

A logika és a számítástudomány alapjai mérnök informatikus hallgatóknak

11. előadás, véges automaták

Mihálydeák Tamás, Aszalós László

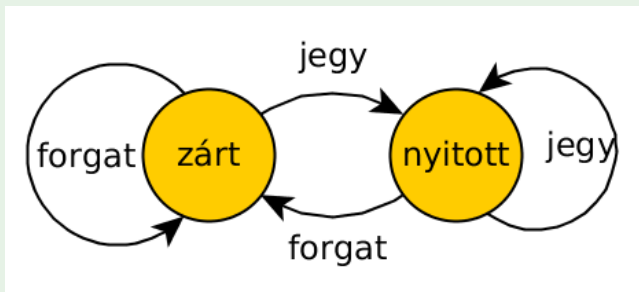
A tananyag elkészítését az EFOP-3.4.3-16-2016-00021 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

2019. november 4.

- 1 Motiváció
- 2 Véges automata
- 3 Nemdeterminisztikus automata

Beengedő-kapu

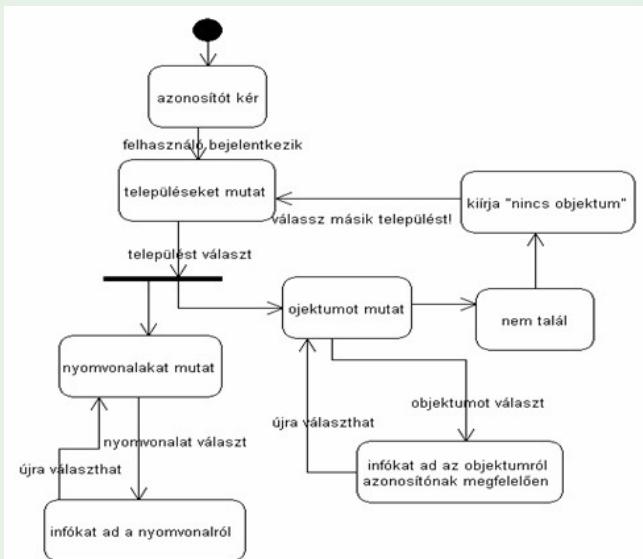
Példa - automata



Ha valaki a jegyét beolvastatja a kapuval, az kinyílik, és ekkor ő átmehet rajta (forgat); ezzel a kapu zárul. A nyitott kapuval hiába olvastatjuk a jegyet, és a zárt kaput is hiába próbáljuk forgatni, a kapu állapota nem változik.

Állapotdiagram (UML)

Példa - szoftvertervezés



Véges automata

Az előbbi példáknak van egy matematikai modellje, a véges automata. Mivel a továbbiakban csak véges automatákkal foglalkozunk, a *véges* jelzőt elhagyjuk.

Definíció

A **véges automata** egy rendezett ötös: $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, ahol

- Q (belső) állapotok véges halmaza,
- Σ (külső) véges ábécé,
- $\delta : Q \times \Sigma \rightarrow Q$ átmenetfüggvény,
- $q_0 \in Q$ kezdőállapot és
- $F \subseteq Q$ az elfogadóállapotok halmaza.

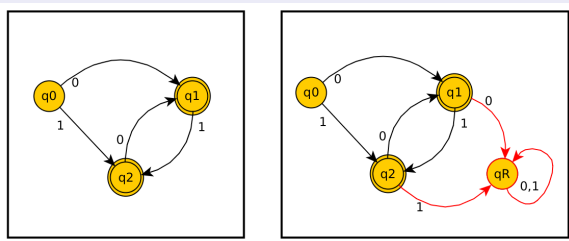
Megjegyzés.

Grafikus ábrázolásban a belső állapotokat körökkel, az átmenetfüggvényt nyilakkal és a rájuk írt külső ábécé betűivel jelöljük. A kezdőállapot egy bemenő, felirat nélküli nyíl jelzi, míg a végállapotokat duplán rajzolt körök.

Hiányos automaták

Megjegyzés

Definíció szerint egy állapotból az ábécé minden betűjéhez kell rendelni egy állapotot (δ teljes függvény). A rajz egyszerűbbé tétele érdekében a hiányzó nyilak egy nem jelölt *rontott* állapotba mutatnak, ahonnan minden inputra ugyanide jutunk vissza. Az ábrán a kiegészítés pirossal lett kiemelve.



Elfogadó automata

Definíció

$A = \langle Q, \Sigma, \delta, q_0, F \rangle$ automata és $w = a_1 a_2 \dots a_n \in \Sigma^*$ szó esetén azt mondjuk, hogy az automata **elfogadja** (felismeri) a w szót, ha

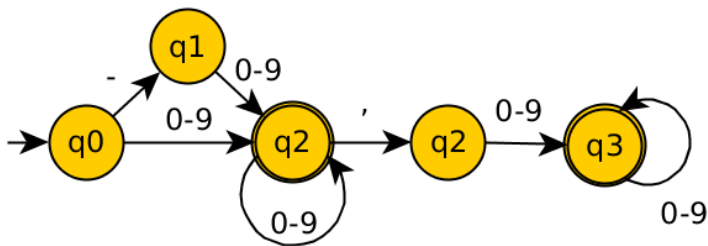
$$\delta(\dots (\delta(\delta(q_0, a_1), a_2) \dots), a_n) \in F.$$

Definíció

Legyen $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ egy automata! Ha

$L = \{w \in \Sigma^* \mid A \text{ elfogadja } w\text{-t}\}$, akkor azt mondjuk, hogy az A **elfogadja** (felismeri) az L nyelvet. Jelölése $L = L_A$

Példa



Az előbbi automata elfogadja a 5; -92; 754; -15, 7 és 79, 642 karaktersorozatokat.

Viszont nem fogadja el a --4; 2-7; -, 5 és 12, karaktersorozatokat. A felismert nyelv a helyesen formált egészek és tizedestörtek nyelve.

Determinisztikus automata Python-ban

```
dfa = {0: {'a': 0, 'b': 1},  
       1: {'a': 2, 'b': 0},  
       2: {'a': 1, 'b': 2}}
```

```
def accepts(transitions, initial, accepting, s):  
    state = initial  
    for c in s:  
        state = transitions[state][c]  
    return state in accepting
```

```
>>> accepts(dfa, 0, {0}, 'babbbab')
```

```
True
```

```
>>> accepts(dfa, 0, {0}, 'babbbabb')
```

```
False
```

Megjegyzés

- Az eddig definiált keretek között pár feladatot igen nehéz megoldani, például *azt a nyelvet, mely egy automata által felismert nyelv szavainak megfordításából áll, automatával fel lehet ismerni?* Ezt a kérdést és továbbiakat igen könnyen megválaszolhatunk, ha általánosabb automata-fogalmat használunk.
- Eddig minden egyes lépés és eredménye egyértelműen meghatározható volt. A nem determinisztikus végrehajtás több ponton is lehetővé tesz választást a soron következő, lehetséges lépések közül. Az általánosabb automatát a megkülönböztetés érdekében *nemdeterminisztikus automatának* nevezzük.
- A hagyományos automatától csak az átmenetfüggvény tekintetében különbözik ez a fajta automata, akár több irányba is haladhatunk, illetve input beolvasása nélküli állapotátmenetre is lehetőség van.

Nemdeterminisztikus automata

Definíció

A **nemdeterminisztikus véges automata** egy rendezett ötös:

$A = \langle Q, \Sigma, \delta, q_0, F \rangle$, ahol

- Q (belső) állapotok véges halmaza,
- Σ (külső) véges ábécé,
- $\delta : Q \times \Sigma_\varepsilon \rightarrow 2^Q$ átmenetfüggvény,
- $q_0 \in Q$ kezdőállapot és
- $F \subseteq Q$ az elfogadóállapotok halmaza.

Megjegyzés

Az előbbieken

- 2^Q a Q állapothalmaz részhalmazainak halmazát (hatványhalmazát) jelöli.
- $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$, ugyanis lehetséges állapotátmenet input nélkül is.

Nemdeterminisztikus automata által felismert szavak

Definíció

Az $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ nemdeterminisztikus automata elfogadja (felismeri) a $w \in \Sigma^*$ szót ($w = a_1 a_2 \dots a_n$) ha létezik olyan $m \geq n$ szám, r_0, r_1, \dots, r_m állapotok sorozata ($r_i \in Q$), és b_0, b_1, \dots, b_m $\Sigma \cup \{\varepsilon\}$ -beli „betűk” sorozata hogy

- 1 $r_0 = q_0$,
- 2 $r_{i+1} \in \delta(r_i, y_{i+1})$ $0 \leq i < m$, és
- 3 $r_m \in F$

Megjegyzés

A nemdeterminisztikus véges automata által elfogadott nyelv az előzőhöz hasonlóan definiálható.

Nemdeterminisztikus automata működése

Megjegyzés

- Míg a determinisztikus automata futása egy ágon halad, és az input beolvasásának végén egyértelműen kiderül, hogy elfogadja, vagy elutasítja az inputot, a nemdeterminisztikus automatának igen sok különböző ág is tartozhat a futásához. Pontosan akkor ismeri fel a szót, ha ezek közül legalább az egyik ág sikeres. Ha programmal szeretnénk szimulálni az automata futását, akkor egyrészt használhatunk *backtrack* algoritmust, mely egymás után bejárja az összes ágot, vagy gondolkozhatunk párhuzamosan.
- Utóbbi esetben nyilván kell tartani, hogy mely állapotokba juthatott el az input szó hatására az automata. Elég hosszú szó esetén már az összes állapot szóba jöhet, így ha $|Q| = k$, 2^k lehetséges végkifejlet lehet. Ezt szisztematikusan követve a nemdeterminisztikus automatához konstruálhatunk egy determinisztikus automatát is.
- A konstrukcióban egyedül az automata definícióját kell használni.

Determinisztikus és nemdeterminisztikus automaták

Az előbbiekben felvázoltuk a bizonyítását:

Tétel

Minden A nemdeterminisztikus véges automatához megkonstruálható egy olyan A' determinisztikus véges automata, hogy $L_A = L_{A'}$

Nemdeterminisztikus automata Pythonban (haladóknak)

Forrás: <https://swizec.com/blog/strangest-line-of-python-you-have-ever-seen/swizec/3012>

```
from optparse import OptionParser
from collections import defaultdict
def run(inputWord):
    a, states = open("a.txt"), defaultdict(list)
    state, final = a.readline().split()[1:], a.readline().split()[1:]
    [states[(i.split()[0], i.split()[1])].append(i.split()[3]) for i in a]
    for letter in inputWord:
        state = [st for s in state for st in states[(s, letter)]]
    return any(i in state for i in final)

print((b, "YES") if run(b) else (b, "NO" ) for b in OptionParser().parse_args()[1])

#a.txt:
#init: q0
#final: q2 q3
#q0 a -> q1
#q0 b -> q3
#q1 a -> q1
#q1 b -> q2
#q2 a -> q2
#q2 b -> q2
```