

Formális nyelvek, nyelvtanok, gépek

Számítógépes nyelvészeti – 2018 tavasz

5. óra

Simon Eszter – Mittelholcz Iván

MTA Nyelvtudományi Intézet

1. A Chomsky-féle nyelvhierarchia
2. Automaták
3. Morfológiai elemzés transzducerekkel
4. Házi feladat

A Chomsky-féle nyelvhierarchia

Definíció (nyelv)

Egy tetszőleges véges A halmazból alkotott A^* halmaz tetszőleges részhalmazát (vagyis az A fölötti füzérekből összegyűjtött tetszőleges halmazt) nyelvnek nevezzük, az A halmazt pedig e nyelv ábécéjének.

A nyelvtan formális meghatározása

Definíció (nyelvtan)

A $G = \langle V_T, V_N, S, R \rangle$ négyest formális nyelvtannak nevezzük, ahol

V_T : a terminális elemek ábécéje

V_N : a nem-terminális elemek ábécéje

S : a nyelvtan kezdőszimbóluma

R : a nyelvtan szabályainak a halmaza

A Chomsky-féle nyelv(tan)hierarchia

Definíció (Chomsky-féle nyelv(tan)hierarchia)

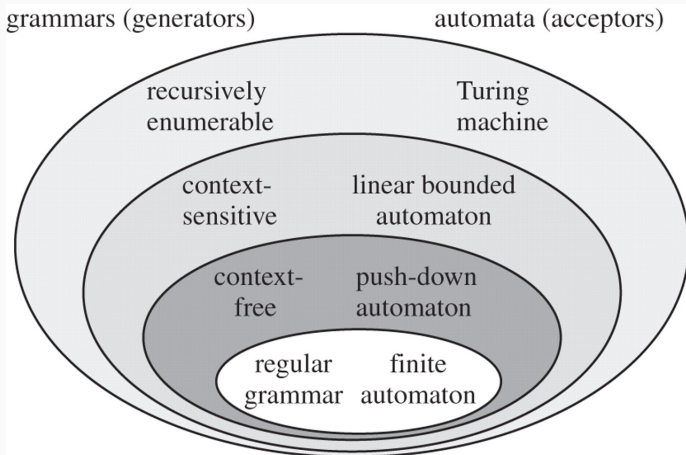
Egy $G = \langle V_T, V_N, S, R \rangle$ nyelvtan i típusú ($i \in \{0, 1, 2, 3\}$) a Chomsky-féle nyelvtanhierarchia szerint, amennyiben az R szabályhalmaz minden elemére teljesül az adott típusban előírt, a szabály felépítésére vonatkozó előírás.

Jelölések: $\alpha, \beta, \gamma \in (V_T \cup V_N)^*$

$A, B \in V_N$

$x \in V_T^*$

- | | |
|--|---|
| 0. típus (megszorítatlan újraíró rendszer) | $\alpha \rightarrow \beta$, ahol $\alpha \neq \varepsilon$ |
| 1. típus (környezetfüggő) | $\alpha A \beta \rightarrow \alpha \gamma \beta$, ahol $\gamma \neq \varepsilon$ |
| 2. típus (környezetfüggetlen) | $A \rightarrow \gamma$ |
| 3. típus (reguláris vagy jobblinéaris) | $A \rightarrow xB$ vagy $A \rightarrow x$ |



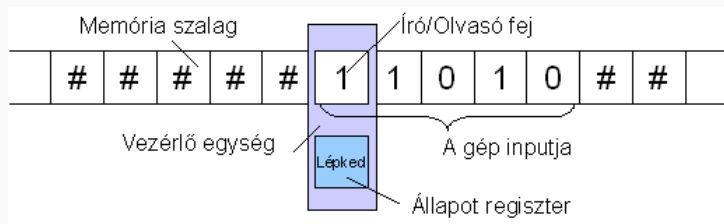
Megszorítatlan (0. típusú) nyelvek

$\alpha \rightarrow \beta$, ahol $\alpha \neq \varepsilon$

- megszorítatlan újraíró rendszer
- a leghasznosabb új eszköz a szimbólumcsere
- a 0. típusú nyelveket Turing-géppel lehet elfogadtatni \rightarrow a legáltalánosabb nyelvtantípusnak a létező és elképzelhető legáltalánosabb absztrakt gép felel meg
- „a Turing-géppel minden kiszámolható, ami egyáltalán kiszámolható, és minden meghatározható, ami egyáltalán meghatározható emberi elménk számára”

Turing-gép

- író-olvasó fej, egy kockákra osztott végtelennek tekintett szalag, amelynek minden kockáján egy szimbólum áll
- a fej jobbra és balra is mozoghat
- az író-olvasó fejnek különböző állapotai vannak, amelyek megszabják, hogy az éppen leolvasott szimbólumot átírja-e, vagy lépjen tovább
- a végtelennek tekintett szalag egy véges részén van csak információ, a többi $\#$ jelet tartalmazó üres hely: $\dots\#a_1a_2\dots a_n\#\dots$
- a számítás kezdetén a fej a bal szélső nem-üres szimbólumon áll, utána: $\langle q_i, a, q_j, X \rangle$
- ha $X \in \{J, B\}$, akkor jobbra vagy balra lép egyet, különben marad és átír
- a Turing-gép akkor fogad el egy füzért, ha a számítás véges számú lépés után leáll





A környezetfüggő (1. típusú) nyelvek

$\alpha A \beta \rightarrow \alpha \gamma \beta$, ahol $\gamma \neq \varepsilon$

- a környezetfüggő szabály egyetlen nem-terminális szimbólum átalakításáról gondoskodik
- a bemeneti oldalon a szimbólum kétoldali környezetére is hivatkozhatunk
- a nem-terminális szimbólum megsemmisítése nem megengedett \rightarrow hosszúságot nem csökkentő szabályok
- a környezetfüggő nyelvek absztrakt számítógépes jellemzésére a lineárisan korlátozott Turing-gépek szolgálnak \rightarrow az író-olvasó fej nem léphet ki egy kezdő- és egy végszimbólumokkal jelölt területen kívülre

Noam Chomsky and Morris Halle: The Sound Pattern of English (1968)

$$A \rightarrow B/[precontext_postcontext]$$

tundrai nyenyec cirill–IPA konverzió

ю → *ju*/#__

ю → *ju*/__ [aeёioуыэюя]

ю → ^j*u*/ [бвгджзйклмнңпрстфхцчшщ] __

else:

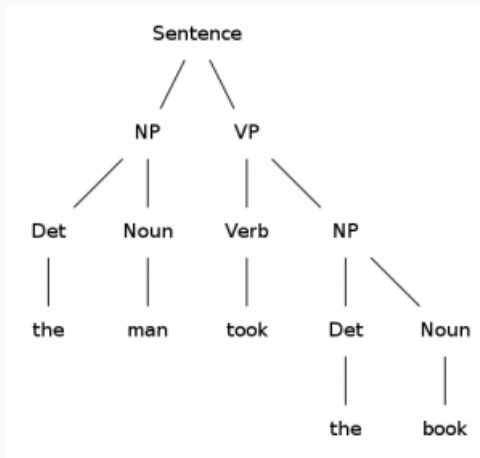
ю → *ju*/__

A környezetfüggetlen (2. típusú) nyelvek

$A \rightarrow \gamma$

- csak a szabály bemeneti oldala van korlátozva: egyetlen nem-terminális szimbólum állhat ott
- a kimeneti oldalon megengedett az üres sztring is
- Chomsky-féle normálalak: $A \rightarrow a$ vagy $A \rightarrow BC$
- elfogadó gép: veremautomata \rightarrow mintha egy véges automata meg lenne toldva egy veremszerű memóriával
- verem: „last in, first out” (LIFO)

Környezetfüggetlen szabályok



Az első környezetfüggetlen elemzési fa (Chomsky, 1956: Three models for the description of language. *IRI Transactions on Information Theory*, 2(3), 113–124.).

A reguláris (3. típusú) nyelvek

$A \rightarrow xB$ vagy $A \rightarrow x$

- a reguláris nyelveket leíró reguláris nyelvtanok reguláris kifejezésekkel ekvivalensek
- a reguláris nyelvtanok lehetnek jobblinéárisak vagy balinéárisak
- egy jobblinéáris szabály bemeneti oldalán egyetlen nem-terminális szimbólum állhat, és maximum egy nem-terminális állhat a kimeneti oldalán, és ez utóbbinak a legutolsónak kell lennie
- elfogadó gép: véges állapotú automata (finite state automaton, FSA)

Hol vannak a természetes nyelvek?

- valahol a környezetfüggetlen és a környezetfüggő között → enyhén környezetfüggő nyelvtanok
- környezetfüggetlen szabályokkal a természetes nyelvi jelenségek nagyon nagy része leírható, de nem minden → ellenpélda egy svájci német dialektusból

Shieber (1985): keresztező függőség az igék és tárgyak között

Jan säit das...

...mer **em** Hans **es** huus **hälfed** **aastriche**

mi **Hans.DAT** **a ház.ACC** **segített** **fest**

‘Jan azt mondta, hogy segítettünk Hansnak festeni a házat.’

két ugyanolyan füzér konkatenációjával előálló ismétléses füzérek
környezetfüggetlen nyelve: $\{xx \mid x \in \{a, b\}^*\}$

Automaták

Definíció (véges állapotú automata)

Az $M = \langle K, \Sigma, d, q_0, F \rangle$ ötös egy véges állapotú automata, ahol

K : az automata állapotainak véges halmaza

Σ : az ábécé

q_0 : a kezdőállapot

F : a végállapotok halmaza

d : az átmenetek halmaza

Determinisztikus bégetőautomata

$$K = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$

$$\Sigma = \{b, e, !\}$$

$$F = \{q_4\}$$

$$\begin{aligned} d = \{ & \langle q_0, b, q_1 \rangle, \langle q_1, e, q_2 \rangle, \langle q_2, e, q_3 \rangle, \langle q_3, e, q_3 \rangle, \\ & \langle q_3, !, q_4 \rangle, \langle q_0, e, q_5 \rangle, \langle q_0, !, q_5 \rangle, \langle q_1, b, q_5 \rangle, \langle q_1, !, q_5 \rangle, \\ & \langle q_2, b, q_5 \rangle, \langle q_2, !, q_5 \rangle, \langle q_3, b, q_5 \rangle, \langle q_4, b, q_5 \rangle, \langle q_4, e, q_5 \rangle, \\ & \langle q_4, !, q_5 \rangle, \langle q_5, b, q_5 \rangle, \langle q_5, e, q_5 \rangle, \langle q_5, !, q_5 \rangle \} \end{aligned}$$

Egyszerűsített determinisztikus bégetőautomata

$$K = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma = \{b, e, !\}$$

$$F = \{q_4\}$$

$$d = \{ \langle q_0, b, q_1 \rangle, \langle q_1, e, q_2 \rangle, \langle q_2, e, q_3 \rangle, \langle q_3, e, q_3 \rangle, \\ \langle q_3, !, q_4 \rangle \}$$

determinisztikus

nincs választási lehetőség, az algoritmus minden inputra egyértelműen tudja, hogy mit kell csinálni

nem-determinisztikus

1. a gép nem minden inputra tudja egyértelműen, hogy mit csináljon
2. egy ϵ -átmenet nem-determinisztikussá teszi az automatát, mert nem tudja, hogy merre menjen tovább, és ebben az input sem segít

a nem-determinisztikus automata mehet rossz irányba →
grammatikus sztringet utasít el → sztenderd megoldások:

- **backup:** megjelöljük a döntési pontot, így ha kiderül, hogy rosszfelé mentünk, akkor vissza tudunk oda térni, és mehetünk a másik ágon
- **look-ahead:** okosan előre nézünk az inputban, hogy el tudjuk dönteni, hogy merre érdemes menni
- **párhuzamosítás:** a döntési pontoknál minden alternatív utat párhuzamosan bejárunk

Nem-determinisztikus bégetőautomaták

$$K = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma = \{b, e, !\}$$

$$F = \{q_4\}$$

$$d = \{ \langle q_0, b, q_1 \rangle, \langle q_1, e, q_2 \rangle, \langle q_2, e, q_2 \rangle, \langle q_2, e, q_3 \rangle, \\ \langle q_3, !, q_4 \rangle \}$$

VAGY

$$K = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma = \{b, e, !, \varepsilon\}$$

$$F = \{q_4\}$$

$$d = \{ \langle q_0, b, q_1 \rangle, \langle q_1, e, q_2 \rangle, \langle q_2, e, q_3 \rangle, \langle q_3, \varepsilon, q_2 \rangle, \\ \langle q_3, !, q_4 \rangle \}$$

determinisztikus

	bemenet		
állapot	b	e	!
0	1	-	-
1	-	2	-
2	-	3	-
3	-	3	4
4:	-	-	-

Állapot-átmenet táblák II.

nem-determinisztikus 1.

	bemenet		
állapot	b	e	!
0	1	-	-
1	-	2	-
2	-	2,3	-
3	-	-	4
4:	-	-	-

nem-determinisztikus 2.

	bemenet			
állapot	b	e	!	ϵ
0	1	-	-	-
1	-	2	-	-
2	-	3	-	-
3	-	-	4	2
4:	-	-	-	-

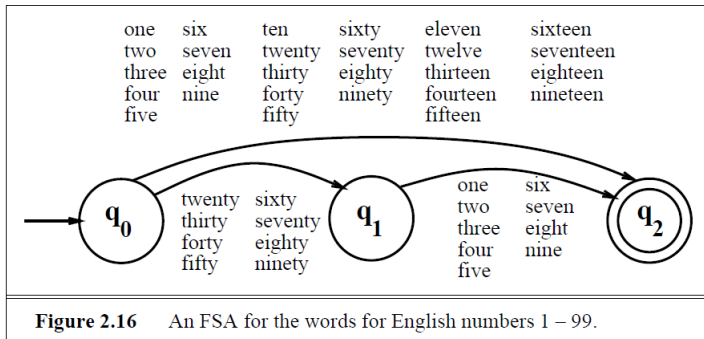


Figure 2.16 An FSA for the words for English numbers 1 – 99.

Morfológiai elemzés transzducerekkel

Definíció (véges állapotú transzducer)

Az $M = \langle K, \Sigma, d, q_0, F \rangle$ ötös egy véges állapotú transzducer, ahol

K : a transzducer állapotainak véges halmaza

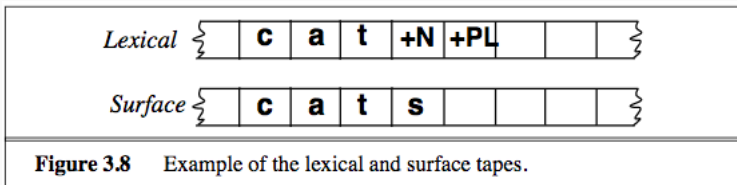
Σ : az ábécé (komplex szimbólumok halmaza, ahol egy komplex szimbólum egy input-output pár $i:o$)

q_0 : a kezdőállapot

F : a végállapotok halmaza

d : az átmenetek halmaza, pl. $\langle q_0, i : o, q_1 \rangle$

- felismerés → morfológiai elemzés
- Koskenniemi (1983): megfeleltetés egy szó felszíni alakja és elemzése között



Automatából transzducer 1.

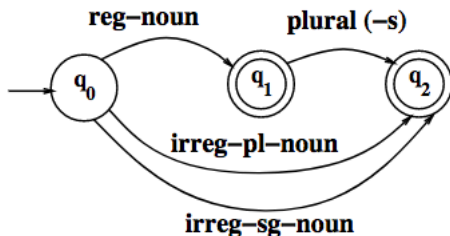


Figure 3.2 A finite-state automaton for English nominal inflection.

reg-noun	irreg-pl-noun	irreg-sg-noun	plural
fox cat dog aardvark	geese sheep mice	goose sheep mouse	-s

Automatából transzducer 2.

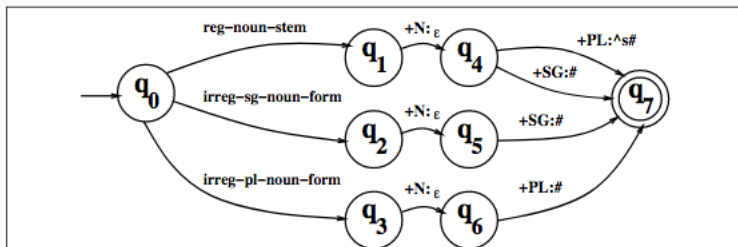
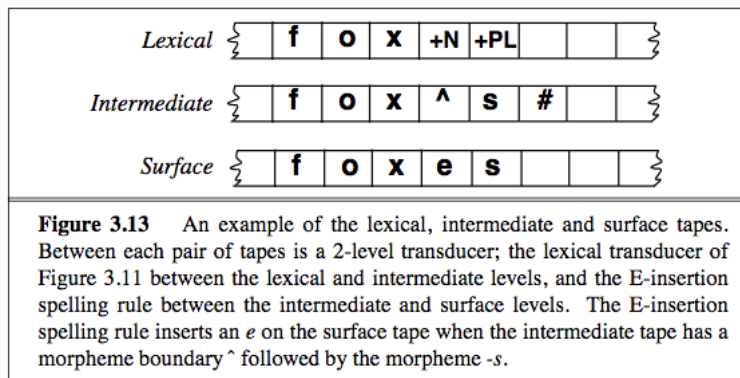


Figure 3.9 A transducer for English nominal number inflection T_{num} . Since both q_1 and q_2 are accepting states, regular nouns can have the plural suffix or not. The morpheme-boundary symbol \wedge and word-boundary marker $\#$ will be discussed below.

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:ε o:ε s ε	goose
cat	sheep	sheep
dog	m o:i u:ε s:c ε	mouse
aardvark		



Agglutinálunk (?)

<i>morfémák:</i>	igekötő	tő	képző	képző	eset	
<i>morfok:</i>	meg	emlék	ez	és	ül	
<i>morfémák:</i>	tő	szám	eset			
<i>morfok:</i>	barát	ok	ért			
<i>morfémák:</i>	igekötő	tő	LEH	képző	képző	eset
<i>morfok:</i>	fel	ismer	het	etlen	ség	ig
<i>morfémák:</i>	tő	Sing	1			
<i>morfok:</i>	könyv		em			
<i>morfémák:</i>	tő	Pl	2			
<i>morfok:</i>	tanul		unk			

Kiefer: A ragozás. In: STRMNy 3.

emMorph-kimenetek:

1. terminálban futtatva magában:

kedvel[V]jük[P1.Def]
róka[N]k[Pl]at[Acc]

2. e-magyar:

kedvel[/V] + jük[Prs.Def.1Pl]
róka[/N]=róká + k[Pl] + at[Acc]

- Helsinki Finite-State Technology (HFST): <https://hfst.github.io/>
- Xerox Finite State Toolkit (XFST)
- Foma: <https://fomafst.github.io/>
- Stuttgart Finite State Toolkit (SFST):
<http://www.cis.uni-muenchen.de/~schmid/tools/SFST/>
- OpenFST:
<http://www.openfst.org/twiki/bin/view/FST/WebHome>

Házi feladat

1. Csinálj egy olyan determinisztikus véges állapotú automatát, amely elfogadja az alábbi magyar szavakat és azok többszámú alakját: *pók*, *póni*, *pék*, *póré*, *szék*. Prezentáld mindhárom tanult módon (állapotdiagram, formális leírás, állapot-átmenet tábla)!

2a. Készítsd el a determinisztikus bégetőautomatának a negáltját, és prezentáld mindhárom tanult módon (állapotdiagram, formális leírás, állapot-átmenet tábla)!

Egy FSA negáltja azokat a sztringeket fogadja el, amiket az FSA elutasít, és azokat utasítja el, amiket az FSA elfogad (ugyanafölött az ábécé fölött).

2b. Programozás

- Alberti Gábor: *Matematika a természetes nyelvek leírásában*. Segédkönyvek a nyelvészet tanulmányozásához 52. Tinta Könyvkiadó, Budapest, 2006.
- Daniel Jurafsky & James H. Martin: *Speech and Language Processing*. 1st edition. Chapter 2, 3 & 13. Prentice Hall, 2000.