$ egy veremautomata az L_c nyelvhez. A kettőből elkészíthetjük azt az M veremautomatát, amelynek állapothalmaza $Q_1 \times Q_2$, kezdőállapota (q_1, q_2) , elfogadó állapotainak halmaza $F_1 \times F_2$, az átmeneteiben az állapot első koordinátájában δ_1 szerint lép, a másodikban δ_2 szerint. Amennyiben a veremautomata nem olvas a bemenetről, akkor az állapot első koordinátája nem változik (M_1 nem lép). Ha a veremautomata nem tud lépni, akkor M számítása elakad. A veremben mindig történjen az, ami az M_2 megfelelő lépésében történik.

Ez az automata lényegében párhuzamosan futtatja az M_1 és M_2 automatát, akkor fogad el, ha mindkettő elfogadó.