  
**Eötvös Lóránd Tudományegyetem**  
Informatikai kar  
Algoritmusok és Alkalmazásaik Tanszék

Turing gép vizualizálása webes környezetben

Nagy SáraKelemen Márton

mestertanárprogramtervező informatikus Bsc  
programtervező matematikus

Budapest 2020

Tartalomjegyzék

[Bevezetés 2](#_Toc59625279)

[Tudományos kitekintés 3](#_Toc59625280)

[Alapfogalmak 3](#_Toc59625281)

[Felhasználói dokumentáció 6](#_Toc59625282)

[A program által megoldott feladat 6](#_Toc59625283)

[Célközönség 6](#_Toc59625284)

[Rendszerkövetelmények 6](#_Toc59625285)

[Minimális hardveres környezet 6](#_Toc59625286)

[Optimális hardveres környezet 7](#_Toc59625287)

[Szoftveres követelmények 7](#_Toc59625288)

[Üzembe helyezés 7](#_Toc59625289)

[Funkciók használata 7](#_Toc59625290)

[Fejlesztői doumentáció 8](#_Toc59625291)

[A rendszer architektúrája 8](#_Toc59625292)

[Alkalmazott technológiák 8](#_Toc59625293)

[Alkalmazott módszerek 8](#_Toc59625294)

[Adatbázis terve 8](#_Toc59625295)

[Modul és osztályszerkezet 8](#_Toc59625296)

[A Kliens-oldal osztályszerkezet 8](#_Toc59625297)

[A felhasználó felület terve 8](#_Toc59625298)

[Telepítés fejlesztői célokra 8](#_Toc59625299)

[Megvalósítás 8](#_Toc59625300)

[Tesztelés 9](#_Toc59625301)

[Irodalomjegyzék 10](#_Toc59625302)

# Bevezetés

A szakdolgozat célja, hogy mások számára vizuális eszközökkel könnyebben megérthető legyen a Turin gép működése. A szakdolgozatnak két fő részre van, az algoritmus implementálása egy programozási nyelvre és a szemléltetés kivitelezése. A modell egyetlen programot hajt végre bármilyen inputra (ami szalagon érkezik), azaz tekinthető egy célszámítógépnek. A gép főbb részei: a vezérlőegység (CPU), a szalag, amely az inputot hivatott megvalósítani és egy író-olvasó fej, amely a szalagot még léptetni is tudja. A megvalósítandó gép determinisztikus, továbbá minden esetben definiált az átmenet. A végtelen szalag potenciálisan végtelen tár.

A lehetséges Turing gépek egy előre definiált fájlformátumban adhatók meg. Ezek közül választva futtathatunk többféle gépet is. Egy-egy futtatásánál választható több mód is: lehetséges az eredmény azonnali megtekintése, illetve a futás végig nézhető lassítva és lépésenként, kézileg léptetve is.

# Tudományos kitekintés

## Alapfogalmak

Forrás: https://web.cs.elte.hu/~tichlerk/logika/h/tg.pdf

A Turing-gép egy olyan M = {Q, Σ, Γ, δ, q0, qi, qn} rendszer, ahol

* Q az állapotok véges, nemüres halmaza,
* q0, qi, qn ∈ Q, q0 a kezdő- qi az elfogadó- és qn az elutasító állapot,
* Σ és Γ ´ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy Σ ⊆ Γ és ⊔ ∈ Γ \ Σ.
* δ : (Q \ {qi, qn}) × Γ -> Q × Γ × {L, R, S} az átmenet függvény.

• A Turing-gép működésének fázisait a gép konfigurációival írjuk le. A Turing-gép konfigurációja egy uqv szó, ahol q ∈ Q és u, v ∈ Γ∗, v != *ε*.  
A konfiguráció a gép azon állapotát tükrözi amikor a szalag tartalma uv (uv előtt és után a szalagon már csak ⊔ van), a gép a q állapotban van, és a gép író-olvasó feje a v szó első betűjén áll.  
• A gép kezdőkonfigurációja egy olyan q0u szó, ahol u csak Σ-beli betűket tartalmaz.  
  
• Egy Turing-gép konfigurációátmenetét az alábbiak szerint definiáljuk. Legyen uqav egy konfiguráció, ahol a ∈ Γ, u, v ∈ Γ∗.

* Ha δ(q; a) = (r; b; R), akkor uqav |- ubrv’ , ahol v’ = v , ha v != *ε*, különben v’ = ⊔,
* ha δ(q; a) = (r; b; S), akkor uqav |- urbv,
* ha δ(q; a) = (r; b; L), akkor uqav |- u’rcbv, ahol c ∈ Γ és u’c = u, ha u != *ε*, különben u’ =u *és* c = ⊔ .

• Azt mondjuk, hogy M véges sok lépésben eljut a C konfigurációból a C’ konfigurációba (jele C |-\* C’), ha van olyan n ≥ 1 és C1,….., Cn konfigurációsorozat, hogy C1 = C, Cn = C’ és minden 1 ≤ i < n-re, Ci |- Ci+1.

• Ha q 2 {qi, qn}, akkor azt mondjuk, hogy az uqv konfiguráció egy megállási konfiguráció. q = qi esetében elfogadó, míg q = qn esetében elutasító konfigurációról beszélünk.

• Az M által felismert nyelv (amit L(M)-mel jelölünk) azoknak az u ∈ Σ∗ szavaknak a halmaza, melyekre igaz, hogy q0u |-∗ xqiy valamely x, y ∈ Γ∗, y != *ε*  szavakra.

• Egy L ⊆ Σ∗ nyelv Turing-felismerhető, ha L = L(M) valamely M Turing-gépre. Továbbá, egy L ⊆ *Σ*∗ nyelv eldönthető, ha létezik olyan M Turing-gép, mely minden bemeneten megállási konfigurációba jut és felismeri az L-et. A Turing-felismerhető nyelveket szokás rekurzívan felsorolhatónak, az eldönthető nyelveket pedig rekurzívnak is nevezni. A rekurzívan felsorolható nyelvek osztályát RE -vel, a rekurzív nyelvek osztályát pedig R-rel jelöljük.  
• Tekintsünk egy M = [Q, Σ, Γ, δ, q0, qi, qn] Turing-gépet és annak egy u ∈ Σ∗ bemenő szavát. Azt mondjuk, hogy M futási ideje (időigénye) az u szón n (n ≥ 0), ha M a q0u kezdőkonfigurációból n lépésben (konfigurációátmenettel) jut el megállási konfigurációba. Ha nincs ilyen szám, akkor M futási ideje az u-n végtelen.

• Legyen f : N -> N egy függvény. Azt mondjuk, hogy M időigénye f(n) (vagy, hogy M egy f(n) időkorlátos gép), ha minden u ∈ Σ∗ input szóra, M időigénye az u szón legfeljebb f(|u|).

• A **k**-szalagos Turing-gép egy olyan M = [Q, Σ, Γ, δ, q0, qi, qn] rendszer, ahol

* Q az állapotok véges, nemüres halmaza,
* q0; qi; qn ∈ Q, q0 a kezdő- qi az elfogadó- és qn az elutasító állapot,
* Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy Σ ⊆ Γ és ⊔ ∈ Γ \ Σ,
* δ : (Q \ {qi, qn}) × Γk -> Q × Γk × {L, R, S}k az átmenet függvény.

• A **k szalagos Turing-gép** konfigurációja egy szó, ahol q ∈ Q és ui, vi ∈ Γ∗, vi != *ε* (*1* ≤ i ≤ k). Az u szóhoz tartozó kezdőkonfiguráció: ui = *ε* (1 ≤ i ≤ k), v1 = u, és vi = ⊔ (2 ≤ i ≤ k). Időigény: mint az egyszalagosnál (konfigurációátmenetek száma alapján).

• Szófüggvényt kiszámító Turing-gép:

Az M (determinisztikus) Turing-gép kiszámítja az f : Σ∗ -> Γ∗ szófüggvényt, ha M minden u ∈ Σ∗-ra olyan vqw megállási konfigurációba jut (q ∈ {qi, qn}), ahol vw = f(u) (szóeleji és szóvégi ⊔-ektől eltekintve). Időigény: mint fent (konfigurációátmenetek száma alapján)

# Felhasználói dokumentáció

## A program által megoldott feladat

Az sok tananyag tanulása során néha kevés idő jut, például egy  
összetettebb algoritmus megértésére. Ez az oktatói program ebben próbál segítséget nyújtani. A gyakorlaton alapul vett gráfos ábrázolással, illetve szalagbemenettel fogja majd illusztrálni a program az algoritmus működését. Ezzel hatékonyabbá válhat a tanulás, egyebeket mellett, például segíthet leellenőrizni az elkészített házifeladat helyességét vagy az órán a tanárnak nem muszáj feltétlen minden alkalommal rajzolni a táblára helyette csak kivetíti ezt a webes alkalmazást.

## Célközönség

A program elsősorban informatikai karon, azon belül formális nyelvek, algoritmusok,  
számításelméletet oktató vagy tanuló egyetemi tanárok és hallgatóik számára készült. Ezt  
mutatja a program weboldal formátuma is: minmális hardverkövetelménye okán nincs szükség komolyabb eszközre, így szinte bármilyen intézményben könnyedén alkalmazni lehet. A program segítséget nyújt a hallgatóknak az egyes előre betápálált példa feladatok  
kipróbálásában, ezek újra szerkesztésében. Továbbá segít az oktatóknak is pédául a zárthelyi feladatok kitalálásában vagy diákok által megírtak kijavításában.

## Rendszerkövetelmények

### Minimális hardveres környezet

Legegyszerűbben személyi számítógépen érdemes használni a programot. Például a piacon amit kapni egyik legolcsóbb ilyen eszköz a Rasperry Pi 4 modell B miniszámítógépen.

* 1,5 GHz (1 500 MHz)
* 4 GB mermória

Már ez is képes futtatni egy web böngészőt és egy webszervert. Tehát minimális hardveres követelményei vannak a programnak.

### Optimális hardveres környezet

Szerver számítógép. Minta a http://webprogramozas.inf.elte.hu/. Egy ilyen szerver képes egyszerre több felhasználót kiszolgálni, ha mondjuk egy tanteremben vagy előadáson a tanár kérésére minden diáknak ki kell próbálni a programot.

### Szoftveres követelmények

#### A szerver-oldali futtatáshoz

Személy számítógépen acpache webszervert érdemes telepíteni. Ilyet legegyszerűbben a XAMPP nevű szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomagot érdemes installálni. Egyszerű és bármely desktop operációs rendszerre jó. Továbbá, szerver számítógépre más eljárással érdemes telepíteni webszervert, itt érdemes a szerver üzemeltetők segítségét kérni.

Kliens-oldali futtatáshoz

ES6 és CSS3 támogatású webböngészők:

* Desktopon: legalább az alábbi kiadásokon: Firefox 24, Chrome 37, Safari 9, Edge 12, Opera 24
* Mobil, tablet eszközökön: IOS safari 10, Android Browser 81

## Üzembe helyezés

Fontos tudni -ahogy fentebb írtam- szükséges telepíteni webszerver alkalmazást ha saját gépünk fogja hostolni, ha meg szervert

## Funkciók használata

# Fejlesztői doumentáció

## A rendszer architektúrája

## Alkalmazott technológiák

## Alkalmazott módszerek

## Adatbázis terve

## Modul és osztályszerkezet

## A Kliens-oldal osztályszerkezet

## A felhasználó felület terve

## Telepítés fejlesztői célokra

## Megvalósítás

# Tesztelés

# Irodalomjegyzék