Számításelmélet

5. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Algoritmikus problémák

Egy $\mathcal{P} : \mathcal{I} \rightarrow \mathcal{O}$ ($\mathcal{I}$ a bemenetek, $\mathcal{O}$ a kimenetek lehetséges halmaza) probléma esetén algoritmikus eljárásról vagy **algoritmikus megoldásról** akkor beszélhetünk ha sikerül találni egy olyan közös véges utasítássorozatot, amely $\mathcal{P}$ minden $I \in \mathcal{I}$ bemenetére kiszámítja a $\mathcal{P}(I) \in \mathcal{O}$ kiementet.

Algoritmikus problémák

Egy $\mathcal{P} : \mathcal{I} \to \mathcal{O}$ ($\mathcal{I}$ a bemenetek, $\mathcal{O}$ a kimenetek lehetséges halmaza) probléma esetén algoritmikus eljárásról vagy **algoritmikus megoldásról** akkor beszélhetünk ha sikerül találni egy olyan közös véges utasítássorozatot, amely $\mathcal{P}$ minden $I \in \mathcal{I}$ bemenetére kiszámítja a $\mathcal{P}(I) \in \mathcal{O}$ kiementet.

Ha $\mathcal{O} = \{$„igen”, „nem”$\}$, akkor **eldöntési problémáról**, egyébként **számítási problémáról** beszélünk.

Algoritmikus problémák

Egy $\mathcal{P} : \mathcal{I} \to \mathcal{O}$ ($\mathcal{I}$ a bemenetek, $\mathcal{O}$ a kimenetek lehetséges halmaza) probléma esetén algoritmikus eljárásról vagy **algoritmikus megoldásról** akkor beszélhetünk ha sikerül találni egy olyan közös véges utasítássorozatot, amely $\mathcal{P}$ minden $I \in \mathcal{I}$ bemenetére kiszámítja a $\mathcal{P}(I) \in \mathcal{O}$ kiementet.

Ha $\mathcal{O} = \{$„igen”, „nem”$\}$, akkor **eldöntési problémáról**, egyébként **számítási problémáról** beszélünk.

Példák:

- $\mathcal{I} = \mathbb{N} \times \mathbb{N}$, $\mathcal{O} = \mathbb{N}$. $\mathcal{P}$ az összeadás. Algoritmikus megoldás: az általános suliban tanult összeadó algoritmus.

Algoritmikus problémák

Egy $\mathcal{P} : \mathcal{I} \to \mathcal{O}$ ($\mathcal{I}$ a bemenetek, $\mathcal{O}$ a kimenetek lehetséges halmaza) probléma esetén algoritmikus eljárásról vagy **algoritmikus megoldásról** akkor beszélhetünk ha sikerül találni egy olyan közös véges utasítássorozatot, amely $\mathcal{P}$ minden $I \in \mathcal{I}$ bemenetére kiszámítja a $\mathcal{P}(I) \in \mathcal{O}$ kiementet.

Ha $\mathcal{O} = \{$„igen”, „nem”$\}$, akkor **eldöntési problémáról**, egyébként **számítási problémáról** beszélünk.

Példák:

- $\mathcal{I} = \mathbb{N} \times \mathbb{N}$, $\mathcal{O} = \mathbb{N}$. $\mathcal{P}$ az összeadás. Algoritmikus megoldás: az általános suliban tanult összeadó algoritmus.
- $\mathcal{I} = \{G \mid G$ környezetfüggetlen grammatika $\} \times T^*$, $\mathcal{O} = \{$„igen”, „nem”$\}$, ahol T egy ábécé. $\mathcal{P}$ a szóprobléma. Algoritmikus megoldás: CYK algoritmus

Algoritmikus problémák

Egy $\mathcal{P} : \mathcal{I} \to \mathcal{O}$ ($\mathcal{I}$ a bemenetek, $\mathcal{O}$ a kimenetek lehetséges halmaza) probléma esetén algoritmikus eljárásról vagy **algoritmikus megoldásról** akkor beszélhetünk ha sikerül találni egy olyan közös véges utasítássorozatot, amely $\mathcal{P}$ minden $I \in \mathcal{I}$ bemenetére kiszámítja a $\mathcal{P}(I) \in \mathcal{O}$ kiementet.

Ha $\mathcal{O} = \{$„igen”, „nem”$\}$, akkor **eldöntési problémáról**, egyébként **számítási problémáról** beszélünk.

Példák:

- $\mathcal{I} = \mathbb{N} \times \mathbb{N}$, $\mathcal{O} = \mathbb{N}$. $\mathcal{P}$ az összeadás. Algoritmikus megoldás: az általános suliban tanult összeadó algoritmus.
- $\mathcal{I} = \{G \mid G$ környezetfüggetlen grammatika $\} \times T^*$, $\mathcal{O} = \{$„igen”, „nem”$\}$, ahol T egy ábécé. $\mathcal{P}$ a szóprobléma. Algoritmikus megoldás: CYK algoritmus
- $\mathcal{I} = \{$elsőrendű formulák$\}$, $\mathcal{O} = \{$„igen”, „nem”$\}$, $\mathcal{P}(\varphi) =$„igen” pontosan akkor, ha $\models \varphi$. Nem ismeretes algoritmikus megoldás.

Algoritmikus eldöntési problémák

Adott egy $\mathcal{P}$ algoritmikus eldöntési probléma. A kérdés azon bemeneteit, amelyekre „igen” a válasz „igen”-példányoknak nevezzük. (A „nem”-példányok fogalmát hasonlóan definiáljuk.)

Algoritmikus eldöntési problémák

Adott egy $\mathcal{P}$ algoritmikus eldöntési probléma. A kérdés azon bemeneteit, amelyekre „igen" a válasz „igen"-példányoknak nevezzük. (A „nem"-példányok fogalmát hasonlóan definiáljuk.)

Az „igen" példányok $\mathcal{I}$ egy részhalmazát alkotják és egy alkalmas ábécé felett kódolva tekinthetünk rájuk egy $L(\mathcal{P}) = \{I \in \mathcal{I} \mid \mathcal{P}(I) = \text{„igen"}\}$ formális nyelvként.

Algoritmikus eldöntési problémák

Adott egy $\mathcal{P}$ algoritmikus eldöntési probléma. A kérdés azon bemeneteit, amelyekre „igen" a válasz „igen"-példányoknak nevezzük. (A „nem"-példányok fogalmát hasonlóan definiáljuk.)

Az „igen" példányok $\mathcal{I}$ egy részhalmazát alkotják és egy alkalmas ábécé felett kódolva tekinthetünk rájuk egy $L(\mathcal{P}) = \{I \in \mathcal{I} \mid \mathcal{P}(I) = \text{„igen"}\}$ formális nyelvként.

Amennyiben bizonyos „nem"-példányokra nem terminál az algoritmus **parciális algoritmusról** (megoldásról) beszélünk.

Eldöntési problémák és matematikai gépek

Az eddig tanult matematikai gépek (véges automata, veremautomata) nyelvfelismerő eszközök, azaz pontosan a felismert nyelv szavaira adnak „igen” választ.

Eldöntési problémák és matematikai gépek

Az eddig tanult matematikai gépek (véges automata, veremautomata) nyelvfelismerő eszközök, azaz pontosan a felismert nyelv szavaira adnak „igen" választ.

Mivel ezekre a gépekre úgy is gondolhatunk, hogy adott bemenetre utasítások egy véges sorozatát hajtják végre, valójában ezek a gépek az algoritmus fogalmának különböző mértékben korlátozott modelljeinek tekinthetők.

Eldöntési problémák és matematikai gépek

Az eddig tanult matematikai gépek (véges automata, veremautomata) nyelvfelismerő eszközök, azaz pontosan a felismert nyelv szavaira adnak „igen” választ.

Mivel ezekre a gépekre úgy is gondolhatunk, hogy adott bemenetre utasítások egy véges sorozatát hajtják végre, valójában ezek a gépek az algoritmus fogalmának különböző mértékben korlátozott modelljeinek tekinthetők.

Amennyiben tehát ha egy mindig termináló matematikai géppel fel tudjuk ismerni $L(\mathcal{P})$-t, akkor a $\mathcal{P}$ problémát algoritmikusan megoldottnak tekinthetjük.

Eldöntési problémák és matematikai gépek

A veremautomata a véges automata általánosítása, így az algoritmusok egy bővebb körét lehet verematomatával modellezni.

Eldöntési problémák és matematikai gépek

A veremautomata a véges automata általánosítása, így az algoritmusok egy bővebb körét lehet verematomatával modellezni.

Elég általános modell-e a veremautomata egy tetszőleges algoritmus modellezésére?

Eldöntési problémák és matematikai gépek

A veremautomata a véges automata általánosítása, így az algoritmusok egy bővebb körét lehet verematomatával modellezni.

Elég általános modell-e a veremautomata egy tetszőleges algoritmus modellezésére?

Nem, a veremautomaták a környezetfüggetlen ($\mathcal{L}_2$-beli) nyelveket ismerik fel. Ugyanakkor $\mathcal{L}_0(\supset \mathcal{L}_2)$ elemei algoritmikusan előállíthatók (például egy 0-típusú grammatika által).

Eldöntési problémák és matematikai gépek

A veremautomata a véges automata általánosítása, így az algoritmusok egy bővebb körét lehet verematomatával modellezni.

Elég általános modell-e a veremautomata egy tetszőleges algoritmus modellezésére?

Nem, a veremautomaták a környezetfüggetlen ($\mathcal{L}_2$-beli) nyelveket ismerik fel. Ugyanakkor $\mathcal{L}_0 (\supset \mathcal{L}_2)$ elemei algoritmikusan előállíthatók (például egy 0-típusú grammatika által).

Van-e tehát olyan nagyobb számítási erővel bíró matematikai gép (általánosabban nyelvleíró eszköz, számítási modell), amely éppen az algoritmus fogalmának felel meg?

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Melyik az „igazi”, melyiket válasszuk?

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Melyik az „igazi", melyiket válasszuk?

Az 1930-as évek második felétől sorra születtek olyan tételek, melyek ezen modellek megegyező számítási erejét mondták ki.

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Melyik az „igazi”, melyiket válasszuk?

Az 1930-as évek második felétől sorra születtek olyan tételek, melyek ezen modellek megegyező számítási erejét mondták ki. A későbbiek során számos további számítási modellről sikerült bebizonyítani, hogy számítási erejük a Turing gépekkel ekvivalens. Például:

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Melyik az „igazi”, melyiket válasszuk?

Az 1930-as évek második felétől sorra születtek olyan tételek, melyek ezen modellek megegyező számítási erejét mondták ki.
A későbbiek során számos további számítási modellről sikerült bebizonyítani, hogy számítási erejük a Turing gépekkel ekvivalens.
Például:

- 0. típusú grammatika

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Melyik az „igazi”, melyiket válasszuk?

Az 1930-as évek második felétől sorra születtek olyan tételek, melyek ezen modellek megegyező számítási erejét mondták ki.
A későbbiek során számos további számítási modellről sikerült bebizonyítani, hogy számítási erejük a Turing gépekkel ekvivalens. Például:

- 0. típusú grammatika
- veremautomata 2 vagy több veremmel

Az algoritmikus fogalmának modelljei

Az 1930-as évektől egyre nagyobb igény mutatkozott az algoritmus matematikai modelljének megalkotására. Egymástól függetlenül több bíztató kísérlet is született:

- Kurt Gödel: rekurzív függvények
- Alonso Church: λ-kalkulus
- Alan Turing: Turing gép

Melyik az „igazi”, melyiket válasszuk?

Az 1930-as évek második felétől sorra születtek olyan tételek, melyek ezen modellek megegyező számítási erejét mondták ki. A későbbiek során számos további számítási modellről sikerült bebizonyítani, hogy számítási erejük a Turing gépekkel ekvivalens. Például:

- 0. típusú grammatika
- veremautomata 2 vagy több veremmel
- C, Java, stb.

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

Már a 30-as években megfogalmazásra került a következő:

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

Már a 30-as években megfogalmazásra került a következő:

Church-Turing tézis

Minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing géppel is.

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

Már a 30-as években megfogalmazásra került a következő:

Church-Turing tézis

Minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing géppel is.

(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

Már a 30-as években megfogalmazásra került a következő:

Church-Turing tézis

Minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing géppel is.

(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

NEM TÉTEL!!!

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

Már a 30-as években megfogalmazásra került a következő:

Church-Turing tézis

Minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing géppel is.

(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

NEM TÉTEL!!!

A Church-Turing tézis nem bizonyítható, hiszen nem egy formális matematikai állítás, az algoritmus intuitív fogalmát használja.

A Church-Turing tézis

Valójában nem ismerünk olyan algoritmikus rendszert, amelyről tudnánk, hogy erősebb a Turing gépnél, és a legtöbb algoritmikus rendszerre bizonyított, hogy gyengébb, vagy ekvivalens.

Már a 30-as években megfogalmazásra került a következő:

Church-Turing tézis

Minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing géppel is.

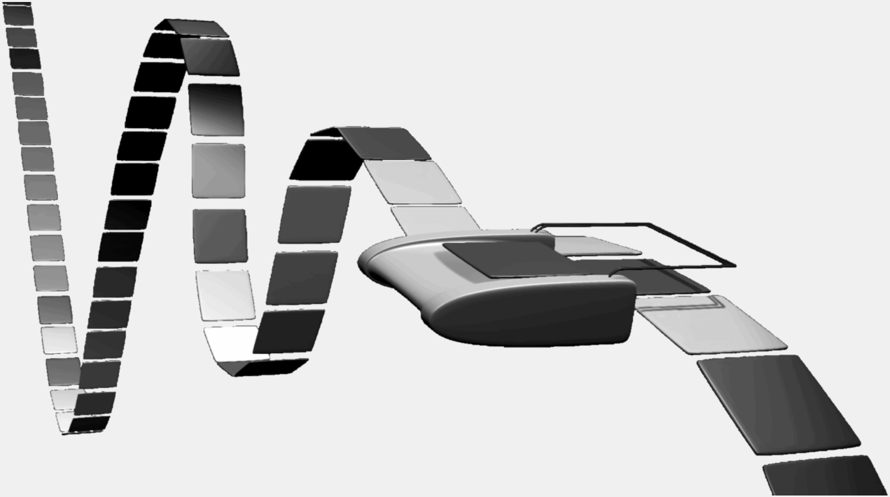
(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

NEM TÉTEL!!!

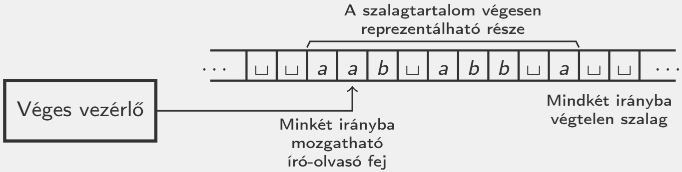
A Church-Turing tézis nem bizonyítható, hiszen nem egy formális matematikai állítás, az algoritmus intuitív fogalmát használja.

Ha elfogadjuk a tézis igazságát, a Turing gép (illetve bármely a Turing gépekkel ekvivalens modell) informálisan tekinthető az algoritmus matematikai modelljének.

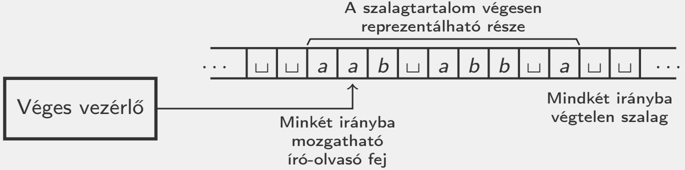
Turing gépek



Turing gépek – Informális kép

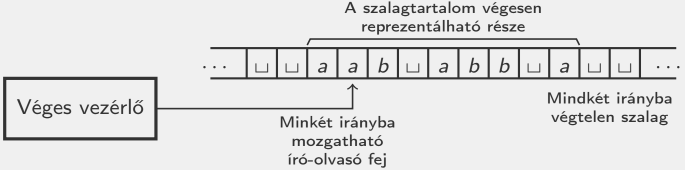


Turing gépek – Informális kép



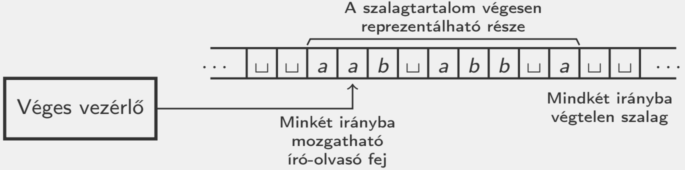
- a Turing gép (TG) az algoritmus egyik lehetséges modellje

Turing gépek – Informális kép



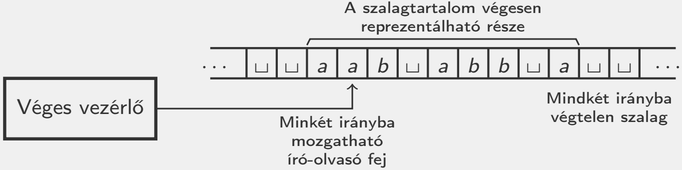
- a Turing gép (TG) az algoritmus egyik lehetséges modellje
- a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.

Turing gépek – Informális kép



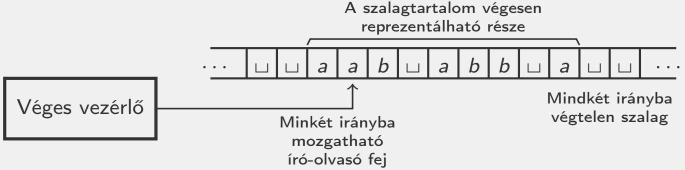
- a Turing gép (TG) az algoritmus egyik lehetséges modellje
- a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.
- informálisan a gép részei a vezérlőegység (véges sok állapottal), egy mindkét irányba végtelen szalag, és egy mindkét irányba lépni képes író-olvasó fej

Turing gépek – Informális kép



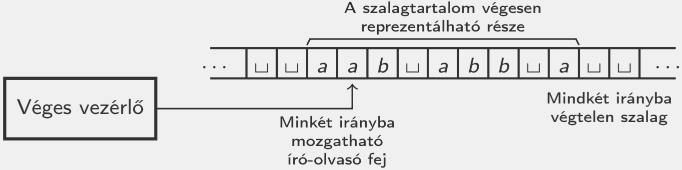
- a Turing gép (TG) az algoritmus egyik lehetséges modellje
- a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.
- informálisan a gép részei a vezérlőegység (véges sok állapottal), egy mindkét irányba végtelen szalag, és egy mindkét irányba lépni képes író-olvasó fej
- kezdetben egy input szó van a szalagon (ε esetén üres), a fej ennek első betűjéről indul, majd a szabályai szerint működik.

Turing gépek – Informális kép



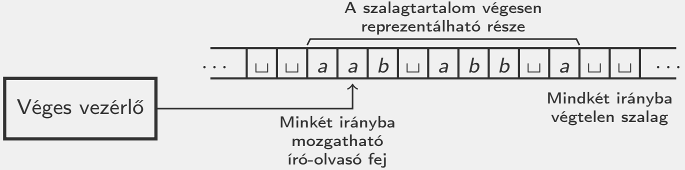
- a Turing gép (TG) az algoritmus egyik lehetséges modellje
- a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.
- informálisan a gép részei a vezérlőegység (véges sok állapottal), egy mindkét irányba végtelen szalag, és egy mindkét irányba lépni képes író-olvasó fej
- kezdetben egy input szó van a szalagon (ε esetén üres), a fej ennek első betűjéről indul, majd a szabályai szerint működik.

Turing gépek – Informális kép



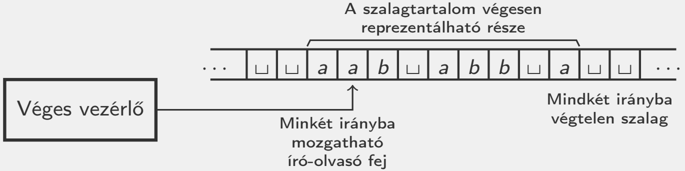
- a Turing gép (TG) az algoritmus egyik lehetséges modellje
- a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.
- informálisan a gép részei a vezérlőegység (véges sok állapottal), egy mindkét irányba végtelen szalag, és egy mindkét irányba lépni képes író-olvasó fej
- kezdetben egy input szó van a szalagon (ε esetén üres), a fej ennek első betűjéről indul, majd a szabályai szerint működik. Ha eljut az elfogadó állapotába elfogadja, ha eljut az elutasító állapotába elutasítja az inputot.

Turing gépek – Informális kép

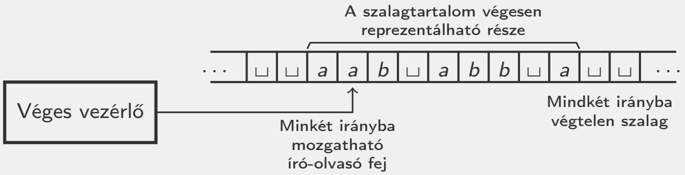


- a Turing gép (TG) az algoritmus egyik lehetséges modellje
- a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.
- informálisan a gép részei a vezérlőegység (véges sok állapottal), egy mindkét irányba végtelen szalag, és egy mindkét irányba lépni képes író-olvasó fej
- kezdetben egy input szó van a szalagon (ε esetén üres), a fej ennek első betűjéről indul, majd a szabályai szerint működik. Ha eljut az elfogadó állapotába elfogadja, ha eljut az elutasító állapotába elutasítja az inputot. Van egy harmadik lehetőség is: nem jut el soha a fenti két állapotába, "végtelen ciklusba" kerül.

Turing gépek – Informális kép

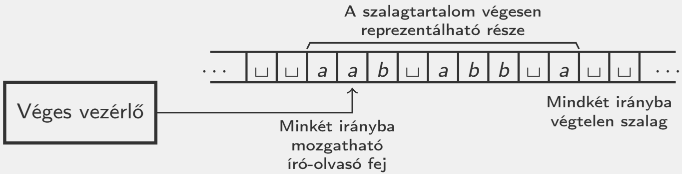


Turing gépek – Informális kép



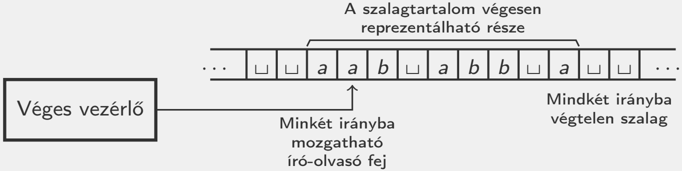
- a gép alapértelmezetten determinisztikus, minden (nem megállási) konfigurációnak van és egyértelmű a rákövetkezője.

Turing gépek – Informális kép



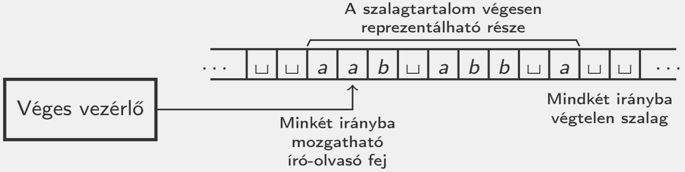
- a gép alapértelmezetten determinisztikus, minden (nem megállási) konfigurációnak van és egyértelmű a rákövetkezője.
- végtelen szalag: potenciálisan végtelen tár

Turing gépek – Informális kép



- a gép alapértelmezetten determinisztikus, minden (nem megállási) konfigurációnak van és egyértelmű a rákövetkezője.
- végtelen szalag: potenciálisan végtelen tár
- egy $\mathcal{P}$ probléma példányait egy megfelelő ábécé felett elkódolva a probléma „igen”-példányai egy $L(\mathcal{P})$ formális nyelvet alkotnak. $L(\mathcal{P})$ (és így a probléma maga is) algoritmikusan eldönthető, ha van olyan mindig termináló Turing gép, mely pontosan $L(\mathcal{P})$ szavait fogadja el.

Turing gépek – Informális kép



- a gép alapértelmezetten determinisztikus, minden (nem megállási) konfigurációnak van és egyértelmű a rákövetkezője.
- végtelen szalag: potenciálisan végtelen tár
- egy $\mathcal{P}$ probléma példányait egy megfelelő ábécé felett elkódolva a probléma „igen”-példányai egy $L(\mathcal{P})$ formális nyelvet alkotnak. $L(\mathcal{P})$ (és így a probléma maga is) algoritmikusan eldönthető, ha van olyan mindig termináló Turing gép, mely pontosan $L(\mathcal{P})$ szavait fogadja el.
- a Church-Turing tézis értelmében informálisan úgy gondolhatjuk, hogy éppen a TG-pel eldönthető problémák (nyelvek) az algoritmikusan eldönthető eldöntési problémák.

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \backslash \Sigma$.

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \backslash \Sigma$.
- $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény.

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \backslash \Sigma$.
- $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény. δ az egész $(Q \backslash \{q_i, q_n\}) \times \Gamma$-n értelmezett függvény.

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \backslash \Sigma$.
- $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény. δ az egész $(Q \backslash \{q_i, q_n\}) \times \Gamma$-n értelmezett függvény.

$\{L, S, R\}$ elemeire úgy gondolhatunk mint a TG lépéseinek irányai (balra, helyben marad, jobbra).

Turing gépek

Definíció

A **Turing gép** (továbbiakban sokszor röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes , ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \backslash \Sigma$.
- $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény. δ az egész $(Q \backslash \{q_i, q_n\}) \times \Gamma$-n értelmezett függvény.

$\{L, S, R\}$ elemeire úgy gondolhatunk mint a TG lépéseinek irányai (balra, helyben marad, jobbra). (Valójában elég lenne 2 irány: A helyben maradó lépések helyettesíthetők egy jobbra és egy balra lépéssel egy, csak erre az átmenetre használt új állapoton keresztül.)

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz:

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz:

- a szalag tartalma uv (uv előtt és után a szalagon már csak $\sqcup$ van),

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz:

- a szalag tartalma uv (uv előtt és után a szalagon már csak $\sqcup$ van),
- a gép a q állapotban van és

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz:

- a szalag tartalma uv (uv előtt és után a szalagon már csak $\sqcup$ van),
- a gép a q állapotban van és
- az író-olvasó fej a v szó első betűjén áll.

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz:

- a szalag tartalma uv (uv előtt és után a szalagon már csak $\sqcup$ van),
- a gép a q állapotban van és
- az író-olvasó fej a v szó első betűjén áll.

Két konfigurációt azonosnak tekintünk, ha csak balra/jobbra hozzáírt $\sqcup$-ekben térnek el egymástól. (Például $\sqcup abq_2\sqcup$ és $abq_2\sqcup\sqcup$.)

A Turing gépek konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

Az uqv szó az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép egy **konfigurációja** ha $q \in Q$, $u, v \in \Gamma^*$ és $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz:

- a szalag tartalma uv (uv előtt és után a szalagon már csak $\sqcup$ van),
- a gép a q állapotban van és
- az író-olvasó fej a v szó első betűjén áll.

Két konfigurációt azonosnak tekintünk, ha csak balra/jobbra hozzáírt $\sqcup$-ekben térnek el egymástól. (Például $\sqcup abq_2 \sqcup$ és $abq_2 \sqcup\sqcup$.)

Amennyiben a fej egy u szó utáni első üres cellán áll a q állapotban, akkor ennek az $uq\sqcup$ konfiguráció felel meg.

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

Elfogadó konfigurációi azon konfigurációk, melyre $q = q_i$.

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

Elfogadó konfigurációi azon konfigurációk, melyre $q = q_i$.

Elutasító konfigurációi azon konfigurációk, melyre $q = q_n$.

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

Elfogadó konfigurációi azon konfigurációk, melyre $q = q_i$.

Elutasító konfigurációi azon konfigurációk, melyre $q = q_n$.

Az elfogadó és elutasító konfigurációkat együttesen **megállási konfigurációknak** nevezzük.

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

Elfogadó konfigurációi azon konfigurációk, melyre $q = q_i$.

Elutasító konfigurációi azon konfigurációk, melyre $q = q_n$.

Az elfogadó és elutasító konfigurációkat együttesen **megállási konfigurációknak** nevezzük.

Megjegyzés: Miért $q_0 u \sqcup$ és nem $q_0 u$ a kezdőkonfiguráció?

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

Elfogadó konfigurációi azon konfigurációk, melyre $q = q_i$.

Elutasító konfigurációi azon konfigurációk, melyre $q = q_n$.

Az elfogadó és elutasító konfigurációkat együttesen **megállási konfigurációknak** nevezzük.

Megjegyzés: Miért $q_0 u \sqcup$ és nem $q_0 u$ a kezdőkonfiguráció?

> Azért, hogy ne legyen két eset. $u = \varepsilon$ esetén ugyanis $q_0 u = q_0$ nem is konfiguráció. Ha $u = \varepsilon$, akkor a fej egy tetszőleges üres celláról indulhat, azaz $q_0 \sqcup$ a kezdőkonfiguráció. Ha $u \neq \varepsilon$, akkor a $q_0 u \sqcup$ és $q_0 u$ ugyanaz a konfiguráció.

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$, $u, v \in \Gamma^*$.

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$, $u, v \in \Gamma^*$.

- Ha $\delta(q, a) = (r, b, R)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$, különben $v' = \sqcup$,
- ha $\delta(q, a) = (r, b, S)$, akkor $uqav \vdash urbv$,
- ha $\delta(q, a) = (r, b, L)$, akkor $uqav \vdash u'rcbv$, ahol $c \in \Gamma$ és $u'c = u$, ha $u \neq \varepsilon$, különben $u' = u$ és $c = \sqcup$.

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$, $u, v \in \Gamma^*$.

- Ha $\delta(q, a) = (r, b, R)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$, különben $v' = \sqcup$,
- ha $\delta(q, a) = (r, b, S)$, akkor $uqav \vdash urbv$,
- ha $\delta(q, a) = (r, b, L)$, akkor $uqav \vdash u'rcbv$, ahol $c \in \Gamma$ és $u'c = u$, ha $u \neq \varepsilon$, különben $u' = u$ és $c = \sqcup$.

Példa: Tegyük fel, hogy $\delta(q_2, a) = (q_5, b, L)$ és $\delta(q_5, c) = (q_1, \sqcup, R)$. Legyen továbbá $C_1 = bcq_2a\sqcup b$, $C_2 = bq_5cb\sqcup b$, $C_3 = b\sqcup q_1b\sqcup b$.

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$, $u, v \in \Gamma^*$.

- Ha $\delta(q, a) = (r, b, R)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$, különben $v' = \sqcup$,
- ha $\delta(q, a) = (r, b, S)$, akkor $uqav \vdash urbv$,
- ha $\delta(q, a) = (r, b, L)$, akkor $uqav \vdash u'rcbv$, ahol $c \in \Gamma$ és $u'c = u$, ha $u \neq \varepsilon$, különben $u' = u$ és $c = \sqcup$.

Példa: Tegyük fel, hogy $\delta(q_2, a) = (q_5, b, L)$ és $\delta(q_5, c) = (q_1, \sqcup, R)$. Legyen továbbá $C_1 = bcq_2a\sqcup b$, $C_2 = bq_5cb\sqcup b$, $C_3 = b\sqcup q_1b\sqcup b$.
Ekkor $C_1 \vdash C_2$ és $C_2 \vdash C_3$.

Többlépéses konfigurációátmenet

Többlépéses konfigurációátmenet: $\vdash$ reflexív, tranzitív lezártja, azaz:

Definíció

A $\vdash^* \subseteq C_M \times C_M$ **többlépéses konfigurációátmenet** relációját a következőképpen definiáljuk: $C \vdash^* C' \Leftrightarrow$

- ha $C = C'$ vagy
- ha $\exists\, n > 0 \wedge C_1, C_2, \ldots C_n \in C_M$, hogy $\forall\, 1 \leqslant i \leqslant n-1$-re $C_i \vdash C_{i+1}$ valamint $C_1 = C$ és $C_n = C'$.

Többlépéses konfigurációátmenet

Többlépéses konfigurációátmenet: $\vdash$ reflexív, tranzitív lezártja, azaz:

Definíció

A $\vdash^* \subseteq C_M \times C_M$ **többlépéses konfigurációátmenet** relációját a következőképpen definiáljuk: $C \vdash^* C' \Leftrightarrow$

- ha $C = C'$ vagy
- ha $\exists\, n > 0 \land C_1, C_2, \ldots C_n \in C_M$, hogy $\forall\, 1 \leqslant i \leqslant n-1$-re $C_i \vdash C_{i+1}$ valamint $C_1 = C$ és $C_n = C'$.

Ekvivalens definíció:

$C \vdash^* C' \Leftrightarrow$

$$(C = C') \lor (\exists\, C'' \text{ konfiguráció: } (C \vdash^* C'') \land (C'' \vdash C'))$$

Többlépéses konfigurációátmenet

Többlépéses konfigurációátmenet: $\vdash$ reflexív, tranzitív lezártja, azaz:

Definíció

A $\vdash^* \subseteq C_M \times C_M$ **többlépéses konfigurációátmenet** relációját a következőképpen definiáljuk: $C \vdash^* C' \Leftrightarrow$

- ha $C = C'$ vagy
- ha $\exists\, n > 0 \wedge C_1, C_2, \ldots C_n \in C_M$, hogy $\forall\, 1 \leqslant i \leqslant n-1$-re $C_i \vdash C_{i+1}$ valamint $C_1 = C$ és $C_n = C'$.

Ekvivalens definíció:

$$C \vdash^* C' \Leftrightarrow (C = C') \vee (\exists\, C'' \text{ konfiguráció: } (C \vdash^* C'') \wedge (C'' \vdash C'))$$

Példa: (folytatás) Legyen C_1, C_2, C_3 ugyanaz, mint a fenti példában. Mivel $C_1 \vdash C_2$ és $C_2 \vdash C_3$ is teljesült, ezért $C_1 \vdash^* C_1$, $C_1 \vdash^* C_2$, $C_1 \vdash^* C_3$ is fennállnak.

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

$L(M) = \left\{ u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon \text{ -ra} \right\}.$

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon \text{ -ra}\}$.

Figyeljük meg, hogy $L(M)$ csak Σ feletti szavakat tartalmaz.

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon \text{ -ra}\}$.

Figyeljük meg, hogy $L(M)$ csak Σ feletti szavakat tartalmaz.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **Turing-felismerhető**, ha $L = L(M)$ valamely M TG-re.

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y$ valamely $x, y \in \Gamma^*, y \neq \varepsilon$ -ra$\}$.

Figyeljük meg, hogy $L(M)$ csak Σ feletti szavakat tartalmaz.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **Turing-felismerhető**, ha $L = L(M)$ valamely M TG-re.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **eldönthető**, ha létezik olyan M TG, mely minden bemeneten megállási konfigurációba jut és $L(M) = L$.

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y$ valamely $x, y \in \Gamma^*, y \neq \varepsilon$ -ra$\}$.

Figyeljük meg, hogy $L(M)$ csak Σ feletti szavakat tartalmaz.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **Turing-felismerhető**, ha $L = L(M)$ valamely M TG-re.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **eldönthető**, ha létezik olyan M TG, mely minden bemeneten megállási konfigurációba jut és $L(M) = L$.

A Turing-felismerhető nyelveket szokás rekurzívan felsorolhatónak (vagy *parciálisan rekurzívnak*, vagy *félig eldönthetőnek*) az eldönthető nyelveket pedig rekurzívnak is nevezni.

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

$L(M) = \left\{ u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon \text{ -ra} \right\}.$

Figyeljük meg, hogy $L(M)$ csak Σ feletti szavakat tartalmaz.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **Turing-felismerhető**, ha $L = L(M)$ valamely M TG-re.

Definíció

Egy $L \subseteq \Sigma^*$ nyelv **eldönthető**, ha létezik olyan M TG, mely minden bemeneten megállási konfigurációba jut és $L(M) = L$.

A Turing-felismerhető nyelveket szokás rekurzívan felsorolhatónak (vagy *parciálisan rekurzívnak*, vagy *félig eldönthetőnek*) az eldönthető nyelveket pedig rekurzívnak is nevezni.

RE és R

A rekurzívan felsorolható nyelvek osztályát RE -vel, a rekurzív nyelvek osztályát pedig R-rel jelöljük:

RE és R

A rekurzívan felsorolható nyelvek osztályát RE -vel, a rekurzív nyelvek osztályát pedig R-rel jelöljük:

Definíció

RE=$\{L \mid \exists\, M$ Turing gép, amelyre $L(M) = L\}$.

R=$\{L \mid \exists\, M$ minden inputra megálló Turing gép, melyre $L(M) = L\}$.

Nyilván $R \subseteq RE$.

RE és R

A rekurzívan felsorolható nyelvek osztályát RE -vel, a rekurzív nyelvek osztályát pedig R-rel jelöljük:

Definíció

RE=$\{L \,|\, \exists\, M$ Turing gép, amelyre $L(M) = L\}$.

R=$\{L \,|\, \exists\, M$ minden inputra megálló Turing gép, melyre $L(M) = L\}$.

Nyilván $R \subseteq RE$.

- Igaz-e hogy minden nyelv RE-beli?
- Igaz-e hogy $R \subset RE$?

Válasz: későbbi előadáson.

A Turing gépek futási ideje

Definíció

Egy M TG **futási ideje** (időigénye) az u szóra t ($t \geqslant 0$), ha M az u-hoz tartozó kezdőkonfigurációból t lépésben (konfigurációátmenettel) jut el megállási konfigurációba. Ha nincs ilyen szám, akkor M futási ideje az u szóra végtelen.

A Turing gépek futási ideje

Definíció

Egy M TG **futási ideje** (időigénye) az u szóra t ($t \geqslant 0$), ha M az u-hoz tartozó kezdőkonfigurációból t lépésben (konfigurációátmenettel) jut el megállási konfigurációba. Ha nincs ilyen szám, akkor M futási ideje az u szóra végtelen.

Definíció

Legyen $f : \mathbb{N} \to \mathbb{N}$ egy függvény. Azt mondjuk, hogy M egy $\boldsymbol{f(n)}$ **időkorlátos** gép (vagy M $f(n)$ időigényű), ha minden $u \in \Sigma^*$ input szóra M futási ideje az u szón legfeljebb $f(|u|)$.

A Turing gépek futási ideje

Definíció

Egy M TG **futási ideje** (időigénye) az u szóra t ($t \geqslant 0$), ha M az u-hoz tartozó kezdőkonfigurációból t lépésben (konfigurációátmenettel) jut el megállási konfigurációba. Ha nincs ilyen szám, akkor M futási ideje az u szóra végtelen.

Definíció

Legyen $f : \mathbb{N} \to \mathbb{N}$ egy függvény. Azt mondjuk, hogy M egy $\boldsymbol{f(n)}$ **időkorlátos** gép (vagy M $f(n)$ időigényű), ha minden $u \in \Sigma^*$ input szóra M futási ideje az u szón legfeljebb $f(|u|)$.

Gyakran megelégszünk azzal, hogy a pontos időkorlát helyett jó aszimptotikus felső korlátot adjunk az időigényre.

Turing gépek – Példa

Feladat: Készítsünk egy M Turing gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Turing gépek – Példa

Feladat: Készítsünk egy M Turing gépet, melyre
$L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Az **átmenetdiagram**.

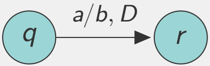
$$q \xrightarrow{a/b,\, D} r$$

megfelel a $\delta(q, a) = (r, b, D)$ átmenetnek
$(q, r \in Q, a, b \in \Gamma, D \in \{L, S, R\})$

Turing gépek – Példa

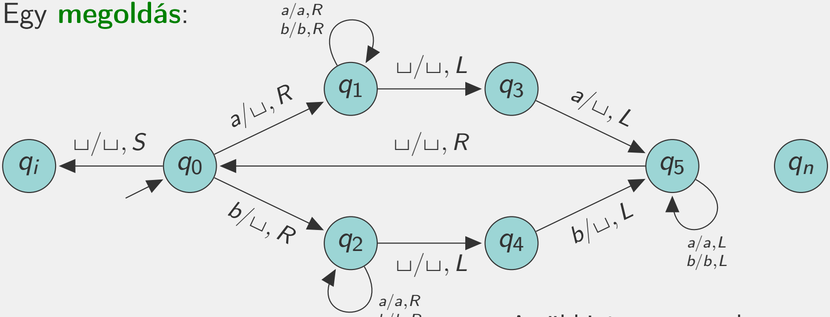
Feladat: Készítsünk egy M Turing gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Az **átmenetdiagram**.



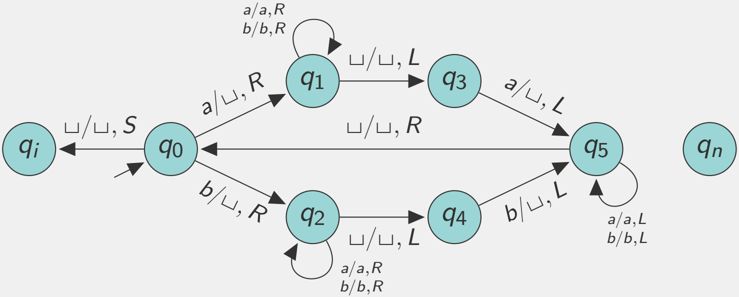
megfelel a $\delta(q, a) = (r, b, D)$ átmenetnek
$(q, r \in Q, a, b \in \Gamma, D \in \{L, S, R\})$

Egy **megoldás**:

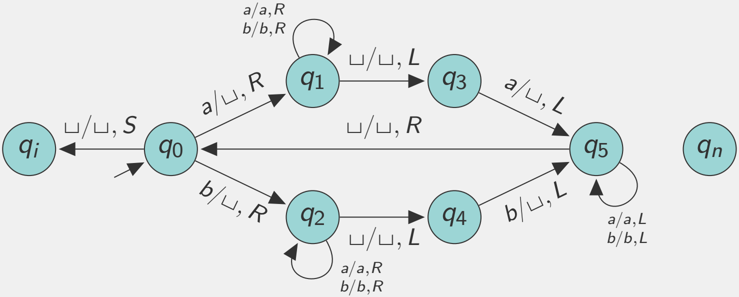


A többi átmenet q_n-be megy.

Turing gépek – Példa (folyt.)

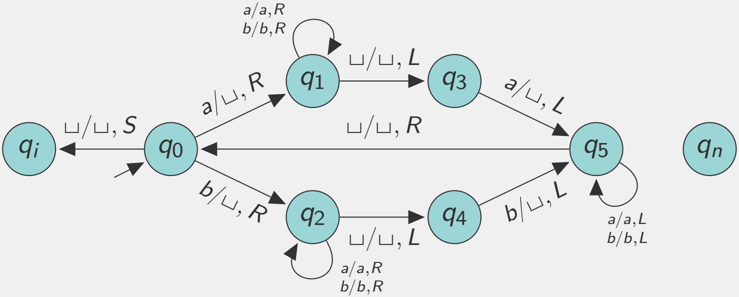


Turing gépek – Példa (folyt.)



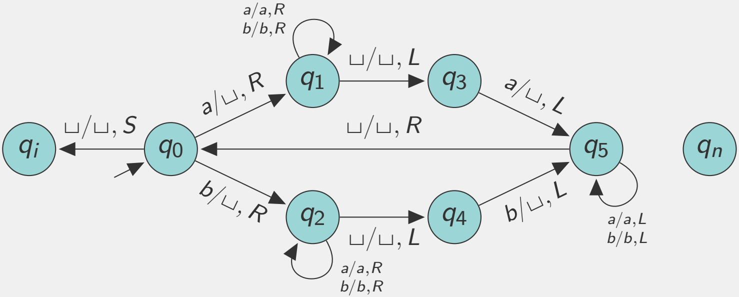
Példa. Konfigurációátmenetek sorozata az *aba* inputra:

Turing gépek – Példa (folyt.)



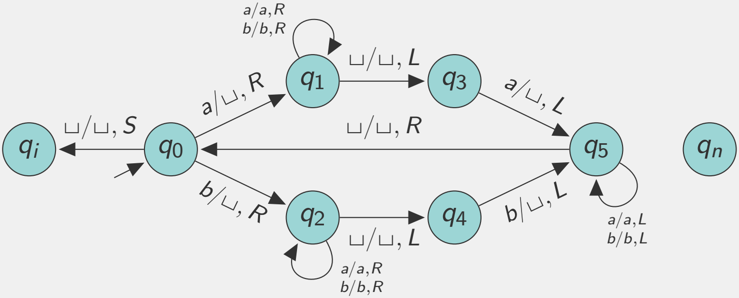
Példa. Konfigurációátmenetek sorozata az aba inputra:
q_0aba

Turing gépek – Példa (folyt.)



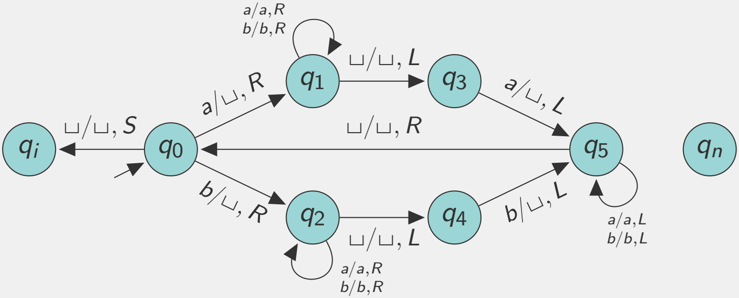
Példa. Konfigurációátmenetek sorozata az aba inputra:
$q_0 aba \vdash q_1 ba$

Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az aba inputra:
$q_0 aba \vdash q_1 ba \vdash bq_1 a$

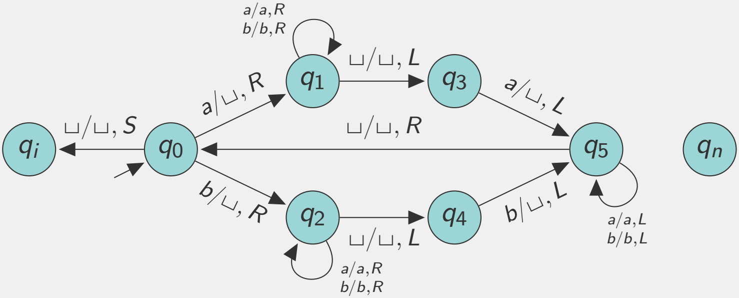
Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az aba inputra:

$q_0 aba \vdash q_1 ba \vdash b q_1 a \vdash ba q_1 \sqcup$

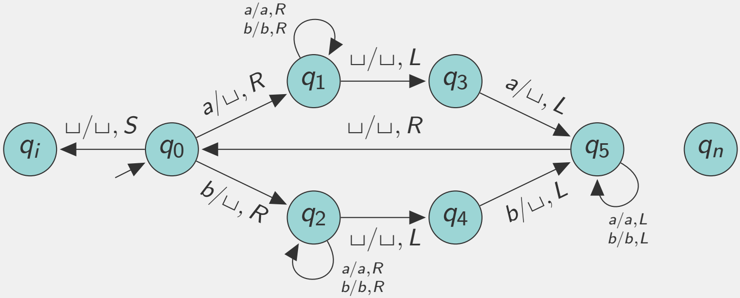
Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az *aba* inputra:

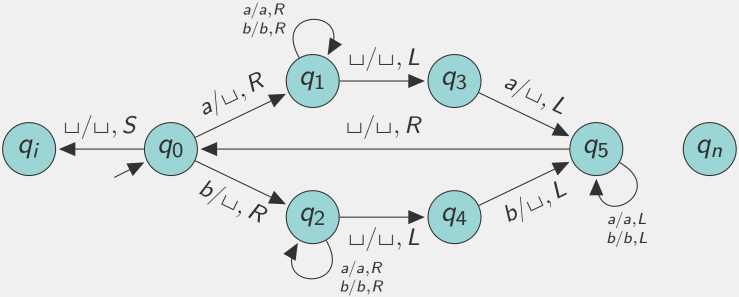
$q_0 aba \vdash q_1 ba \vdash b q_1 a \vdash ba q_1 \sqcup \vdash b q_3 a$

Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az aba inputra:
$q_0aba \vdash q_1ba \vdash bq_1a \vdash baq_1\sqcup \vdash bq_3a \vdash q_5b$

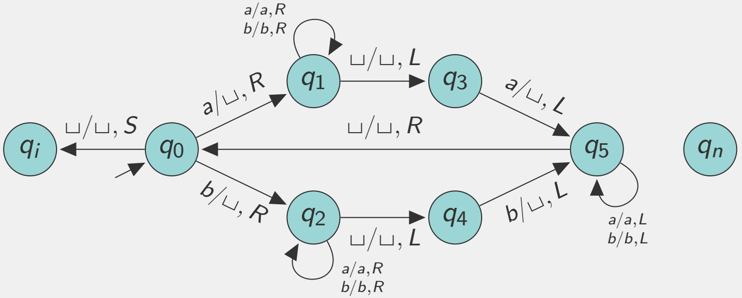
Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az *aba* inputra:

$q_0 aba \vdash q_1 ba \vdash b q_1 a \vdash ba q_1 \sqcup \vdash b q_3 a \vdash q_5 b \vdash q_5 \sqcup b$

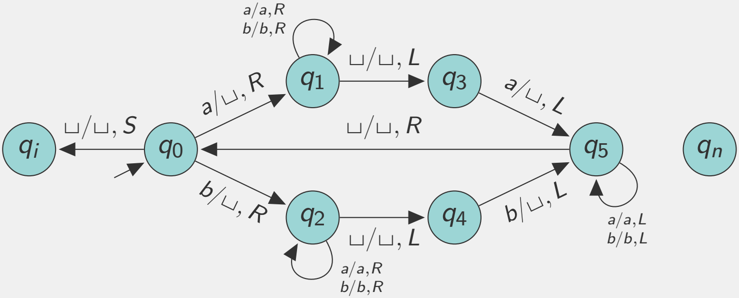
Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az *aba* inputra:

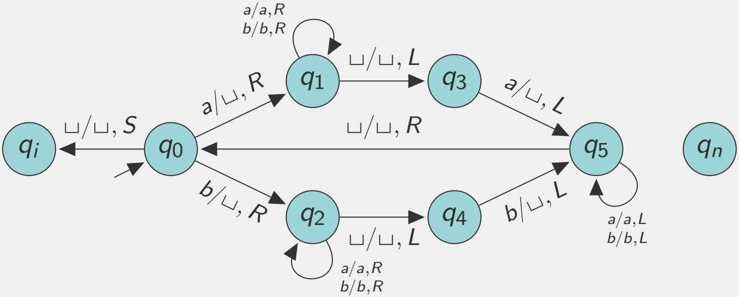
$q_0 aba \vdash q_1 ba \vdash bq_1 a \vdash baq_1 \sqcup \vdash bq_3 a \vdash q_5 b \vdash q_5 \sqcup b \vdash q_0 b$

Turing gépek – Példa (folyt.)



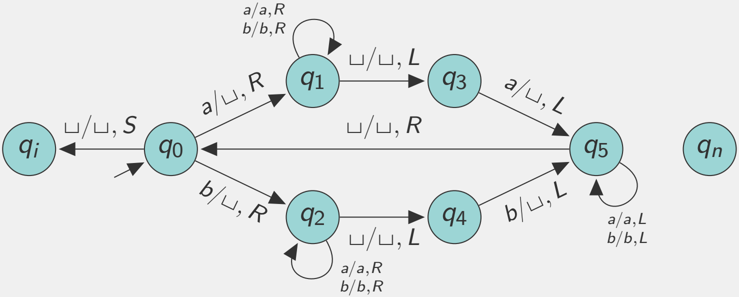
Példa. Konfigurációátmenetek sorozata az aba inputra:
$q_0aba \vdash q_1ba \vdash bq_1a \vdash baq_1\sqcup \vdash bq_3a \vdash q_5b \vdash q_5\sqcup b \vdash q_0b \vdash q_2\sqcup$

Turing gépek – Példa (folyt.)



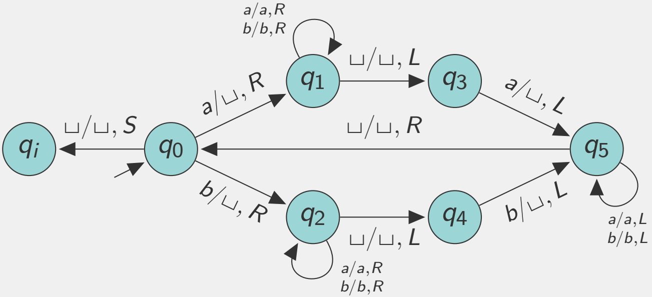
Példa. Konfigurációátmenetek sorozata az aba inputra:
$q_0 aba \vdash q_1 ba \vdash b q_1 a \vdash ba q_1 \sqcup \vdash b q_3 a \vdash q_5 b \vdash q_5 \sqcup b \vdash q_0 b \vdash$
$q_2 \sqcup \vdash q_4 \sqcup$

Turing gépek – Példa (folyt.)



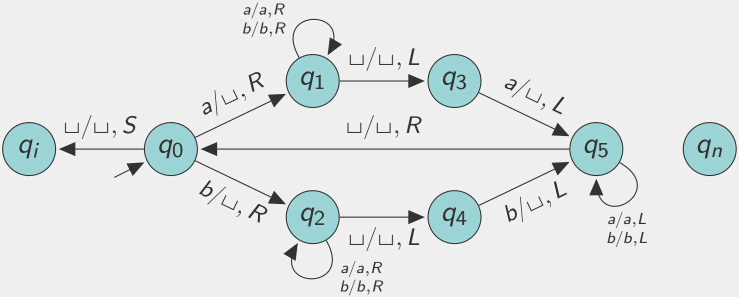
Példa. Konfigurációátmenetek sorozata az aba inputra:
$q_0aba \vdash q_1ba \vdash bq_1a \vdash baq_1\sqcup \vdash bq_3a \vdash q_5b \vdash q_5\sqcup b \vdash q_0b \vdash$
$q_2\sqcup \vdash q_4\sqcup \vdash q_n\sqcup$

Turing gépek – Példa (folyt.)



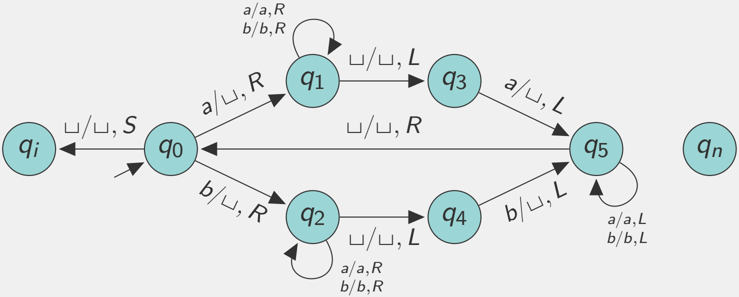
Példa. Konfigurációátmenetek sorozata az *aba* inputra:
$q_0 aba \vdash q_1 ba \vdash b q_1 a \vdash ba q_1 \sqcup \vdash b q_3 a \vdash q_5 b \vdash q_5 \sqcup b \vdash q_0 b \vdash q_2 \sqcup \vdash q_4 \sqcup \vdash q_n \sqcup$.
Az *aba* inputra 10 lépésben jut a gép megállási konfigurációba.

Turing gépek – Példa (folyt.)



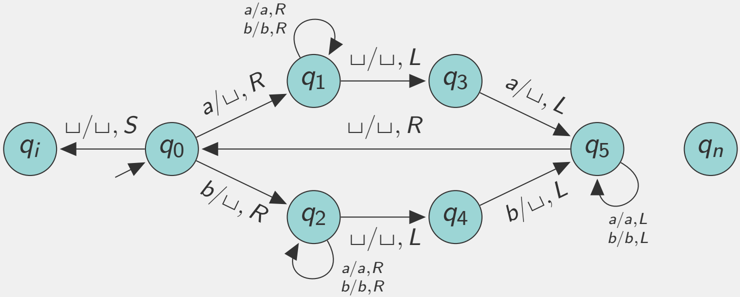
Példa. Konfigurációátmenetek sorozata az *aba* inputra:
$q_0 aba \vdash q_1 ba \vdash bq_1 a \vdash baq_1 \sqcup \vdash bq_3 a \vdash q_5 b \vdash q_5 \sqcup b \vdash q_0 b \vdash q_2 \sqcup \vdash q_4 \sqcup \vdash q_n \sqcup$.
Az *aba* inputra 10 lépésben jut a gép megállási konfigurációba.
Ebben a példában tetszőleges n-re ki tudjuk számolni a pontos időigényt is, de egyszerűbb (és gyakran elegendő) egy jó aszimptotikus felső korlát megadása.

Turing gépek – Példa (folyt.)



A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, +1 lépés q_i-be vagy q_n-be.

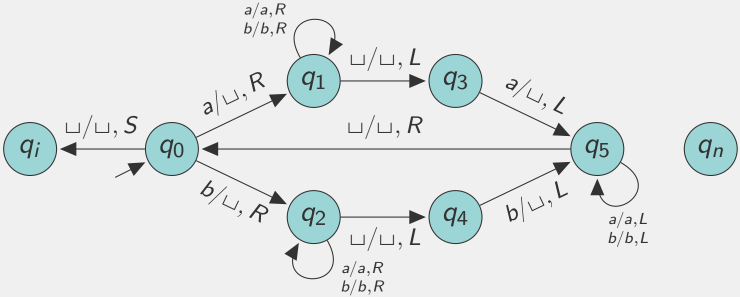
Turing gépek – Példa (folyt.)



A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

Van-e jobb aszimptotikus felső korlát?

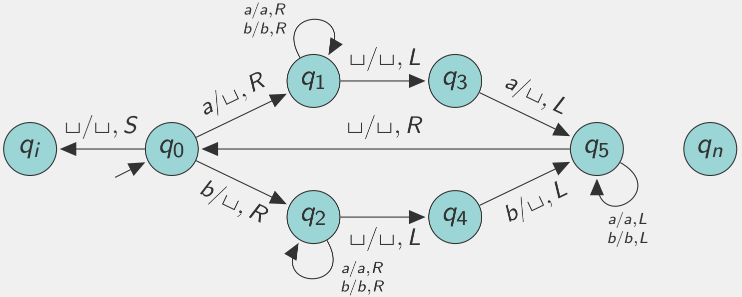
Turing gépek – Példa (folyt.)



A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

Van-e jobb aszimptotikus felső korlát? Nincs, mert van végtelen sok szó, melyre $\Omega(n^2)$-et lép.

Turing gépek – Példa (folyt.)

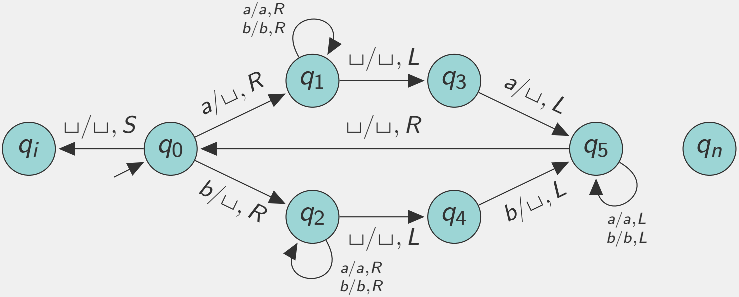


A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

Van-e jobb aszimptotikus felső korlát? Nincs, mert van végtelen sok szó, melyre $\Omega(n^2)$-et lép.

Eldönti $L = \{ww^{-1} \mid w \in \{a, b\}^*\}$-et vagy „csak” felismeri?

Turing gépek – Példa (folyt.)

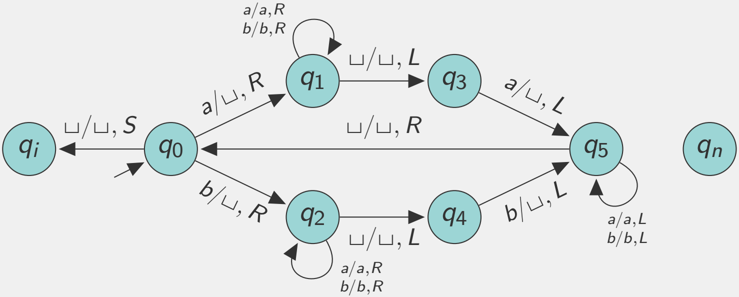


A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

Van-e jobb aszimptotikus felső korlát? Nincs, mert van végtelen sok szó, melyre $\Omega(n^2)$-et lép.

Eldönti $L = \{ww^{-1} \mid w \in \{a, b\}^*\}$-et vagy „csak" felismeri? Eldönti.

Turing gépek – Példa (folyt.)



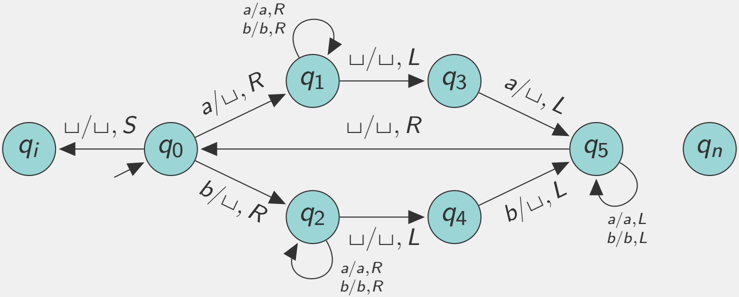
A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

Van-e jobb aszimptotikus felső korlát? Nincs, mert van végtelen sok szó, melyre $\Omega(n^2)$-et lép.

Eldönti $L = \{ww^{-1} \mid w \in \{a, b\}^*\}$-et vagy „csak" felismeri? Eldönti.

Van-e olyan TG, ami nem dönti el, de azért felismeri L-et?

Turing gépek – Példa (folyt.)



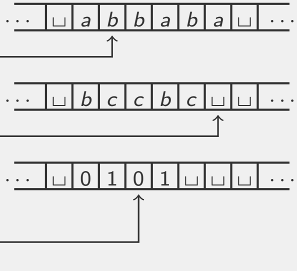
A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

Van-e jobb aszimptotikus felső korlát? Nincs, mert van végtelen sok szó, melyre $\Omega(n^2)$-et lép.

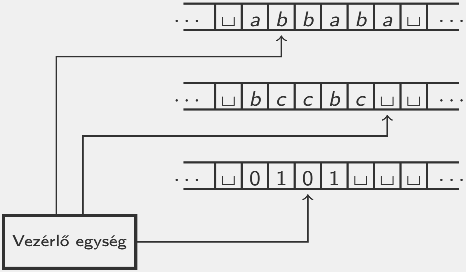
Eldönti $L = \{ww^{-1} \mid w \in \{a,b\}^*\}$-et vagy „csak" felismeri? Eldönti.

Van-e olyan TG, ami nem dönti el, de azért felismeri L-et? Igen, a q_n-be menő átmeneteket vezessük végtelen ciklusba.

Többszalagos Turing gép – Informális kép

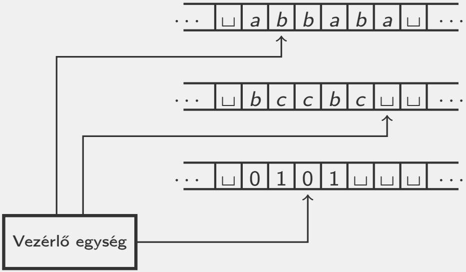


Többszalagos Turing gép – Informális kép



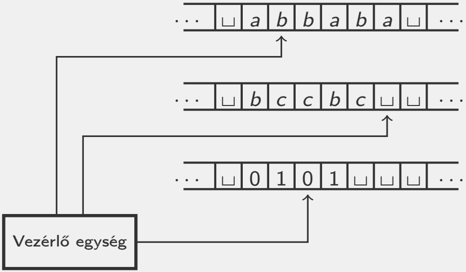
- Véges vezérlő egység, $k(\geqslant 1)$ darab kétirányba végtelen szalag, minden szalaghoz egy-egy saját író-olvasó fej.

Többszalagos Turing gép – Informális kép



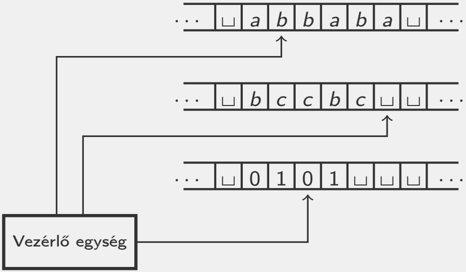
- Véges vezérlő egység, $k(\geqslant 1)$ darab kétirányba végtelen szalag, minden szalaghoz egy-egy saját író-olvasó fej.
- Egy ütem: Minden szalagról a fejek által mutatott betűk egyszerre történő beolvasása, átírása és a fejek léptetése egyszerre, de egymástól független irányokba.

Többszalagos Turing gép – Informális kép



- Véges vezérlő egység, $k (\geqslant 1)$ darab kétirányba végtelen szalag, minden szalaghoz egy-egy saját író-olvasó fej.
- Egy ütem: Minden szalagról a fejek által mutatott betűk egyszerre történő beolvasása, átírása és a fejek léptetése egyszerre, de egymástól független irányokba.
- Az egyszalagos géppel analóg elfogadás fogalom.

Többszalagos Turing gép – Informális kép



- Véges vezérlő egység, $k (\geqslant 1)$ darab kétirányba végtelen szalag, minden szalaghoz egy-egy saját író-olvasó fej.
- Egy ütem: Minden szalagról a fejek által mutatott betűk egyszerre történő beolvasása, átírása és a fejek léptetése egyszerre, de egymástól független irányokba.
- Az egyszalagos géppel analóg elfogadás fogalom.
- Az egyszalagos géppel analóg időigény fogalom (1 lépés $=$ 1 ütem).

k-szalagos Turing gép

Definíció

Adott egy $k \geqslant 1$ egész szám. A ***k*-szalagos Turing gép** egy olyan $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes, ahol

- Q az állapotok véges, nemüres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \backslash \Sigma$,
- $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma^k \to Q \times \Gamma^k \times \{L, S, R\}^k$ az átmenet függvény.

δ az egész $(Q \backslash \{q_i, q_n\}) \times \Gamma^k$-n értelmezett függvény.

k-szalagos Turing gépek konfigurációi

Definíció

k-szalagos TG **konfigurációja** egy $(q, u_1, v_1, \ldots, u_k, v_k)$ szó, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*, v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$.

k-szalagos Turing gépek konfigurációi

Definíció

k-szalagos TG **konfigurációja** egy $(q, u_1, v_1, \ldots, u_k, v_k)$ szó, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*, v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$.

Ez azt reprezentálja, hogy

- az aktuális állapot q és
- az i. szalag tartalma $u_i v_i$ $(1 \leqslant i \leqslant k)$ és
- az i. fej v_i első betűjén áll $(1 \leqslant i \leqslant k)$.

k-szalagos Turing gépek konfigurációi

Definíció

k-szalagos TG **konfigurációja** egy $(q, u_1, v_1, \ldots, u_k, v_k)$ szó, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$.

Ez azt reprezentálja, hogy

- az aktuális állapot q és
- az i. szalag tartalma $u_i v_i$ $(1 \leqslant i \leqslant k)$ és
- az i. fej v_i első betűjén áll $(1 \leqslant i \leqslant k)$.

Definíció

Az u szóhoz tartozó **kezdőkonfiguráció**: $u_i = \varepsilon$ $(1 \leqslant i \leqslant k)$, $v_1 = u\sqcup$, és $v_i = \sqcup$ $(2 \leqslant i \leqslant k)$.

k-szalagos Turing gépek konfigurációi

Definíció

k-szalagos TG **konfigurációja** egy $(q, u_1, v_1, \ldots, u_k, v_k)$ szó, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$.

Ez azt reprezentálja, hogy

- az aktuális állapot q és
- az i. szalag tartalma $u_i v_i$ $(1 \leqslant i \leqslant k)$ és
- az i. fej v_i első betűjén áll $(1 \leqslant i \leqslant k)$.

Definíció

Az u szóhoz tartozó **kezdőkonfiguráció**: $u_i = \varepsilon$ $(1 \leqslant i \leqslant k)$, $v_1 = u\sqcup$, és $v_i = \sqcup$ $(2 \leqslant i \leqslant k)$.

Azaz, az input szó az első szalagon van, ennek az első betűjéről indul az első szalag feje. A többi szalag kezdetben üres.

k-szalagos Turing gépek megállási konfigurációi

Definíció

A $(q, u_1, v_1, \ldots, u_k, v_k)$ konfiguráció, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$,

- **elfogadó konfiguráció**, ha $q = q_i$,

k-szalagos Turing gépek megállási konfigurációi

Definíció

A $(q, u_1, v_1, \ldots, u_k, v_k)$ konfiguráció, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$,

- **elfogadó konfiguráció**, ha $q = q_i$,
- **elutasító konfiguráció**, ha $q = q_n$,

k-szalagos Turing gépek megállási konfigurációi

Definíció

A $(q, u_1, v_1, \ldots, u_k, v_k)$ konfiguráció, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ $(1 \leqslant i \leqslant k)$,

- **elfogadó konfiguráció**, ha $q = q_i$,
- **elutasító konfiguráció**, ha $q = q_n$,
- **megállási konfiguráció**, ha $q = q_i$ vagy $q = q_n$.

k-szalagos TG – egylépéses konfigurációátmenet

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

k-szalagos TG – egylépéses konfigurációátmenet

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Legyen $C = (q, u_1, a_1 v_1, \ldots, u_k, a_k v_k)$ egy konfiguráció, ahol $a_i \in \Gamma,\ u_i, v_i \in \Gamma^*$ $(1 \leqslant i \leqslant k)$. Legyen továbbá $\delta(q, a_1, \ldots, a_k) = (r, b_1, \ldots, b_k, D_1, \ldots, D_k)$, ahol $q, r \in Q$, $b_i \in \Gamma, D_i \in \{L, S, R\}$ $(1 \leqslant i \leqslant k)$. Ekkor $C \vdash (r, u_1', v_1', \ldots, u_k', v_k')$, ahol minden $1 \leqslant i \leqslant k$-ra

k-szalagos TG – egylépéses konfigurációátmenet

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos Turing gép $\vdash \subseteq C_M \times C_M$ **egylépéses konfigurációátmenet** relációját az alábbiak szerint definiáljuk.

Legyen $C = (q, u_1, a_1 v_1, \ldots, u_k, a_k v_k)$ egy konfiguráció, ahol $a_i \in \Gamma$, $u_i, v_i \in \Gamma^*$ $(1 \leqslant i \leqslant k)$. Legyen továbbá $\delta(q, a_1, \ldots, a_k) = (r, b_1, \ldots, b_k, D_1, \ldots, D_k)$, ahol $q, r \in Q$, $b_i \in \Gamma, D_i \in \{L, S, R\}$ $(1 \leqslant i \leqslant k)$. Ekkor $C \vdash (r, u_1', v_1', \ldots, u_k', v_k')$, ahol minden $1 \leqslant i \leqslant k$-ra

- ha $D_i = R$, akkor $u_i' = u_i b_i$ és $v_i' = v_i$, ha $v_i \neq \varepsilon$, különben $v_i' = \sqcup$,
- ha $D_i = S$, akkor $u_i' = u_i$ és $v_i' = b_i v_i$,
- ha $D_i = L$, akkor $u_i = u_i' c$ $(c \in \Gamma)$ és $v_i' = c b_i v_i$ ha $u_i \neq \varepsilon$, különben $u_i' = \varepsilon$ és $v_i' = \sqcup b_i v_i$.

k-szalagos TG – többlépéses konfigurációátmenet

Tehát egy szalagjára vetítve a többszalagos TG pont úgy működik, mint az egyszalagos TG.

k-szalagos TG – többlépéses konfigurációátmenet

Tehát egy szalagjára vetítve a többszalagos TG pont úgy működik, mint az egyszalagos TG.

Példa:

Legyen k=2 és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v_1', u_2, b_2v_2)$, ahol $v_1' = v_1$, ha $v_1 \neq \varepsilon$, különben $v_1' = \sqcup$.

k-szalagos TG – többlépéses konfigurációátmenet

Tehát egy szalagjára vetítve a többszalagos TG pont úgy működik, mint az egyszalagos TG.

Példa:

Legyen k=2 és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v_1', u_2, b_2v_2)$, ahol $v_1' = v_1$, ha $v_1 \neq \varepsilon$, különben $v_1' = \sqcup$.

Vegyük észre, hogy a fejek nem kell, hogy egyazon irányba lépjenek.

k-szalagos TG – többlépéses konfigurációátmenet

Tehát egy szalagjára vetítve a többszalagos TG pont úgy működik, mint az egyszalagos TG.

Példa:

Legyen k=2 és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v_1', u_2, b_2v_2)$, ahol $v_1' = v_1$, ha $v_1 \neq \varepsilon$, különben $v_1' = \sqcup$.

Vegyük észre, hogy a fejek nem kell, hogy egyazon irányba lépjenek.

Definíció

A k-szalagos TG-ek **többlépéses konfigurációátmenet** relációját ugyanúgy definiáljuk, mint az egyszalagos esetben, az egylépéses konfigurációátmenet reláció reflexív, tranzitív lezártjaként.

k-szalagos TG – többlépéses konfigurációátmenet

Tehát egy szalagjára vetítve a többszalagos TG pont úgy működik, mint az egyszalagos TG.

Példa:

Legyen k=2 és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v_1', u_2, b_2v_2)$, ahol $v_1' = v_1$, ha $v_1 \neq \varepsilon$, különben $v_1' = \sqcup$.

Vegyük észre, hogy a fejek nem kell, hogy egyazon irányba lépjenek.

Definíció

A k-szalagos TG-ek **többlépéses konfigurációátmenet** relációját ugyanúgy definiáljuk, mint az egyszalagos esetben, az egylépéses konfigurációátmenet reláció reflexív, tranzitív lezártjaként.
Jelölés: $\vdash^*$.

k-szalagos TG – felismert nyelv és időigény

Definíció

Az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos TG által **felismert nyelv**:
$L(M) = \big\{ u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \ldots x_k, y_k),$
valamely $x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots, , y_k \neq \varepsilon$ -ra$\big\}$.

k-szalagos TG – felismert nyelv és időigény

Definíció

Az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos TG által **felismert nyelv**: $L(M) = \{ u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \ldots x_k, y_k)$, valamely $x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots,, y_k \neq \varepsilon$ -ra$\}$.

Azaz, csakúgy mint az egyszalagos esetben, azon inputábécé feletti szavak halmaza, melyekkel (az első szalagján) a TG-et indítva az az elfogadó, q_i állapotában áll le.

k-szalagos TG – felismert nyelv és időigény

Definíció

Az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos TG által **felismert nyelv**:
$L(M) = \{u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \ldots x_k, y_k)$, valamely $x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots, , y_k \neq \varepsilon$ -ra$\}$.

Azaz, csakúgy mint az egyszalagos esetben, azon inputábécé feletti szavak halmaza, melyekkel (az első szalagján) a TG-et indítva az az elfogadó, q_i állapotában áll le.
A k-szalagos TG-ek által **felismerhető** illetve **eldönthető** nyelvek fogalma szintén analóg az egyszalagos esettel.

k-szalagos TG – felismert nyelv és időigény

Definíció

Az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos TG által **felismert nyelv**:
$L(M) = \{ u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \ldots x_k, y_k),$
valamely $x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots,, y_k \neq \varepsilon$ -ra$\}$.

Azaz, csakúgy mint az egyszalagos esetben, azon inputábécé feletti szavak halmaza, melyekkel (az első szalagján) a TG-et indítva az az elfogadó, q_i állapotában áll le.
A k-szalagos TG-ek által **felismerhető** illetve **eldönthető** nyelvek fogalma szintén analóg az egyszalagos esettel.

Definíció

Egy k-szalagos Turing gép **futási ideje** egy u szóra a hozzá tartozó kezdőkonfigurációból egy megállási konfigurációba megtett lépések száma.

k-szalagos TG – felismert nyelv és időigény

Definíció

Az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ k-szalagos TG által **felismert nyelv**:
$L(M) = \left\{ u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \ldots x_k, y_k),\right.$
valamely $x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots, , y_k \neq \varepsilon$ -ra$\left.\right\}$.

Azaz, csakúgy mint az egyszalagos esetben, azon inputábécé feletti szavak halmaza, melyekkel (az első szalagján) a TG-et indítva az az elfogadó, q_i állapotában áll le.
A k-szalagos TG-ek által **felismerhető** illetve **eldönthető** nyelvek fogalma szintén analóg az egyszalagos esettel.

Definíció

Egy k-szalagos Turing gép **futási ideje** egy u szóra a hozzá tartozó kezdőkonfigurációból egy megállási konfigurációba megtett lépések száma.

Az **időigény** ($f(n)$ időkorlátos TG) definíciója megegyezik az egyszalagos esetnél tárgyalttal.

k-szalagos Turing gép – átmenetdiagram

A k-szalagos TG-ek **átmenetdiagramja** egy csúcs- és élcímkézett irányított gráf, melyre

$$\begin{tikzcd} q \arrow[rrr, "{a_1,\dots,a_k/b_1,\dots,b_k,D_1,\dots,D_k}"] & & & r \end{tikzcd}$$

$$\iff \delta(q, a_1, \dots, a_k) = (r, b_1, \dots, b_k, D_1, \dots, D_k)$$

$$(q, r \in Q, a_1, \dots, a_k, b_1, \dots, b_k \in \Gamma, D_1, \dots, D_k \in \{L, S, R\})$$

k-szalagos Turing gép – átmenetdiagram

A k-szalagos TG-ek **átmenetdiagramja** egy csúcs- és élcímkézett irányított gráf, melyre

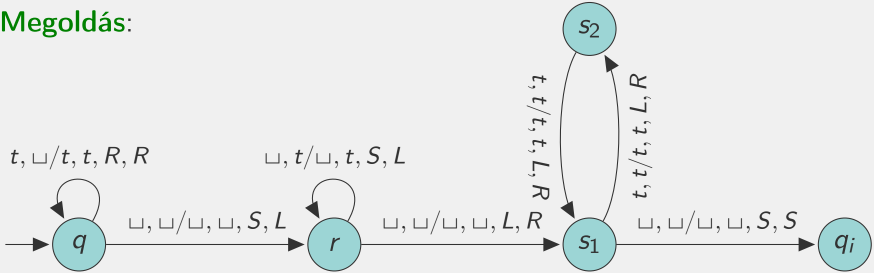
$$q \xrightarrow{a_1, \ldots, a_k / b_1, \ldots, b_k, D_1, \ldots, D_k} r$$

$$\iff \delta(q, a_1, \ldots, a_k) = (r, b_1, \ldots, b_k, D_1, \ldots, D_k)$$
$$(q, r \in Q, a_1, \ldots, a_k, b_1, \ldots, b_k \in \Gamma, D_1, \ldots, D_k \in \{L, S, R\})$$

Feladat: Készítsünk egy M kétszalagos Turing gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

k-szalagos Turing gép – példa

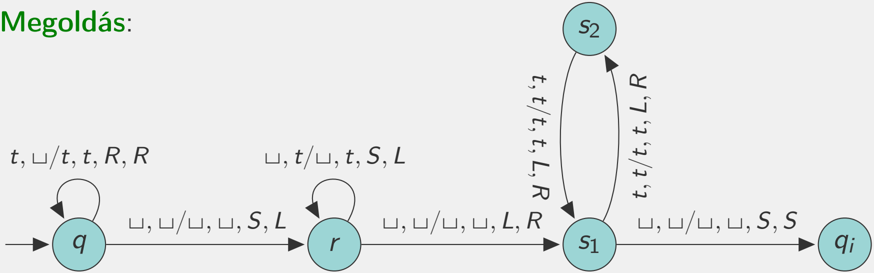
Megoldás:



$t \in \{a, b\}$ tetszőleges. A többi átmenet q_n-be megy.

k-szalagos Turing gép – példa

Megoldás:

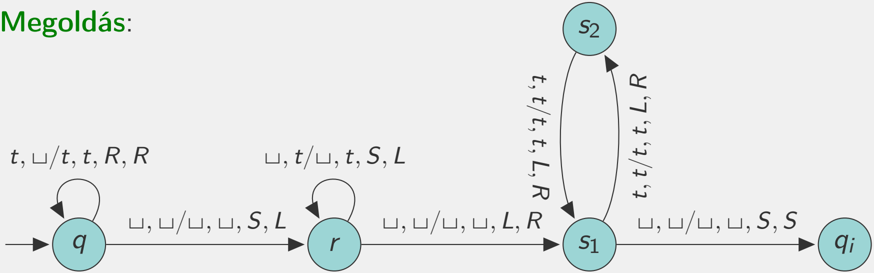


$t \in \{a, b\}$ tetszőleges. A többi átmenet q_n-be megy.

Például $(q, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, a, bba, a, \sqcup) \vdash (q, ab, ba, ab, \sqcup) \vdash$ $(q, abb, a, abb, \sqcup) \vdash (q, abba, \sqcup, abba, \sqcup) \vdash (r, abba, \sqcup, abb, a) \vdash$ $(r, abba, \sqcup, ab, ba) \vdash (r, abba, \sqcup, a, bba) \vdash (r, abba, \sqcup, \varepsilon, abba) \vdash$ $(r, abba, \sqcup, \varepsilon, \sqcup abba) \vdash (s_1, abb, a, \varepsilon, abba) \vdash (s_2, ab, ba, a, bba) \vdash$ $(s_1, a, bba, ab, ba) \vdash (s_2, \varepsilon, abba, abb, a) \vdash$ $(s_1, \varepsilon, \sqcup abba, abba, \sqcup) \vdash (q_i, \varepsilon, \sqcup abba, abba, \sqcup)$

k-szalagos Turing gép – példa

Megoldás:



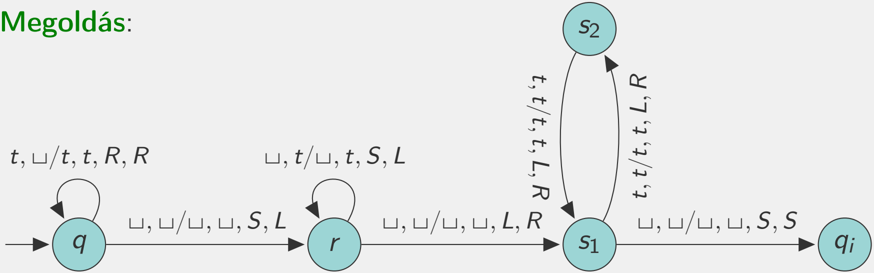
$t \in \{a, b\}$ tetszőleges. A többi átmenet q_n-be megy.

Például $(q, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, a, bba, a, \sqcup) \vdash (q, ab, ba, ab, \sqcup) \vdash$
$(q, abb, a, abb, \sqcup) \vdash (q, abba, \sqcup, abba, \sqcup) \vdash (r, abba, \sqcup, abb, a) \vdash$
$(r, abba, \sqcup, ab, ba) \vdash (r, abba, \sqcup, a, bba) \vdash (r, abba, \sqcup, \varepsilon, abba) \vdash$
$(r, abba, \sqcup, \varepsilon, \sqcup abba) \vdash (s_1, abb, a, \varepsilon, abba) \vdash (s_2, ab, ba, a, bba) \vdash$
$(s_1, a, bba, ab, ba) \vdash (s_2, \varepsilon, abba, abb, a) \vdash$
$(s_1, \varepsilon, \sqcup abba, abba, \sqcup) \vdash (q_i, \varepsilon, \sqcup abba, abba, \sqcup)$

Mennyi a TG időigénye?

k-szalagos Turing gép – példa

Megoldás:



$t \in \{a, b\}$ tetszőleges. A többi átmenet q_n-be megy.

Például $(q, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, a, bba, a, \sqcup) \vdash (q, ab, ba, ab, \sqcup) \vdash$
$(q, abb, a, abb, \sqcup) \vdash (q, abba, \sqcup, abba, \sqcup) \vdash (r, abba, \sqcup, abb, a) \vdash$
$(r, abba, \sqcup, ab, ba) \vdash (r, abba, \sqcup, a, bba) \vdash (r, abba, \sqcup, \varepsilon, abba) \vdash$
$(r, abba, \sqcup, \varepsilon, \sqcup abba) \vdash (s_1, abb, a, \varepsilon, abba) \vdash (s_2, ab, ba, a, bba) \vdash$
$(s_1, a, bba, ab, ba) \vdash (s_2, \varepsilon, abba, abb, a) \vdash$
$(s_1, \varepsilon, \sqcup abba, abba, \sqcup) \vdash (q_i, \varepsilon, \sqcup abba, abba, \sqcup)$

Mennyi a TG időigénye? Ez egy $O(n)$ időkorlátos TG, mivel egy n hosszú inputra legfeljebb $3n + 3$ lépést tesz.

k-szalagos TG szimulálása egyszalagossal

Definíció

Két TG **ekvivalens**, ha ugyanazt a nyelvet ismerik fel.

k-szalagos TG szimulálása egyszalagossal

Definíció

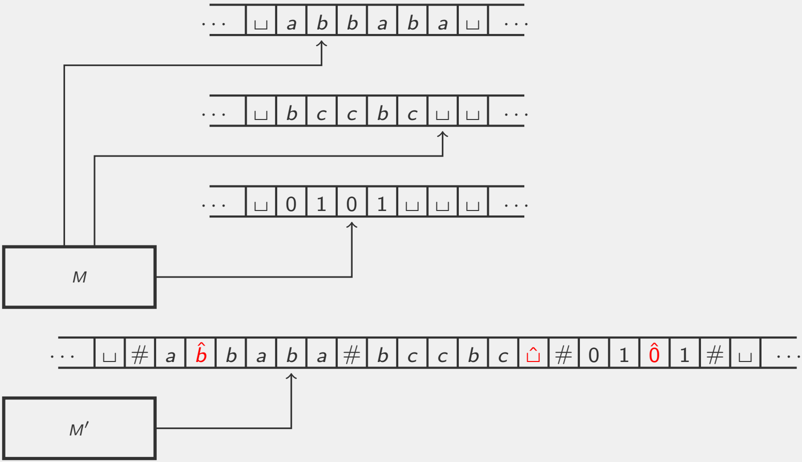
Két TG **ekvivalens**, ha ugyanazt a nyelvet ismerik fel.

Tétel

Minden M k-szalagos Turing géphez megadható egy vele ekvivalens M' egyszalagos Turing gép. Továbbá, ha M legalább lineáris időigényű $f(n)$ időkorlátos gép (azaz $f(n) = \Omega(n)$), akkor M' $O(f(n)^2)$ időkorlátos.

k-szalagos TG szimulálása egyszalagossal

Bizonyítás (vázlat): A szimuláció alapötlete



k-szalagos TG szimulálása egyszalagossal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q_0' \# \hat{a}_1 a_2 \cdots a_n \# \hat{\sqcup} \# \cdots \hat{\sqcup} \#$

k-szalagos TG szimulálása egyszalagossal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q_0'\#\hat{a}_1 a_2 \cdots a_n\#\hat{\sqcup}\# \cdots \hat{\sqcup}\#$
2. M' először végigmegy a szalagon (számolja a $\#$-okat) és eltárolja a $\hat{}$ -pal megjelölt szimbólumokat az állapotában (Például az ábrán látható esetben ha M a q állapotában van akkor M' a $(q), (q, b), (q, b, \sqcup)$ állapotokon keresztül a $(q, b, \sqcup, 0)$ állapotába kerül.)

k-szalagos TG szimulálása egyszalagossal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q_0' \# \hat{a}_1 a_2 \cdots a_n \# \hat{\sqcup} \# \cdots \hat{\sqcup} \#$
2. M' először végigmegy a szalagon (számolja a $\#$-okat) és eltárolja a $\hat{}$ -pal megjelölt szimbólumokat az állapotában (Például az ábrán látható esetben ha M a q állapotában van akkor M' a $(q), (q, b), (q, b, \sqcup)$ állapotokon keresztül a $(q, b, \sqcup, 0)$ állapotába kerül.)
3. M' mégegyszer végigmegy a szalagján és M átmenetfüggvénye alapján aktualizálja azt

k-szalagos TG szimulálása egyszalagossal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q_0' \# \hat{a}_1 a_2 \cdots a_n \# \hat{\sqcup} \# \cdots \hat{\sqcup} \#$
2. M' először végigmegy a szalagon (számolja a $\#$-okat) és eltárolja a $\hat{}$ -pal megjelölt szimbólumokat az állapotában (Például az ábrán látható esetben ha M a q állapotában van akkor M' a $(q), (q, b), (q, b, \sqcup)$ állapotokon keresztül a $(q, b, \sqcup, 0)$ állapotába kerül.)
3. M' mégegyszer végigmegy a szalagján és M átmenetfüggvénye alapján aktualizálja azt
4. ha M valamelyik szalagján nő a szalagtartalmat leíró szó hossza, akkor M'-nek az adott ponttól egy cellával jobbra kell mozgatnia a szalagja tartalmát, hogy legyen hely az új betű számára. Ez O(mozgatandó betűk száma) lépés.

k-szalagos TG szimulálása egyszalagossal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q_0' \# \hat{a}_1 a_2 \cdots a_n \# \hat{\sqcup} \# \cdots \hat{\sqcup} \#$
2. M' először végigmegy a szalagon (számolja a $\#$-okat) és eltárolja a $\hat{}$ -pal megjelölt szimbólumokat az állapotában (Például az ábrán látható esetben ha M a q állapotában van akkor M' a $(q), (q, b), (q, b, \sqcup)$ állapotokon keresztül a $(q, b, \sqcup, 0)$ állapotába kerül.)
3. M' mégegyszer végigmegy a szalagján és M átmenetfüggvénye alapján aktualizálja azt
4. ha M valamelyik szalagján nő a szalagtartalmat leíró szó hossza, akkor M'-nek az adott ponttól egy cellával jobbra kell mozgatnia a szalagja tartalmát, hogy legyen hely az új betű számára. Ez O(mozgatandó betűk száma) lépés.
5. Ha M elfogadó vagy elutasító állapotba lép, akkor M' is belép a saját elfogadó vagy elutasító állapotába

k-szalagos TG szimulálása egyszalagossal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q_0'\#\hat{a}_1 a_2 \cdots a_n\#\hat{\sqcup}\# \cdots \hat{\sqcup}\#$
2. M' először végigmegy a szalagon (számolja a $\#$-okat) és eltárolja a ˆ -pal megjelölt szimbólumokat az állapotában (Például az ábrán látható esetben ha M a q állapotában van akkor M' a $(q), (q, b), (q, b, \sqcup)$ állapotokon keresztül a $(q, b, \sqcup, 0)$ állapotába kerül.)
3. M' mégegyszer végigmegy a szalagján és M átmenetfüggvénye alapján aktualizálja azt
4. ha M valamelyik szalagján nő a szalagtartalmat leíró szó hossza, akkor M'-nek az adott ponttól egy cellával jobbra kell mozgatnia a szalagja tartalmát, hogy legyen hely az új betű számára. Ez O(mozgatandó betűk száma) lépés.
5. Ha M elfogadó vagy elutasító állapotba lép, akkor M' is belép a saját elfogadó vagy elutasító állapotába
6. Egyébként M' folytatja a szimulációt a 2-ik ponttal

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár).

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' a szalagján, legfeljebb k-szor kell egy $\sqcup$-nek helyet csinálni, ami szintén O(felhasznált cellaterület) lépésben megoldható.)

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' a szalagján, legfeljebb k-szor kell egy $\sqcup$-nek helyet csinálni, ami szintén O(felhasznált cellaterület) lépésben megoldható.)
- a felhasznált cellaterület $O(1)$-el nőtt.

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' a szalagján, legfeljebb k-szor kell egy $\sqcup$-nek helyet csinálni, ami szintén O(felhasznált cellaterület) lépésben megoldható.)
- a felhasznált cellaterület $O(1)$-el nőtt. ($\leqslant k$-val, hiszen $\leqslant k$-szor kellhet egy $\sqcup$-t beszúrni.)

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' a szalagján, legfeljebb k-szor kell egy $\sqcup$-nek helyet csinálni, ami szintén O(felhasznált cellaterület) lépésben megoldható.)
- a felhasznált cellaterület $O(1)$-el nőtt. ($\leqslant k$-val, hiszen $\leqslant k$-szor kellhet egy $\sqcup$-t beszúrni.)

Az M' által felhasznált cellaterület mérete kezdetben $\Theta(n)$, lépésenként $O(1)$-gyel nőhet, így $\leqslant f(n)$ darab lépés után az M' szalagján lévő szó hossza $O(n + f(n)O(1)) = O(n + f(n))$.

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' a szalagján, legfeljebb k-szor kell egy $\sqcup$-nek helyet csinálni, ami szintén O(felhasznált cellaterület) lépésben megoldható.)
- a felhasznált cellaterület $O(1)$-el nőtt. ($\leqslant k$-val, hiszen $\leqslant k$-szor kellhet egy $\sqcup$-t beszúrni.)

Az M' által felhasznált cellaterület mérete kezdetben $\Theta(n)$, lépésenként $O(1)$-gyel nőhet, így $\leqslant f(n)$ darab lépés után az M' szalagján lévő szó hossza $O(n + f(n)O(1)) = O(n + f(n))$.
Tehát M minden egyes lépésének M' általi szimulációja $O(n + f(n))$ lépés.

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' a szalagján, legfeljebb k-szor kell egy $\sqcup$-nek helyet csinálni, ami szintén O(felhasznált cellaterület) lépésben megoldható.)
- a felhasznált cellaterület $O(1)$-el nőtt. ($\leqslant k$-val, hiszen $\leqslant k$-szor kellhet egy $\sqcup$-t beszúrni.)

Az M' által felhasznált cellaterület mérete kezdetben $\Theta(n)$, lépésenként $O(1)$-gyel nőhet, így $\leqslant f(n)$ darab lépés után az M' szalagján lévő szó hossza $O(n + f(n)O(1)) = O(n + f(n))$.
Tehát M minden egyes lépésének M' általi szimulációja $O(n + f(n))$ lépés.
Mivel bármely n hosszú szóra az M gép $\leqslant f(n)$ lépést tesz, ezt az M' gép összesen $f(n) \cdot O(n + f(n))$ lépéssel tudja szimulálni, azaz $f(n) \cdot O(n + f(n))$ időkorlátos. Ez $O(f(n)^2)$, ha $f(n) = \Omega(n)$.

Turing gép egy irányban végtelen szalaggal

Néhány tankönyv tárgyalásában a Turing gépek szalagja csak az egyik irányban végtelen.

Turing gép egy irányban végtelen szalaggal

Néhány tankönyv tárgyalásában a Turing gépek szalagja csak az egyik irányban végtelen.

- Az **egy irányban végtelen szalagos Turing gép** egy, a bal oldalán zárt szalaggal rendelkezik

Turing gép egy irányban végtelen szalaggal

Néhány tankönyv tárgyalásában a Turing gépek szalagja csak az egyik irányban végtelen.

- Az **egy irányban végtelen szalagos Turing gép** egy, a bal oldalán zárt szalaggal rendelkezik
- A fej nem tud „leesni” a bal oldalon, még ha az állapot-átmeneti függvény balra lépést ír is elő a legbaloldalibb cellán. Ilyenkor a fej helyben marad.

Turing gép egy irányban végtelen szalaggal

Néhány tankönyv tárgyalásában a Turing gépek szalagja csak az egyik irányban végtelen.

- Az **egy irányban végtelen szalagos Turing gép** egy, a bal oldalán zárt szalaggal rendelkezik
- A fej nem tud „leesni” a bal oldalon, még ha az állapot-átmeneti függvény balra lépést ír is elő a legbaloldalibb cellán. Ilyenkor a fej helyben marad.

Nyilvánvalóan minden egyirányban végtelen szalagos Turing gép könnyen szimulálható kétirányban végtelen szalagossal.

Turing gép egy irányban végtelen szalaggal

Néhány tankönyv tárgyalásában a Turing gépek szalagja csak az egyik irányban végtelen.

- Az **egy irányban végtelen szalagos Turing gép** egy, a bal oldalán zárt szalaggal rendelkezik
- A fej nem tud „leesni” a bal oldalon, még ha az állapot-átmeneti függvény balra lépést ír is elő a legbaloldalibb cellán. Ilyenkor a fej helyben marad.

Nyilvánvalóan minden egyirányban végtelen szalagos Turing gép könnyen szimulálható kétirányban végtelen szalagossal. Igaz azonban a megfordítás is:

Turing gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing géphez van vele ekvivalens egy irányban végtelen szalagos M'' Turing gép.

Turing gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing géphez van vele ekvivalens egy irányban végtelen szalagos M'' Turing gép.

Bizonyítás (vázlat):

1. Szimuláljuk M-et egy olyan M' TG-pel, ami két darab egy irányban végtelen szalaggal rendelkezik:

Turing gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing géphez van vele ekvivalens egy irányban végtelen szalagos M'' Turing gép.

Bizonyítás (vázlat):

1. Szimuláljuk M-et egy olyan M' TG-pel, ami két darab egy irányban végtelen szalaggal rendelkezik:
   M' megjelöli mindkét szalagjának első celláját egy speciális szimbólummal. Ezután M'

Turing gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing géphez van vele ekvivalens egy irányban végtelen szalagos M'' Turing gép.

Bizonyítás (vázlat):

1. Szimuláljuk M-et egy olyan M' TG-pel, ami két darab egy irányban végtelen szalaggal rendelkezik:
   M' megjelöli mindkét szalagjának első celláját egy speciális szimbólummal. Ezután M'
   – az első szalagján szimulálja M-et akkor, amikor az a fej kezdőpozícióján vagy attól jobbra dolgozik,

Turing gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing géphez van vele ekvivalens egy irányban végtelen szalagos M'' Turing gép.

Bizonyítás (vázlat):

1. Szimuláljuk M-et egy olyan M' TG-pel, ami két darab egy irányban végtelen szalaggal rendelkezik:
   M' megjelöli mindkét szalagjának első celláját egy speciális szimbólummal. Ezután M'
   – az első szalagján szimulálja M-et akkor, amikor az a fej kezdőpozícióján vagy attól jobbra dolgozik,
   – a második szalagján pedig akkor, amikor az M a fej kezdőpozíciótól balra dolgozik (ezen a szalagon az ettől a pozíciótól balra lévő szó tükörképe van)

Turing gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing géphez van vele ekvivalens egy irányban végtelen szalagos M'' Turing gép.

Bizonyítás (vázlat):

1. Szimuláljuk M-et egy olyan M' TG-pel, ami két darab egy irányban végtelen szalaggal rendelkezik:
   M' megjelöli mindkét szalagjának első celláját egy speciális szimbólummal. Ezután M'
   – az első szalagján szimulálja M-et akkor, amikor az a fej kezdőpozícióján vagy attól jobbra dolgozik,
   – a második szalagján pedig akkor, amikor az M a fej kezdőpozíciótól balra dolgozik (ezen a szalagon az ettől a pozíciótól balra lévő szó tükörképe van)
2. Szimuláljuk M'-t egy egyirányban végtelen szalagos M'' Turing géppel (az előző tételben látott bizonyításhoz hasonlóan)