## Ismétlés

Az elmélet legérdekesebb kérdése: Hogyan lehet egy nyelv esetén eldönteni, hogy egy adott szó helyes-e?

Parciálisan eldöntő algoritmus, jobb nincs is (általában).

Totálisan eldöntő algoritmus (de a futásidő a legrosszabb esetben exponenciális, ami gyakorlatilag használhatatlan.)

Van totálisan eldöntő algoritmus, CYK, futásidő polinomiális, ez már használható, de azért ez is nagy.

Most jön.

## i=3

Adott , eldöntendő:

### Emlékeztető nf 3

Ötlet:

(="az szabály")

Visszavezettük erre: Ennek futásideje

fokozatos közelítéssel:

1 db H előállítsa:

db van.

Összesen:

A szó hosszával arányos futásidejű algoritmus.

Ezzel a szóprobléma vizsgálatának vége. Jönnek rá az algoritmusok

# Célhardverek a szóprobléma eldöntésére

## Véges determinisztikus automata (legegyszerűbb)

### Informális leírása

Számítógépmodell

Input szalag, minden cellában 1-1 betű

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Központi egység (CPU), véges sok állapota van.

Működése diszkrét időskálában, ütemenként.

1 ütem két félütemből áll: kigyűjtés és akció

Kigyűjtés: Az aktuális input és állapot meghatározása.

Akció: A kigyűjtés függvényében új állapot. Az inputfej (olvasófej) egy cellával jobbra lép.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

A következő ütem az előző konfigurációját folytatja.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

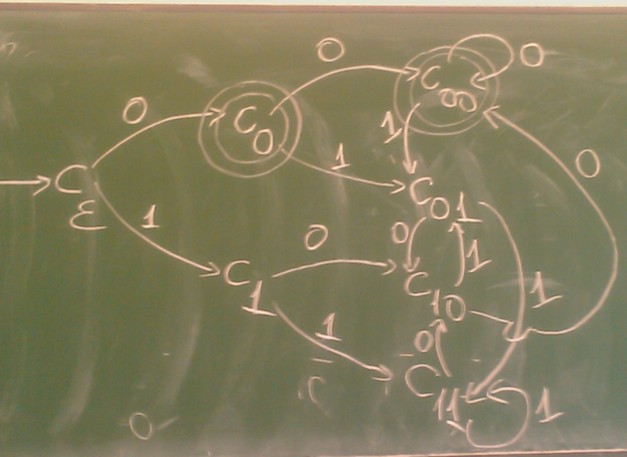
Felismerés: A szó teljes végigolvasása után speciális állapotba, úgynevezett végállapotba kerül.

### Példa

(Bitsorozatok, melyek oszthatók 4-gyel.)

Invariánsok:

Automata:



Elfogadó állapotok (koncentrikus körrel jelöljük):

### Konfiguráció

q: jelenlegi állapot

v: mi van még hátra

## Véges determinisztikus automata formális definíciója

Véges determinisztikus automatán (VDA-n) a következő ötöst értjük:

az állapotok véges halmaza input ábécé átmenetfüggvény (átmeneti függvény) kezdőállapot, ahonnan elindulunk végállapotok halmaza

### Átmeneti függvény ()

Mindenhol értelmezett függvény, állapothoz és bemenő jelhez rendel új állapotot.

### megadásának módszerei

Táblázattal:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Képlettel:  
Később

Átmeneti gráffal:

→rajz fent

Irányított gráf, pontjai Q elemeivel, élei T elemeivel címkézettek.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Képlettel:

aaaaaaaaa

: 2 hosszúságú suffixa

Gráffal:

→rajz fent

## VDA által felismert, elfogadott nyelv

### Először terjesszük ki -t tágabb értelmezési tartományra

a q-ból a v hozzáadása létrejövő állapot

(-re már definiálva van, új betű)

### Valódi kiterjesztés

A kalapot nem használjuk. helyett -t használunk. (mert valódi kiterjesztés.

### Most jöjjön a definíció

### VDA-ban felismerhető nyelvek összessége

## TÉTEL:

Pontosan 3. típusú nyelvek esetében működik.

### Emlékeztető 3. típusú nyelvtan

…

### BIZ: kétirányú tartalmazás

### ⊆ irány

Elegendő bizonyítani, hogy tetszőleges VDA-hoz nyelvtan, hogy

Első ötlet: generálja, amit olvasna

Második ötlet: nyelvtani jel = állapota (?)

-ra vonatkozó teljes indukcióval bizonyítható, hogy:

Innen 2. miatt

### Példa

### ⊇ irány

Megjegyzés: A készült nyelvtan nem csak 3. típusú, hanem nf 3 is. / Nem tetszőleges 3. típusból fogunk elindulni, hanem nf 3-ból.

Elegendő bizonyítani, hogy tetszőleges nyelvtanhoz , melyre

Ötlet: Az előző konstrukció megfordítása.

Az végállapotokat a 2. pont megfordításával, -t pedig az 1. pont megfordításával kapjuk.

Probléma: nem egyértelműsíthető. Nem determinisztikus az átmenet.

Példa erre:

Ennek átmenetfüggvénye:



Látjuk, hogy ez nem definiált szerkezet. Véges nemdeterminisztikus automata.

(Néhány helyen helyett -t kellett volna írnom.)