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

Turing gép vizualizálása webes környezetben

Nagy SáraKelemen Márton

mesteroktatóprogramtervező informatikus Bsc  
programtervező matematikus

Budapest 2021

Tartalomjegyzék

[Bevezetés 3](#_Toc60163836)

[Elméleti háttér 4](#_Toc60163837)

[Alapfogalmak 4](#_Toc60163838)

[Felhasználói dokumentáció 7](#_Toc60163839)

[A program által megoldott feladat (nem jó ez cím) 7](#_Toc60163840)

[Célközönség (nem jó ez a cím, nem kell ide) 7](#_Toc60163841)

[Rendszerkövetelmények 7](#_Toc60163842)

[Minimális hardveres környezet 7](#_Toc60163843)

[Optimális hardveres környezet 8](#_Toc60163844)

[Szoftveres követelmények 8](#_Toc60163845)

[A szerver-oldali futtatáshoz 8](#_Toc60163846)

[Kliens-oldali futtatáshoz 8](#_Toc60163847)

[Üzembe helyezés 8](#_Toc60163848)

[Hordozhatóság 12](#_Toc60163849)

[A program használata 12](#_Toc60163850)

[Hibák és egyéb üzenetek 12](#_Toc60163851)

[Turing-gép létrehozás 12](#_Toc60163852)

[Turing-gép szerkesztő és futtató felület 12](#_Toc60163853)

[Gráfszerkesztő 12](#_Toc60163854)

[Kommunikáció a szerver felé 12](#_Toc60163855)

[Fejlesztői dokumentáció 13](#_Toc60163856)

[A rendszer architektúrája 13](#_Toc60163857)

[Alkalmazott technológiák 13](#_Toc60163858)

[Alkalmazott módszerek 13](#_Toc60163859)

[Adatbázis terve 13](#_Toc60163860)

[Modul és osztályszerkezet 13](#_Toc60163861)

[A Kliens-oldal osztályszerkezet 13](#_Toc60163862)

[A felhasználó felület terve 13](#_Toc60163863)

[Telepítés fejlesztői célokra 13](#_Toc60163864)

[Megvalósítás 13](#_Toc60163865)

[Tesztelés 14](#_Toc60163866)

[Irodalomjegyzék 15](#_Toc60163867)

# 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.

# Elméleti háttér

## 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 (nem jó ez cím)

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 (nem jó ez a cím, nem kell ide)

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: minimális hardverkövetelménye okán nincs szükség komolyabb eszközre, így szinte bármilyen intézményben könnyedén alkalmazni lehet. A program segítséget nyújt a hallgatóknak az egyes előre betáplált példa feladatok  
kipróbálásában, ezek újra szerkesztésében. Továbbá segít az oktatóknak is például 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.

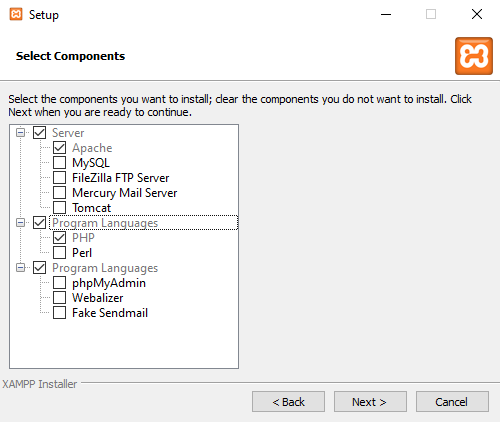
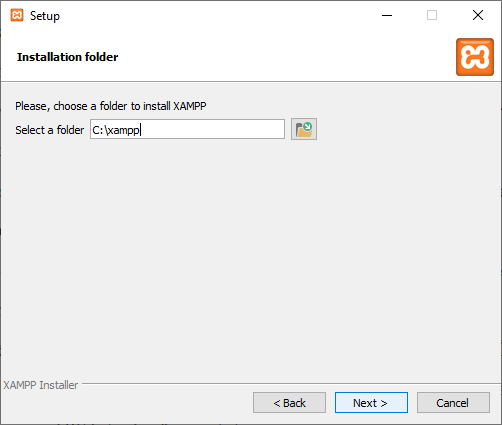
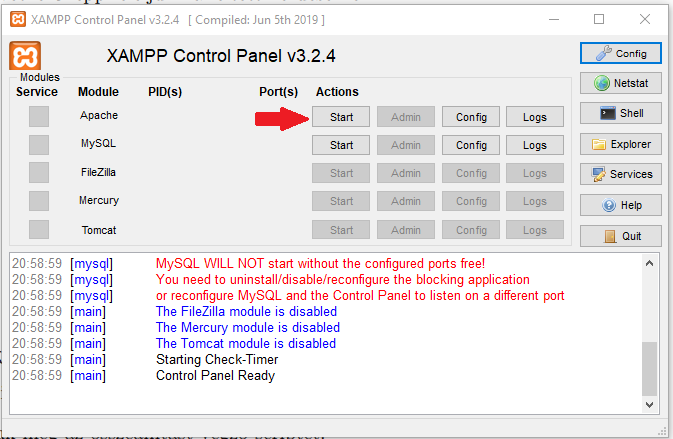
Kliens-oldali futtatáshoz

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

Ha **személyi számítógépről** szeretnénk hostolni a programot, akkor szűkség van egy webszerver szoftverre. Én a XAMPP-t tudom ajánlani én is ezt használtam a fejlesztés során.

* https://www.apachefriends.org/hu/download.html (a link 2020.12.23-án érvényben lévő) töltsük le az operációs rendszerünkhöz a megfelelőt illetve PHP 8-as verziót tartalmazzon.
* Telepítéskor lehet választani a szoftver csomag mely részei legyenek csak telepítve, ekkor a csak az Apache webservert pipáljuk be, másra nincs szükségünk.
* Ezután kérdezni fogja hol legyen a XAMPP telepítési helye, itt érdemes az operációs rendszerrel azonos lemezre telepíteni annak is főkönyvtárjába.  
  
* Ha telepítés végbement akkor, akkor lehet máris indítani a webszerververt. Ehhez megkell keresni a „xampp-control.exe” indító alkalmazást. Erre kattintva, jelenik meg a control panel ahol az Apache webservernek a start gombjára kell rá kattintani.   
  
* Ahhoz, hogy lefordítódjanak a PHP fájlok, el is kell őket helyezni egy megfelelő könyvtárban. Telepítés során ha az általam ajánlott könyvtárat vesszük akkor annak egy alkönyvtárába kell a tenni ami ezen az elérési úton van. C:\xampp\htdocs

Továbbá, ha **szerver számítógépről** szeretnék több felhasználónak elérhetővé tenni a programot akkor ellenőrizni kell rajta, van e már telepítve *PHP* fordító, ha nincs akkor érdemes a rendszer karbantartóját segítségül hívni.

**Kliens oldalon,** ha a felhasználó szeretné használni a programot, akkor csak egy böngészőre van szüksége, ez nem jelenthet nagyobb gondot, legtöbb operációs rendszerrel ellátott eszközön alapvetően van. Ha személy számítógépről történik a hostolás akkor az apache szervert indítva, a böngészőbe címsorába „localhost/szakdolgozat” -t kell begépelni. A „szakdolgozat” szót arra kell kicserélni amilyen a „C:\xampp\htdocs” alatt található alkönyvtár neve.

Ha szerver számítógépen van futtatva a szakdolgozati program, akkor egy URL link segítségével bármilyen webböngészővel ellátott eszközzel ellehet érni a programot, feltéve ha nincs jogosultsághoz kötve a szerver látogatása.

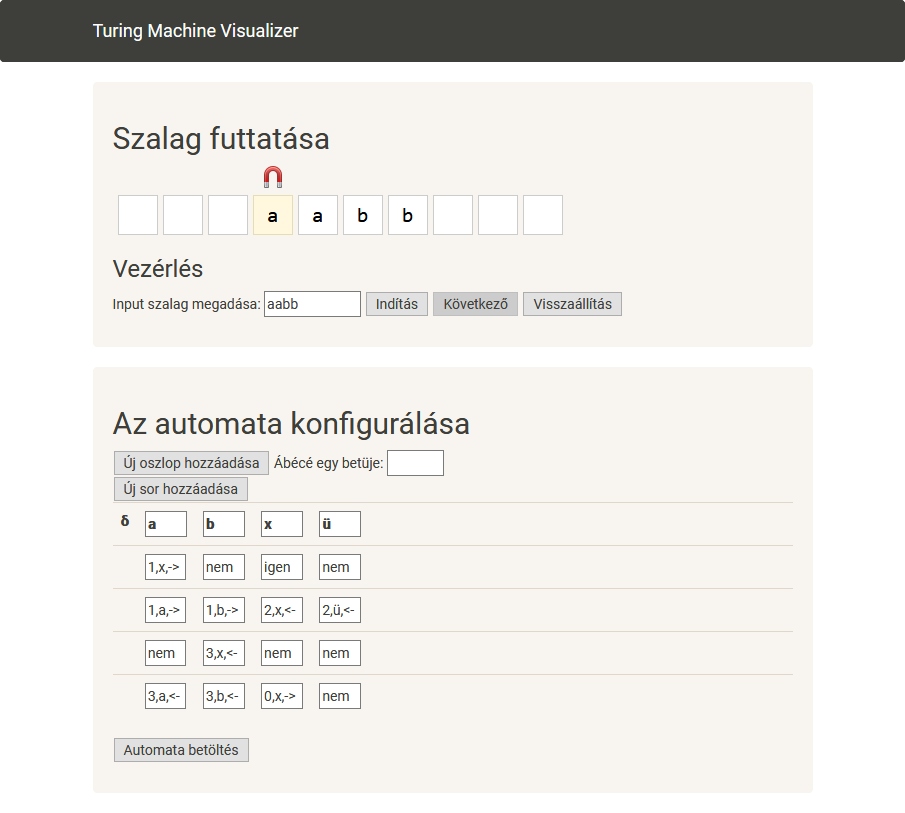
### Hordozhatóság

A program egyszerűen költöztethető. Ellenőrőzni kell futtatható e az új környezetben PHP fájl illetve még milyen régi a fordító program. Ha ezek teljesülnek akkor program forrásfájlait egyszerűen átlehet másolni egyik helyről a másikra. Semmilyen egyéb más be teendő nincs

## A program használata

### Turing-gép létrehozás

A felhasználónak a webböngészőbe kell begépelnie megfelelő URL-t ami a programra mutat. Ez lehet localhost vagy lehet egy szokványos internetre mutató link. Ha megfelelő az URL akkor az alábbi [ÁBRA] ablakával találkozik. A program 2 fő részből áll. Kezdetben ez csak részben láthatók. További elemek a program futása során kerülnek csak elő.  
 Csak akkor indítható, ha a két komponens is készen áll. Ugyanis egyik komponens a másik komponens nélkül nem adnak elég adatot ahhoz, hogy gép el tudjon indulni. Első a szalag bediktálás, második a gép konfigurációjának megadása. Ha



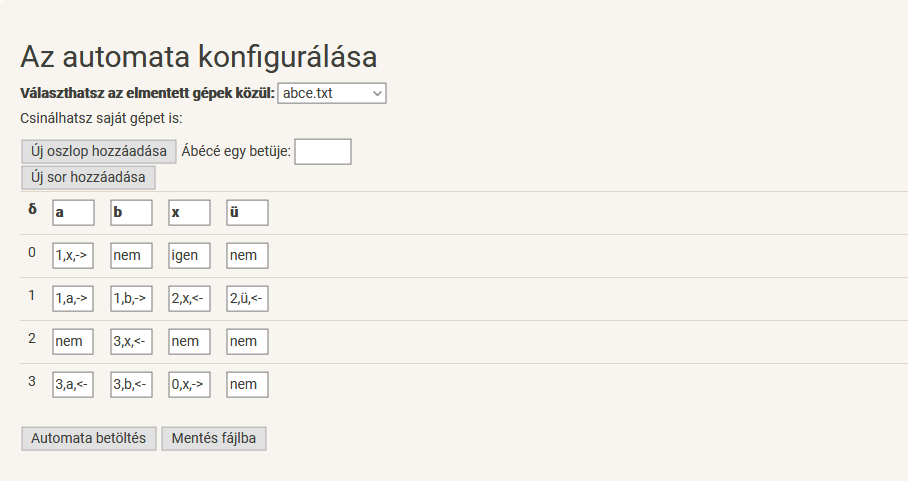
### Turing-gép szerkesztő és futtató felület

Az itt látható komponens [ÁBRA] segítségével lehet megadni a szalag input szavát. (*Input szalag megadása* ) mezőbe bele kell kattintani egérrel, és be kell gépelni egy szót ami csak betűket tartalmaz, nem lehet ékezetes, nem lehet nagy betű sem és szám sem lehet. Az angol ábácé betűt fogadja csal el. Az (*Input szalag megadása* ) mező megfelelő kitöltés után lehet csak az indítás gombra kattintani. Ha a konfiguráció is megfelelően van kitöltve akkor . algoritmust betáplálás és indítás után nem lehet szerkeszteni csak ha végbement.Lehet majd megállítani, vissza tekerni, újraínditani. Itt a felhasználó megadhatja az input ábácé szavát. Csak betűket adhat meg. Illetve korlátozva van szó hosszúsága, túl hosszú hoszú szót nem lehet megadni

### 

* **Szalag:**
* **Futtatás vezérlés:**
* **Állapot gráf:**

### Konfiguráló



### Kommunikáció a szerver felé

### Hibák és egyéb üzenetek

Minden hiba amit a program adhat, szöveges formában jelenik meg.

# Fejlesztői dokumentáció

## 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