6. tétel

Gráf fogalma és megadásának módjai. Egyszerű, irányított és irányítatlan gráfok. Séta, út, összefüggőség. Nevezetes gráfok: páros gráf, teljes gráf, fa, kör, súlyozott gráf. Generatív nyelvtanok, nyelvosztályok, a Chomsky-hierarchia. Véges automaták, lineáris idejű felismerés, veremautomaták.

A gráf fogalma

A **gráf** a és a egyik alapvető fogalma. A gráf csomópontok, csúcsok és rajtuk értelmezett összeköttetések (élek) halmaza. Egy gráfot megadhatunk csúcsainak és éleinek felsorolásával, vagy szemléletesebben egy diagram formájában, ahol a pontok felelnek meg a gráf csúcsainak, az őket összekötő ívek pedig az éleknek. A két megadási mód ekvivalens, azaz a gráf pusztán egy struktúra, semmilyen megjelenítési információt nem tartalmaz, így különböző diagramok is tartozhatnak ugyanahhoz a gráfhoz.

Alapértelmezésben a gráf irányítatlan, azaz nem teszünk különbséget „A-ból B-be", illetve „B-ből A-ba" menő élek között. Ezzel szemben az irányított gráfokban (angolosan: digráf) a két iránynak irányított élek felelnek meg.

Szintén alapértelmezésben, a gráf csúcsai **címkézettek**, azaz meg lehet különböztetni őket. Bizonyos problémák azonban könnyebben kezelhetők, ha nem különböztetjük meg a csúcspontokat. Persze egy-egy csúcspont így is megkülönböztethető maradhat egyéb jellemzőik alapján, mint például a vele szomszédos csúcsok száma. Hasonlóan, a gráf élei alapértelmezésben címkézettek, de előfordulhat, hogy ezt nem követeljük meg. Az olyan gráfok, amikben sem a csúcspontok, sem az élek nem címkézettek, **címkézetlen** gráfok. Megjegyzés: a „címkézés" szó más kontextusban is elfordul a gráfoknál, itt most az élek-csúcsok megkülönböztetésére szolgáló címkékkel foglalkoztunk.

Definíció: Adott egy A halmaz, és egy rajta értelmezett $\rho \subseteq A \times A$ bináris (kétváltozós) reláció. Ekkor a $G = (A, \rho)$ párt, vagyis az A halmaz feletti relációs struktúrát az A halmaz feletti **gráf**nak nevezzük.

Megjegyezzük, hogy e definíció szerint a ρ reláció rendezett elempárokból áll, azaz a gráf **irányított**, viszont „többszörös" éleket nem tartalmaz, azaz **egyszerű**.

Ezen értelmezésen belül az irányítatlan gráf fogalma úgy értelmezhető, hogy megköveteljük a ρ reláció szimmetriáját, azaz hogy érvényes legyen $\forall x, y \in A : (x, y) \in \rho \Leftrightarrow (y, x) \in \rho$, és ekkor az irányítatlan gráf az irányított gráf speciális esete. Más, filozófiailag kevésbé problematikusnak látszó, de kényelmetlenebb lehetőség is van. Ld. még irányítatlan gráf.

Definíció: Az *A* halmazt a *G* = *(A,ρ)* gráf *tartóhalmaz*ának vagy *csúcshalmaz*ának mondjuk, és (az angol „vertex"=csúcs szó rövidítéseként) *V(G)*-vel jelöljük.

A gráf itt megadott fogalmának rengeteg, nemcsak a matematika, hanem a szociológia, számítástechnika stb. fejlődése által egyenesen kikényszerített általánosítása vagy változata létezik, lásd általánosítások, speciális esetek és változatok.

Alapfogalmak:

A gráf két élét **szomszédos**nak nevezzük, ha van egy közös csúcspontjuk. Hasonlóan, két csúcspont **szomszédos**, ha van egy közös élük, másként fogalmazva egy éllel vannak összekötve. Egy **séta** szomszédos csúcsok és élek váltakozó sorozata. Az önmagát nem metsző sétát **út**nak hívunk, ha első és utolsó csúcsa különbözik, illetve **kör**nek, ha ez a két csúcs megegyezik. Egy gráf **összefüggő**, ha (élei esetleges irányításáról megfeledkezve) bármely két csúcs között van út.

Az ún. **súlyozott** gráfban (ami lehet irányított gráf is), minden élhez hozzárendelünk egy értéket, ami az él költsége, súlya vagy hossza az alkalmazástól függően. Az ilyen gráfok sok helyen előfordulnak, például optimalizálási feladatokban, mint az utazo ugynok problema

Gráf megadásának módjai:

Szomszédsági listás:

$G = (V, E)$ szomszédsági listás ábrázolása során egy Adj tömböt használunk. Ez |V| darab listából áll, és az Adj tömbben minden csúcshoz egy lista tartozik. Minden $u \in V$ csúcs esetén az $Adj[u]$ szomszédsági lista tartalmazza az összes olyan v csúcsot, amelyre létezik $(u, v) \in E$ él. Azaz: $Adj[u]$ elemei az u csúcs G-beli szomszédjai. (Sokszor nem csúcsokat, hanem megfelelő mutatókat tartalmaz a lista.) A szomszédsági listákban a csúcsok sorrendje rendszerint tetszőleges.

Ha G irányított gráf, akkor a szomszédsági listák hosszainak összege $|E|$, hiszen egy (u, v) élt úgy ábrázolunk, hogy v-t felvesszük az $Adj[u]$ listába. Ha G irányítatlan gráf, akkor az összeg $2|E|$, mert (u, v) irányítatlan él ábrázolása során u-t betesszük v szomszédsági listájába, és fordítva. Akár irányított, akár irányítatlan a gráf, a szomszédsági listás ábrázolás azzal a kedvező tulajdonsággal rendelkezik, hogy az ábrázoláshoz szükséges tárterület $\theta(V + E)$.

A szomszédsági listákat könnyen módosíthatjuk úgy, hogy azokkal súlyozott gráfokat ábrázolhassunk. Például, legyen $G = (V, E)$ súlyozott gráf w súlyfüggvénnyel. Ekkor az $(u, v) \in E$ él $w(u, v)$ súlyát egyszerűen a v csúcs mellett tároljuk u szomszédsági listájában. A szomszédsági listás ábrázolás könnyen alkalmassá tehető sok gráfváltozat reprezentálására.

A szomszédsági listás ábrázolás hátránya, hogy nehéz eldönteni szerepel-e egy (u, v) él a gráfban, hiszen ehhez az $Adj[u]$ szomszédsági listában kell v-t keresni. Ez a hátrány kiküszöbölhető csúcsmátrix használatval, ez azonban aszimptotikusan növeli a szükséges tárterület méretét.

Szomszédsági mátrixos:

Ha egy $G = (V, E)$ gráfot szomszédsági mátrixszal (vagy más néven csúcsmátrixszal) ábrázolunk, feltesszük, hogy a csúcsokat tetszőleges módon megszámozzuk az 1,2, ... , $|V|$ értékekkel. A G ábrázolásáhz használt $A = (a_{ij})$ csúcsmátrix mérete $|V| \times |V|$, és

$$a_{ij} = \begin{cases} 1, & \text{ha } (i,j) \in E, \\ 0, & \text{különben} \end{cases}$$

A csúcsmátrix $\theta(V^2)$ tárterültetet foglal el, függetlenül a gráf éleinek számától. Gyakran kifizetődő a csúcsmátrixból csak a főátlóban és az efölött szereplő elemeket tárolni, ezzel majdnem felére csökkenthetjük az ábrázoláshoz szükséges tárterület méretét.

A szomszédsági listás ábrázoláshoz hasonlóan csúcsmátrixokkal is reprezentálhatunk súlyozott gráfokat. Ha $G = (V, E)$ súlyozott gráf w súlyfüggvénnyel, akkor az $(u, v) \in E$ él $w(u, v)$ súlyát a csúcsmátrix u sorában és v oszlopában tároljuk. Nem létező él esetén a mátrix megfelelő elemét nil-nek választjuk, noha sokszor célszerű ehelyett 0 vagy végtelen értéket használni.

A szomszédsági listák együttesen aszimptotikusan kevesebb tárterületet igényelnek, mint a csúcsmátrix, azonban a használat során hatékonyságban ugyanennyivel elmaradnak attól, így ha a gráf mérete nem túl nagy, akkor kedvezőbb a hatékonyabb és egyszerűbb csúcsmátrixos ábrázolást használni. Ha a gráf nem súlyozott, akkor a csúcsmátrixos ábrázolás tovább javítható. Ebben az esetben a mátrix elemei lehetnek bitek, így jelentősen csökkenthetjük a szükséges tárterület méretét.

Nevezetes séták

Tudunk-e olyan sétát tenni a gráf élein, hogy

minden élen pontosan egyszer megyünk végig? (lásd:Königsbergi hidak problemaja, Euler kor)

minden csúcsot pontosan egyszer érintünk? (lásd:Hamilton ut)

Bár a két kérdés hasonlónak tűnik, az első megválaszolására van gyors (lineáris idejű) algoritmus (néha a diagram alapján ránézésre is eldönthető), míg a második az egyik ismert legnehezebb probléma (NP teljes).

Súlyozott gráfban:

Melyik a legrövidebb (legkisebb összsúlyú) út A-*ból* B-*be?*

Például, ha egy valódi úthálózatban, ahol a csúcsok a csomópontok, az élek az útszakaszok, és *A*-ból *B*-be szeretnénk eljutni, de a különböző lehetséges utak nem egyformán kedvezőek, van rövidebb (gyorsabb, olcsóbb), akkor az élekhez az éleknek megfelelő hosszt (időtartamot, árat) rendelve, a válasz a számunkra legkedvezőbb út.

Irányítatlan gráf

A *G* irányítatlan gráfot a *G*=(*V*, *E*) rendezett párral jellemezzük, ahol

- *V* a csúcsok halmaza (melyről általában feltesszük, hogy véges) és
- *E* az irányítatlan éleknek megfelelő csúcsok rendezetlen párjainak halmaza.

Az *e*={*u*, *v*} élről azt mondjuk, hogy *u* és *v* ***között fut***, ***összeköti*** *u*-t és *v*-t.

Irányított gráf

A $\overrightarrow{G} = (V, \overrightarrow{E})$ irányított gráf:

- $\overrightarrow{E}$ az irányított élek végpontjai rendezett párjának halmaza

Az *e*=(*u*, *v*) élről azt mondjuk, hogy *u*-ból indul és *v*-be ***megy***, *v* az *u* ***közvetlen leszármazott***ja (***gyereke***), *u* a *v* ***közvetlen őse*** (***szülője***).

Ha egy irányított gráf nem tartalmaz irányított kört, akkor irányított körmentes gráfnak vagy angol nevének rövidítése szerint DAG-nak (*directed acyclic graph*) hívjuk.

Egyszerű gráf:

Az olyan gráfokat amelyekben sem többszörös élek, sem hurokélek nincsenek ***egyszerű gráf***nak hívjuk.

Ekkor *E* valódi halmaz és $E \subset \binom{V}{2}.$

Általánosítások

- A ***hurokél*** olyan él, amelynek mindkét végpontja megegyezik.
- A gráfokban megengedhetünk ***többszörös-*** vagy ***párhuzamos él***eket, melyek végpontjai megegyeznek. Ehhez az élek halmazát multihalmazra, vagy más többszöri előfordulást lehetővé tevő struktúrára kell cserélnünk. Az olyan gráfot, amiben a többszörös élek (és esetleg a hurokélek) megengedettek, **multigráf**nak vagy **pszeudográf**nak hívjuk.

- Néha olyan gráfokat is megengednek, amiben olyan élek is vannak, amiknek csak egy végük van „fél-él”, vagy egy csúcshoz sem kapcsolódnak „szabad él”, például az előjeles gráfoknál.

- Hipergráf – a hipergráfban egy él kettőnél több csúcspontot is összeköthet.
- Végtelen gráf

- Egy irányítatlan gráf felfogható szimpliciális komplexusnak, ami 1-szimplexekből (élekből) és 0-szimplexekből (csúcspontokból) áll. Ilyen értelemben ezek a komplexusok a gráfok általánosításai, mert magasabb dimenziójú szimplexeket is tartalmazhatnak.
- Minden gráfból felírható egy matroid (pontosabban: grafikus matroid), de általában a gráfot nem lehet visszaállítani a matroidjából, ezért a matroidok nem valódi általánosításai a gráfoknak.

Teljes gráf:

A **teljes gráf** olyan egyszerű gráf, amelynek minden csúcsa össze van kötve minden más csúccsal. Az *n* csúcsú teljes gráfot K_n -nel jelöljük Kazimierz Kuratowski lengyel matematikus emlékére.

Tulajdonságok[szerkesztés]

- K_n reguláris, minden csúcsának fokszáma *n - 1*.
- K_n összesen $\binom{n}{2} = \frac{n(n-1)}{2}$ élt tartalmaz.
- a Kuratowski-tétel szerint síkbarajzolható gráf nem tartalmazhat a K_5 gráffal topologikusan izomorf részgráfot.
- K_n az (n+1)-szimplex éleit adja.

Kör:

A gráfelméletben a **kör** élek olyan egymáshoz csatlakozó sorozata, amelyben az élek és pontok egynél többször nem szerepelhetnek, és a kiindulási pont megegyezik a végponttal.

Jelölések, fogalmak:

A körben szereplő élek száma a kör hossza. Az *n* hosszú kört C_n-nel jelöljük. Speciálisan C_1 neve hurokél, C_2 kétszög, C_3 pedig háromszög.

Egy kört *páros körnek* hívunk, ha a hossza páros, különben *páratlan kör*. Egy gráf akkor és csak akkor páros gráf, ha nem tartalmaz páratlan kört.

Egy gráf girthparaméterét kör segítségével definiáljuk: a gráfban található legrövidebb kör hossza.

Euler-kör:

Egy kör *Hamilton-kör*, ha minden csúcsot pontosan egyszer érint. Hamilton-kör létezésére vannak elégséges feltételek (Dirac-feltétel, Pósa-feltétel, Ore-feltétel, Chvátal-feltétel), de, mivel a „Van-e adott gráfban Hamilton-kör?" feladat NP-teljes, pontos feltétel megtalálása nem remélhető.

A Hamilton-körrel duális fogalom az *Euler-kör* (avagy Euler-vonal); ami olyan zárt élsorozat, mely minden élet pontosan egyszer tartalmaz. Egy összefüggő gráfban akkor és csak akkor létezik Euler-kör, ha minden csúcsának fokszáma páros. Euler-kör létezésével kapcsolatos a königsbergi hidak problémája.

Körmentes-gráf:

Körmentesnek (vagy *aciklikusnak*) nevezzük az olyan gráfot, amely nem tartalmaz kört. Ha egy körmentes gráf összefüggő, akkor *fa*. Mivel minden körmentes gráf fák (esetleg egyelemű) halmaza, a körmentes gráfokat *erdőnek* is nevezzük. Az irányított körmentes gráfok jelentős szerepet játszanak a számítástudományban.

Páros gráf:

Páros gráfnak, **kétrészes gráf**nak vagy **páros körüljárású gráf**nak nevezünk egy *G* gráfot, ha *G* csúcsainak halmazát fel tudjuk úgy osztani egy *A* és *B* halmazra, hogy az összes *G*-beli élre teljesül, hogy az egyik végpontja *A*-ban van, a másik pedig *B*-ben. Egy *G* páros gráfot következőképpen jelölünk: *G = (A, B)*.

Páros gráf minden részgráfja is páros. Minden fa páros gráf.

Teljes páros gráfnak nevezünk egy olyan páros gráfot melyben minden *A*-beli pont össze van kötve minden *B*-beli ponttal. Jelölés: $K_{a,b}$, ahol *a* = |*A*| és *b* = |*B*|.

Fa, fák, erdő

A **fának** vagy **fagráfnak** nevezzük azokat a , amelynek bármely két csúcsát *pontosan egy* út köti össze, azaz a fák körmentes gráfok. **Erdőnek** nevezzük azokat a gráfokat, amelynek bármely két csúcsát *legfeljebb egy* út köti össze, azaz ahogy az elnevezés is utal rá, az erdő olyan gráf, aminek komponensei fák, vagy ami ezzel ekvivalens, az erdők körmentes gráfok.

Minden fa paros graf . Minden fa, amelynek megszamlalhato sok csúcspontja van, sikgraf .

Minden összefüggő G gráfnak van feszítő fája, azaz létezik hozzá olyan fa, ami tartalmazza a G összes csúcspontját, és amelynek élei egyben a G gráfnak is élei. Továbbá minden gráfnak van feszítő erdője, tehát létezik hozzá olyan erdő, aminek a komponensei azonosak az eredeti gráféival.

Minden fának, amelynek van legalább két csúcspontja, van legalább két levele, azaz van legalább két olyan csúcsa, amelynek a fokszama 1.

Egy fa csúcsainak száma 1-gyel nagyobb az élek számánál. Erdő esetén a csúcsok és az élek számának különbsége a komponensek száma.

- Egy erdő $\chi_s(G)$ csillagkromatikus száma a fakomponensek közül a legnagyobb sugár+1, illetve 3, amelyik ezek közül a maximális.

Generatív nyelvtanok, nyelvosztályok, a Chomsky-hierarchia. Véges automaták, lineáris idejű felismerés, veremautomaták.

Generatív nyelvtanok

DEF: ***Generatív nyelvtannak*** egy $G=(N,\Sigma,P,S)$ rendezett négyest nevezünk, ahol:

- N nemterminális ábécé
- Σ terminális ábécé amire $N \cap \Sigma = \emptyset$
- $S \in N$ a kezdőszimbólum
- P pedig $\alpha \rightarrow \beta$ alakú átjárási szabályok véges halmaza, ahol $\alpha, \beta \in (N \cup \Sigma)^*$ és α-ban van legalább egy nemterminális betű.

Egy generatív nyelvtan a jelsorozatok transzformációs szabályait leíró szabályok halmazából áll. A nyelvet alkotó jelsorozatok létrehozásához szükséges, hogy legyen egy egyedi „kezdő” szimbólum, ezután csak a szabályokat kell egymás után alkalmazni (bárhányszor, tetszés szerinti sorrendben) a kezdő szimbólum átalakítására. A nyelv azokból a jelsorozatokból áll, amelyeket az említett módon elő lehet állítani. A szabályok alkalmazásának megengedett lehetőségei közül bármilyen különleges sorrend alkalmazásával az átalakításokkal létrehozhatók jelsorozatok, de ha ezek közül a jelsorozatok közül egyet is többféleképpen is elő lehet állítani, akkor a nyelvtant kétértelműnek nevezik.

Nyelvosztályok

DEF: (***Chomsky nyelvosztályok***) Azt mondjuk, hogy a $G=(N,\Sigma,P,S)$ nyelvtan:

- 0 típusú (kifejezés struktúrájú), ha rá semmilyen korlátozás nincs.
- 1 típusú (környezetfüggõ), ha P-ben minden szabály $\alpha A \beta \rightarrow \alpha \delta \beta$ alakú, ahol $\delta \neq \lambda$. Kivétel lehet az $S \rightarrow \lambda$ szabály, ekkor azonban S nem szerepelhet semelyik szabály jobb oldalán.
- 2 típusú (környezetfüggetlen), ha P-ben minden szabály $A \rightarrow \alpha$ alakú.
- 3 típusú (reguláris), ha P-ben minden szabály $A \rightarrow xB$ vagy $A \rightarrow x$ alakú.

Amikor az 1950-es években Noam Chomsky először formalizálta a generatív nyelvtanokat, akkor négy típusba sorolta be azokat. Ezek a csoportok ma Chomsky-féle hierarchiaként ismertek. Az egyes típusok között a különbség a produkciós szabályok szigorúságában jelenik meg. Két fontos típus a környezetfüggetlen nyelvtanok és a szabályos nyelvtanok. Azok a nyelvek, amelyek valamilyen nyelvtannal leírhatók azok az úgynevezett környezetfüggetlen nyelvek illetőleg a szabályos nyelvek. Habár kevésbé hatékonyak, mint a megkötés nélküli nyelvtanok, amelyek ténylegesen le tudnak írni bármilyen nyelvet, amelyek a Turing-gépekkel elfogadhatóak, ezen megkötések miatt inkább az ilyen nyelvtanok hatékonyan elemzők segítségével valósíthatók meg. Például, a környezetfüggetlen nyelvtanok a jól ismert algoritmusokat megvalósító LL elemzők és LR elemzők segítségével elemezhetők hatékonyan.

Chomsky-hierarchia

	Nyelvtanok	Nyelvek	Automaták	Produkciós szabályok
0. nyelvosztály	Rekurzívan megszámlálható	Rekurzívan megszámlálható	Turing-gép	Nincs megkötés
1. nyelvosztály	környezetfüggő	környezetfüggő	Korlátos nem determinisztikus Turing-gép	$\alpha A\beta \to \alpha\gamma\beta$
2. nyelvosztály	környezetfüggetlen	környezetfüggetlen	Nem determinisztikus veremautomata	$A \to \gamma$
3. nyelvosztály	Szabályos	Szabályos	Véges determinisztikus automata	$A \to a$ és egyébként $A \to aB$ vagy $A \to Ba$

Véges automaták

Determinisztikus véges állapotú gép

A számítástudományban a determinisztikus véges állapotú gép vagy determinisztikus véges állapotú automata (angolul deterministic finite state machine vagy deterministic finite automaton, általánosan használt rövidítéssel: DFA) egy véges állapotú gép, ahol minden állapot–bejövő szimbólum párhoz egy és csakis egy másik állapotba való átmenet tartozik.

A DFA a reguláris nyelvek halmazába tartozó nyelvek felismerésénél használható, más nyelveknél nem alkalmazható.

A DFA egy bejövő szimbólumokból álló stringgel dolgozik. Minden egyes bejövő szimbólum hatására a gép állapota az adott átmeneti függvény alapján megváltozik (vagy ugyanaz marad). Amikor az utolsó bejövő szimbólum beérkezik, azt a gép állapotától függően vagy elfogadják vagy visszautasítják.

A DFA-t tekinthetjük egy speciális Turing-gépnek, amely nem tudja az olvasófejet mozgatni, és csak előre képes mozgatni a szalagját.

Egy DFA a következő ötössel írható le: (S, Σ, T, s, A), ahol

(S) az állapotok véges halmaza

(Σ) az ábécének nevezett véges halmaz

($T : S \times \Sigma \to S$) az átmeneti függvény

($s \in S$) a kezdő állapot

($A \subseteq S$) az elfogadó állapotok halmaza.

Nemdeterminisztikus véges állapotú gép

A számítógép-tudományban a nemdeterminisztikus véges állapotú gép vagy a nemdeterminisztikus véges állapotú automata, angol terminológiával a nondeterministic finite state machine vagy nondeterministic finite automaton (NFA) egy véges állapotú gép ahol bármelyik állapot–bejövő szimbólum párhoz több következő állapot is tartozhat.

Az NFA egy bejövő szimbólumokból álló stringgel dolgozik. Minden egyes bejövő szimbólum hatására a gép állapota az adott átmenetfüggvény alapján megváltozik (vagy ugyanaz marad). Az utolsó szimbólum feldolgozása után az NFA akkor, és csak akkor fogadja el a stringet, ha az automata a néhány elfogadó állapot valamelyikébe kerül, azaz: csak akkor utasítja vissza a stringet, ha nem kerül elfogadó állapotba.

A gép nem determinisztikus:

- Lehetséges olyan belső állapota, amiből a beolvasott szimbólum hatására több lehetséges állapot egyikébe mehet át.

Például, az automata az 1-es állapotban van, és a következő bejövő szimbólum az a. Az a beolvasása nélkül a 2-es állapotba kerülne, de a bejövő a hatására 3-as vagy 4-es állapotba kerül. Azaz egy helyes belső állapotból nem determinált a következő állapot még adott input esetén sem.

- Létrejöhet állapotváltás bemenő szimbólum nélkül is.

A bejövő szimbólum nélküli állapotváltozást epszilon átmenetnek nevezik, és általában a görög ε betűvel jelölik.

Egy NFA a következő 5-össel írható le: (S, Σ, T, s_0, A), ahol

- (S) az állapotok véges halmaza
- (Σ) az ábécének nevezett véges halmaz
- (T: $S \times (\Sigma \backslash\{\varepsilon\}) \to P(S)$) az átmenetfüggvény. Ez lehet olyan reláció is, ami nem függvény.
- s_0 a kitüntetett kezdeti (vagy indító) állapotok halmaza (s_0 S)
- (A S) az elfogadó állapotok halmaza

ahol $P(S)$ S hatványhalmaza, ε az üres string, és Σ a bemenő jelek ábécéje.

Lineáris idejű felismerés

Veremautomaták

A veremautomata a véges automatának egy olyan kiterjesztése, amikor az állapotokon kívül az automata, egy vermet (LIFO tulajdonságú tárolót) is használ memóriaként. A veremautomata is elsősorban nyelvek felismerésére szolgál, és mint látni fogjuk, többet tud, mint a véges automata.

1. Nemdeterminisztikus veremautomata

A későbbi jelölések egyszerűsítésére, ha Σ egy ábécé, akkor vezessük be a

$$\Sigma_0 = \Sigma \cup \{\varepsilon\}$$

jelölést.

1. Definíció. *A* veremautomata *(vagy* PDA*) egy $M = (Q, \Sigma, \Gamma, q_0, Z_0, F, \delta)$ hetessel írható le, ahol*

- *Q egy véges, nem üres halmaz, az automata* állapota*inak halmaza.*
- *Σ egy véges, nem üres halmaz, az automata (bemeneti)* ábécéj*e.*
- *Γ egy véges, nem üres halmaz, a* verem ábécéj*e vagyis a lehetséges veremszimbólumok halmaza.*
- *$q_0 \in Q$ a* kezdő állapot*.*
- *$Z_0 \in \Gamma$ a verem* kezdő szimbóluma*.*
- *$F \subseteq Q$ az* elfogadó *állapotok halmaza.*
- *δ az* állapotátmeneti függvény*, $\delta(q, a, A) \subseteq Q \times \Gamma^*$, ahol $q \in Q$, $a \in \Sigma_0$ és $A \in \Gamma_0$.*

3. Determinisztikus veremautomata

Természetesen lehet a veremautomatának determinisztikus változatát is definiálni, és az alkalmazások szempontjából ez a hasznosabb változat.

A determinisztikus veremautomatában minden esetben legfeljebb egy lépési lehetőség van. (Ez a véges automatáknál használt terminológia szerint igazából a hiányos automata.) Az ε-átmeneteket ilyenkor is megtartjuk, csak ezek sem kerülhetnek konfliktusba a ténylegesen olvasó (író) lépésekkel, tehát pl. ha van $\delta(q, \varepsilon, A)$ átmenet, akkor ugyanerre a q állapotra és A veremtetőre egyetlen $a \in \Sigma$ karakterre sem lehet a $\delta(q, a, A)$ átmenet is definiálva.

Az első példa veremautomatája egy determinisztikus veremautomata, de az a PDA, amit CF nyelvtanból definiáltunk, nem determinisztikus.

A következő eredmény indokolja, hogy veremautomatának a nemdeterminisztikus változatot hívjuk.

1. Veremautomaták

Azt már láttuk, hogy nemdeterminisztikus véges automaták éppen a reguláris nyelveket ismerik fel. Azt is láttuk, hogy $\{a^n b^n | n \geqslant 0\}$ nem reguláris. Ezt lehet úgy is értelmezni, hogy egy automata "nem képes megjegyezni dolgokat, legfeljebb az állapotai segítségével". Azaz nem képes pl. megszámolni, hogy hány a betű van a szó elején, hogy aztán összehasonlítsa a b-k számával; feltéve, hogy az a-k számára nincsen felső korlát.

Mi lenne, ha az automatánkat ellátnánk azzal a képességgel, hogy számoljon? Mondjuk egy veremmel!

Definíció: Veremautomata alatt egy $(Q, \Sigma, \Gamma, \delta, q_0, F)$ rendezett hatost értünk, ahol

Q az állapotok halmaza,

$q_0 \in Q$ kezdőállapot,

$F \subseteq Q$ végállapotok halmaza,

és Σ az input ábécé, továbbá

Γ a verem ábécé,

$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \to P(Q \times (\Gamma \cup \{\varepsilon\}))$ pedig az átmenetfüggvény, teljesen definiált.

Intuitívan: Az automata input szalagján egy Σ^*-beli szó van, a veremben egy Γ^*-beli. Kezdetben a veremben ε szerepel, az input szalagon meg az input szó. Az átmenetfüggvény fog egy állapotot, beolvas egy Σ-beli betűt az input szalagról és egy Γ-beli betűt a verem tetejéről, majd átmegy valamely állapotba, az input szalagján a mutatót jobbra mozgatja és a verem tetején levő betűt lecseréli valami másra.

Ehhez még hozzájön az, hogy megengedünk üres átmeneteket (azaz megengedjük, hogy ε-t olvasson be), és alapból nemdeterminisztikusnak definiáltuk az automatát.

Az automata által felismert nyelvben pont azon szavak vannak, melyeket az automata feldolgozhat (tehát az input szalagon végigolvashatja a szót) úgy, hogy végállapotba kerül. A működés során a verem tartalmát külön rögzíteni kell.