

A számításelmélet alapjai II.

1. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Vázlatos tematika

- ▶ nulladrendű logika
- ▶ elsőrendű logika
- ▶ függvények aszimptotikus viselkedése
- ▶ Turing gépek (TG), alapfogalmak
- ▶ TG változatok (többszalagos, nemdet., számító, ...)
- ▶ számosság
- ▶ eldönthetatlenség, R és RE
- ▶ eldönthetetlen problémák
- ▶ bonyultságelmélet, idő- és tárbonyolultság
- ▶ NP-teljesség, NP-teljes problémák
- ▶ további bonyolultsági osztályok
- ▶ kitekintés, összefoglaló

Ítéletkalkulus (nulladrendű logika)

A modell formális kereteket biztosít olyan következtetések helyességének eldöntésére, melyek elemi állításokból (ítéletekből) épülnek fel. Az ítéletek fontos jellemzője, hogy igazságértékük (igaz/hamis) egyértelműen eldönthető. Ítéletek például a „Süt a nap” vagy a „Lemegyek a térre” de nem tekinthető ítéletnek például a „Laci magas” (mihez képest?), „Lejössz a térre?” (kérdő mondat) vagy „Bárcsak itt lennél” (óhajtó mondat). Az elemi állításokból logikai műveleteknek megfeleltethető nyelvi összekötők segítségével összetett állítások építhetők. Például „Süt a nap, de mégis otthon maradok.” (logikai és kapcsolat, konjunkció) vagy „Ha süt a nap, lemegyek a térre.” (ha ... akkor, implikáció).

Beláthatók olyan következtetések, mint:

- (1) „Ha süt a nap, lemegyek a térre.”
- (2) „Süt a nap.”
- Tehát (3) „Lemegyek a térre.”

Formulák

Definíció

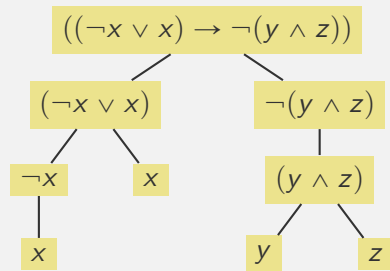
Adott **ítéletváltozók** egy előre rögzített megszámlálhatóan végtelen $\text{Var} = \{x_1, x_2, \dots\}$ halmaza. Az **ítéletlogikai formulák** Form halmaza a legszűkebb halmaz melyre

- ▶ Minden $x \in \text{Var}$ esetén $x \in \text{Form}$,
- ▶ Ha $\varphi \in \text{Form}$, akkor $\neg\varphi \in \text{Form}$,
- ▶ Ha $\varphi, \psi \in \text{Form}$, akkor $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi) \in \text{Form}$.

A műveleti jelek elnevezése: **negáció** (\neg), **konjunkció** (\wedge), **diszjunkció** (\vee), **implikáció** (\rightarrow).

Jelölés: Jelölje $\text{Var}(\varphi)$ a φ -ben előforduló itéletváltozók halmazát. Ha \mathcal{F} egy formulahalmaz, akkor $\text{Var}(\mathcal{F}) := \bigcup_{\varphi \in \mathcal{F}} \text{Var}(\varphi)$.

Szerkezeti fa, részformula, fő logikai összekötő



A **szerkezeti fa** egy csúcscímkezett bináris fa. Egy csúcs gyerekei a csúcshoz tartozó formula **közvetlen részformuláival** címkezettek. ($\neg\varphi$ esetén φ -vel címkezett az egyetlen gyerek. ($\varphi \circ \psi$) esetén két gyerek van, melyek φ -vel és ψ -vel címkezettek $\circ \in \{\wedge, \vee, \rightarrow\}$.)

Az előforduló címkék a formula **részformulái**. (A példában a sárgával megjelölt formulák.)

A **fő logikai összekötő** az az összekötő, amelyik csak a gyökérben szerepel. (A példában az \rightarrow ez az összekötő.)

Zárójelelhagyás

A zárójelelhagyás célja a formulából a lehető legtöbb zárójel elhagyása a formula szerkezetének visszaállíthatósága mellett.

\neg , \wedge , \vee , \rightarrow csökkenő precedenciasorrend

- ▶ a formula külső zárójel párja elhagyható (ha van ilyen)
- ▶ egy binér fő logikai összekötővel rendelkező részformula zárójelei elhagyhatók, ha ennek a fő logikai összekötőnek a precedenciája nagyobb, mint a szerkezeti fában szülő formula fő logikai összekötőjének precedenciája

Láncformulák zárójelelhagyása:

- ▶ Konjunkció illetve diszjunkciólánc esetén minden belső zárójelpár elhagyható. (Ennek a magyarázata a konjunkció és a diszjunkció műveletek asszociativitása, lásd mindjárt)
- ▶ Implikációlánc: $(X_1 \rightarrow (X_2 \rightarrow (X_3 \rightarrow \dots X_n)))$ az alapértelmezett zárójelezés. Csakis akkor hagyhatók el a zárójelek, ha a formula zárójelezése alapértelmezett. (Ennek az a magyarázata, hogy \rightarrow nem asszociatív.)

Zárójelelhagyás

1. Példa:

$$\neg((x \rightarrow y) \wedge z) \vee (\neg x \wedge z))$$

$$\neg((x \rightarrow y) \wedge z) \vee \neg x \wedge z$$

2. Példa:

$$x \rightarrow y \vee \neg z \rightarrow y \wedge x.$$

Melyik a fő logikai összekötő?

Visszazárójelezve:

$$(x \rightarrow ((y \vee \neg z) \rightarrow (y \wedge x))).$$

Az első \rightarrow .

Interpretáció

Definíció

Egy $I : \text{Var}(\varphi) \rightarrow \{i, h\}$ függvényt φ egy **interpretációjának** (változókiértékelésének) nevezünk.

Ha \mathcal{F} egy formulahalmaz, akkor egy $I : \text{Var}(\mathcal{F}) \rightarrow \{i, h\}$ függvényt \mathcal{F} egy **interpretációjának** (változókiértékelésének) nevezünk.

Példa: $\varphi = x \rightarrow \neg y$. Ekkor például $I(x) = i, I(y) = h$ φ egy interpretációja.

Ha $\mathcal{F} = \{x \rightarrow y, y \rightarrow z\}$, akkor $I(x) = i, I(y) = h, I(z) = h$ \mathcal{F} egy interpretációja.

A formulák igazságértéke

Egy I interpretációban egy $\varphi \in \text{Form}$ formula $\mathcal{B}_I(\varphi)$ **igazságértékét** (helyettesítési értékét, Boole értéket) a következő rekurzíval definiáljuk:

Definíció

- ha $x \in \text{Var}$ akkor $\mathcal{B}_I(x) := I(x)$,
- ha $\varphi \in \text{Form}$ formula, akkor $\mathcal{B}_I(\neg\varphi) := \neg\mathcal{B}_I(\varphi)$,
- ha $\varphi, \psi \in \text{Form}$ formulák, akkor $\mathcal{B}_I(\varphi \circ \psi) := \mathcal{B}_I(\varphi) \circ \mathcal{B}_I(\psi)$, ahol $\circ \in \{\wedge, \vee, \rightarrow\}$,

ahol a műveleteket eredményét az alábbi táblázat definiálja.

$\mathcal{B}_I(\varphi)$	$\mathcal{B}_I(\psi)$	$\mathcal{B}_I(\neg\varphi)$	$\mathcal{B}_I(\varphi \wedge \psi)$	$\mathcal{B}_I(\varphi \vee \psi)$	$\mathcal{B}_I(\varphi \rightarrow \psi)$
<i>i</i>	<i>i</i>	<i>h</i>	<i>i</i>	<i>i</i>	<i>i</i>
<i>i</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>i</i>	<i>h</i>
<i>h</i>	<i>i</i>	<i>i</i>	<i>h</i>	<i>i</i>	<i>i</i>
<i>h</i>	<i>h</i>	<i>i</i>	<i>h</i>	<i>h</i>	<i>i</i>

Az ítélettábla

$|\text{Var}(\varphi)| = n$ esetén φ -nek 2^n lehetséges interpretációja van.

Definíció

Egy φ ítéletlogikai formula **ítélettáblája** egy $2^n \times (n+1)$ -es táblázat, ahol $n = |\text{Var}(\varphi)|$. A sorok megfelelnek a lehetséges interpretációknak. Az I interpretációnak megfelelő sor az első n oszlopban tartalmazza az ítéletváltozók I szerinti kiértékelését, míg utolsó, $n+1$. oszlopa $\mathcal{B}_I(\varphi)$ -t.

Példa:

x	y	$\neg x \vee y$
<i>i</i>	<i>i</i>	<i>i</i>
<i>i</i>	<i>h</i>	<i>h</i>
<i>h</i>	<i>i</i>	<i>i</i>
<i>h</i>	<i>h</i>	<i>i</i>

Formulák szemantikus tulajdonságai

Definíció

- Egy I interpretáció **kielégít** egy φ formulát ($I \models_0 \varphi$) ha a formula helyettesítési értéke *i* az I interpretációban.
- Egy φ formula **kielégíthető**, ha legalább egy interpretáció kielégíti.
- Egy φ formula **kielégíthetetlen**, ha egyetlen interpretáció sem elégíti ki.
- Egy φ formula **tautologia** (ítéletlogikai törvény) ($\models_0 \varphi$), ha minden interpretáció kielégíti.
- Egy φ formulának a ψ formula **tautologikus következménye** ($\varphi \models_0 \psi$), ha minden φ -t kielégítő interpretáció kielégíti ψ -t is.
- φ és ψ **tautologikusan ekvivalensek** ($\varphi \sim_0 \psi$), ha $\varphi \models_0 \psi$ és $\psi \models_0 \varphi$ is teljesül.

Formulák szemantikus tulajdonságai

- Egy I interpretáció kielégít egy φ formulát ha φ ítélettáblájában I sorában az utolsó oszlopban *i* áll.
- Egy φ formula kielégíthető, ha ítélettáblájának van *i* sora.
- Egy φ formula kielégíthetetlen, ha ítélettáblájának csak *h* sora van.
- Egy φ formula tautologia, ha ítélettáblájának csak *i* sora van.
- Egy φ formulának a ψ formula tautologikus következménye, ha minden olyan I -re, amelyre φ igazságtáblájában *i* áll ott ψ is igaz.
- φ és ψ tautologikusan ekvivalensek, ha sorról sorra megegyezik az ítélettáblájuk.

Fontosabb logikai törvények

\top : tautológia, \perp : kielégíthetetlen formula.

- (a) $\neg\neg\varphi \sim_0 \varphi$,
- (b) $\varphi \vee \varphi \sim_0 \varphi$ valamint $\varphi \wedge \varphi \sim_0 \varphi$,
- (c) $\varphi \vee \psi \sim_0 \psi \vee \varphi$ valamint $\varphi \wedge \psi \sim_0 \psi \wedge \varphi$,
- (d) $(\varphi \vee \psi) \vee \xi \sim_0 \varphi \vee (\psi \vee \xi)$ valamint $(\varphi \wedge \psi) \wedge \xi \sim_0 \varphi \wedge (\psi \wedge \xi)$,
- (e) $(\varphi \vee \psi) \wedge \xi \sim_0 (\varphi \wedge \xi) \vee (\psi \wedge \xi)$ valamint $(\varphi \wedge \psi) \vee \xi \sim_0 (\varphi \vee \xi) \wedge (\psi \vee \xi)$,
- (f) $(\varphi \vee \psi) \wedge \psi \sim_0 \psi$ valamint $(\varphi \wedge \psi) \vee \psi \sim_0 \psi$,
- (g) $\varphi \rightarrow \psi \sim_0 \neg\varphi \vee \psi$,
- (h) $\neg(\varphi \wedge \psi) \sim_0 \neg\varphi \vee \neg\psi$ valamint $\neg(\varphi \vee \psi) \sim_0 \neg\varphi \wedge \neg\psi$,
- (i) $\varphi \vee \neg\varphi \sim_0 \top$ valamint $\varphi \wedge \neg\varphi \sim_0 \perp$,
- (j) $\varphi \vee \top \sim_0 \top$ valamint $\varphi \wedge \perp \sim_0 \perp$,
- (k) $\varphi \vee \perp \sim_0 \varphi$ valamint $\varphi \wedge \top \sim_0 \varphi$.

Formulák szemantikus tulajdonságai

Állítás

Legyen φ egy formula és φ_0 egy részformulája. Tegyük fel, hogy $\varphi_0 \sim_0 \psi_0$ valamely ψ_0 formulára és legyen ψ az a formula, amit φ -ból úgy kapunk, hogy a φ_0 részformulát ψ_0 -val helyettesítjük. (Például φ szerkezeti fájában az φ_0 -nak megfelelő részfat ψ_0 szerkezeti fájával helyettesítjük.) Ekkor $\varphi \sim_0 \psi$.

Ötlet: A részformulákban szereplő műveletek számára vonatkozó teljes indukcióval belátható, hogy φ és ψ részformulái megfeleltethetők egymásnak úgy, hogy minden részformulának vele ekvivalens formula feleljen meg.

Példa: Lássuk be hogy $\models_0 x \rightarrow (y \rightarrow x)$!

$$x \rightarrow (y \rightarrow x) \sim_0 \neg x \vee (\neg y \vee x) \sim_0 \neg x \vee (x \vee \neg y) \sim_0 (\neg x \vee x) \vee y \sim_0 (x \vee \neg x) \vee y \sim_0 \top \vee y \sim_0 \top.$$

Formulahalmazok szemantikus tulajdonságai

Definíció

- ▶ Egy I interpretáció **kielégít** egy \mathcal{F} formulahalmazt ($I \models_0 \mathcal{F}$), ha a formulahalmaz minden formuláját kielégíti.
- ▶ Egy \mathcal{F} formulahalmaz **kielégíthető**, ha legalább egy interpretáció kielégíti.
- ▶ Egy \mathcal{F} formulahalmaz **kielégíthetetlen**, ha nincs olyan interpretáció, ami egyszerre minden \mathcal{F} -beli formulát kielégít.
- ▶ Egy \mathcal{F} formulahalmaznak a φ formula **tautologikus következménye** ($\mathcal{F} \models_0 \varphi$), ha minden \mathcal{F} -t kielégítő interpretáció kielégíti φ -t is.

Példa: $\{x \rightarrow y, x\} \models_0 y$

x	y	$x \rightarrow y$	x	y
i	i	i	i	i
i	h	h	i	h
h	i	i	h	i
h	h	i	h	h

A szemantikus fogalmak egymással való kapcsolata

Tétel

Legyen \mathcal{F} egy formulahalmaz és φ egy formula. Akkor a következők teljesülnek.

- ▶ φ akkor és csak akkor kielégíthetetlen, ha $\neg\varphi$ tautológia.
- ▶ $\mathcal{F} \models_0 \varphi$ akkor és csak akkor, ha $\mathcal{F} \cup \{\neg\varphi\}$ kielégíthetetlen.

Bizonyítás:

- ▶ ha $I \models_0 \varphi$ akkor és csak akkor $I \not\models_0 \neg\varphi$.
- ▶ $\mathcal{F} \models_0 \varphi$ akkor és csak akkor, ha minden olyan I interpretációra, amelyre $I \models_0 \mathcal{F}$ teljesül $I \models_0 \varphi$ is fennáll, azaz $I \not\models_0 \neg\varphi$. Tehát $\mathcal{F} \models_0 \varphi$ esetén nincs olyan interpretáció, amely \mathcal{F} -et és $\neg\varphi$ -t egyszerre kielégítené. Fordítva, ha nincs olyan interpretáció, amely \mathcal{F} -et és $\neg\varphi$ -t egyszerre kielégítené, akkor minden \mathcal{F} -et kielégítő interpretáció $\neg\varphi$ -t hamisra, így φ -t igazra értékeli, azaz $\mathcal{F} \models_0 \varphi$.

A konjunktív normálforma

Definíció

- ▶ **Literálnak** nevezünk egy x vagy $\neg x$ alakú formulát, ahol $x \in \text{Var}$. x és $\neg x$ **komplement literálpár**. Egy literál **alapja** az az ítéleváltozó, amelyik a literálban szerepel.
- ▶ **Elemi diszjunkciónak** (vagy röviden **klóznak**) hívunk egy $\ell_1 \vee \dots \vee \ell_n$ alakú formulát ($n \in \mathbb{N}$), ahol ℓ_1, \dots, ℓ_n páronként különböző alapú literálok.
- ▶ **Konjunktív normálformának** (röviden KNF-nek) nevezünk egy $C_1 \wedge C_2 \wedge \dots \wedge C_m$ ($m \geq 1$) alakú formulát, ahol minden $1 \leq i \leq m$ -re C_i egy klóz (a KNF egy **tagja**).
- ▶ Az **elemi konjunkciót** és a **diszjunktív normálformát** (DNF) ezzel analóg módon definiáljuk \wedge és \vee szerepének felcserélésével.

Példa:

$x \vee \neg y \vee z$ egy klóz (és egy 1-tagú KNF egy 3 tagú DNF is egyben)
 $(x \vee \neg y \vee z) \wedge (\neg x \vee z) \wedge \neg y$ egy 3-tagú KNF.

A diszjunktív normálforma

Tétel

Minden φ ítéletkalkulusbeli formulához megadható egy vele tautológikusan ekvivalens DNF.

Bizonyítás: Legyen $\text{Var}(\varphi) = \{x_1, \dots, x_n\}$ a φ változói és $\varphi^i = \{I \mid \mathcal{B}_I(\varphi) = i\}$ a φ formula igaz halmaza.

Ekkor minden $I \in \varphi^i$ esetén

$$\psi_I := \bigwedge_{x: I(x)=i} x \wedge \bigwedge_{x: I(x)=h} \neg x$$

egy elemi konjunkció és $\psi_I^i = \{I\}$.

Tehát a $\psi = \bigvee_{I \in \varphi^i} \psi_I$ formulára

$$\psi^i = \bigcup_{I \in \varphi^i} \psi_I^i = \bigcup_{I \in \varphi^i} \{I\} = \varphi^i.$$

Tehát $\psi \sim_0 \varphi$ és ψ diszjunktív normálformájú.

A konjunktív normálforma

Tétel

Minden φ ítéletkalkulusbeli formulához megadható egy vele tautológikusan ekvivalens KNF.

Bizonyítás: Legyen $\text{Var}(\varphi) = \{x_1, \dots, x_n\}$ a φ változói és $\varphi^h = \{I \mid \mathcal{B}_I(\varphi) = h\}$ a φ formula hamis halmaza.

Ekkor minden $I \in \varphi^h$ esetén

$$\psi_I := \bigvee_{x: I(x)=i} \neg x \vee \bigvee_{x: I(x)=h} x$$

egy elemi diszjunkció és $\psi_I^h = \{I\}$.

Tehát a $\psi = \bigwedge_{I \in \varphi^h} \psi_I$ formulára

$$\psi^h = \bigcup_{I \in \varphi^h} \psi_I^h = \bigcup_{I \in \varphi^h} \{I\} = \varphi^h.$$

Tehát $\psi \sim_0 \varphi$ és ψ konjunktív normálformájú.

A konjunktív normálforma

Példa: Legyen $\varphi = (x \rightarrow y) \rightarrow z$. φ ítélet táblája:

x	y	z	φ
i	i	i	i
i	i	h	h
i	h	i	i
i	h	h	i
h	i	i	i
h	i	h	h
h	h	i	i
h	h	h	h

Tehát DNF:

$(x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (\neg x \wedge \neg y \wedge z).$

KNF: $(\neg x \vee \neg y \vee z) \wedge (x \vee \neg y \vee z) \wedge (x \vee y \vee z).$

A konjunktív normálforma

A bizonyítás konstrukciója a gyakorlatban nem nagyon használható, mivel szükséges az igaz/hamis halmaz meghatározására és mert az eredmény az input méretében akár exponenciális is lehet.

Sokszor praktikusabb az eredeti formulát átalakítani a kívánt alakra az alábbiak szerint

1. az \rightarrow operátorok eliminálása ($\varphi \rightarrow \psi \sim_0 \neg\varphi \vee \psi$)
2. \neg operátorok csak közvetlenül ítéletváltozók előtt forduljanak elő (De Morgan azonosságok és kettős tagadás törvénye)
3. a formula 2 szintűvé lapítása (disztributív szabályok)

Példa:

$$\begin{aligned}(\neg x \rightarrow y) &\rightarrow (x \wedge \neg(\neg y \wedge z)) \sim_0 \neg(x \vee y) \vee (x \wedge (y \vee \neg z)) \sim_0 \\(\neg x \wedge \neg y) &\vee (x \wedge (y \vee \neg z)) \sim_0 (\neg x \vee x) \wedge (\neg x \vee y \vee \neg z) \wedge \\(\neg y \vee x) &\wedge (\neg y \vee y \vee \neg z) \sim_0 (\neg x \vee y \vee \neg z) \wedge (\neg y \vee x).\end{aligned}$$

Ez KNF. A DNF ebből egy disztributív szabályalkalmazással adódik:

$$\begin{aligned}(\neg x \wedge \neg y) &\vee (\neg x \wedge x) \vee (y \wedge \neg y) \vee (y \wedge x) \vee (\neg z \wedge \neg y) \vee (\neg z \wedge x) \sim_0 \\(\neg x \wedge \neg y) &\vee (y \wedge x) \vee (\neg z \wedge \neg y) \vee (\neg z \wedge x).\end{aligned}$$

KNF szerepe következmények bizonyításában

Láttuk, hogy $\mathcal{F} \models_0 \varphi$ akkor és csak akkor, ha $\mathcal{F} \cup \{\neg\varphi\}$ kielégíthetetlen.

Ha $\mathcal{F} = \{F_1, \dots, F_n\}$ véges formulahalmaz, akkor $\mathcal{F} \cup \{\neg\varphi\}$ kielégíthetlensége ekvivalens $F_1 \wedge \dots \wedge F_n \wedge \neg\varphi$ kielégíthetlenségével.

Mivel a KNF külső operátora is \wedge , ezért ha a formulák KNF alakúak, akkor a feladat valójában egy klózhalmaz kielégíthetlenségének eldöntése.

Rezolvens

Rezolvens

Legyenek C_1 és C_2 pontosan 1 komplement literálpárt tartalmazó klózok. Tehát $C_1 = C'_1 \vee \ell_1$, $C_2 = C'_2 \vee \ell_2$, ahol ℓ_1 és ℓ_2 komplement literálpár, C'_1 és C'_2 viszont nem tartalmaz ilyen. A $\text{res}(C_1, C_2) := C'_1 \vee C'_2$ klózt (esetleges egyszerűsítés után) a (C_1, C_2) klózpár rezolvensének nevezzük. (Ha $C_1 = \ell_1$, $C_2 = \ell_2$, akkor $\text{res}(C_1, C_2) = \square$.)

Példa: Mi a rezolvensük?

klózpár	rezolvens
$(x \vee \textcolor{red}{y}, \neg\textcolor{red}{y} \vee z)$	$x \vee z$
$(x \vee \textcolor{red}{y} \vee z, \neg\textcolor{red}{y} \vee z)$	$x \vee z$
$(x \vee \neg y, \neg y \vee z)$	nincs: mindkét azonos alapú literál negált
$(x \vee \neg y, z \vee \neg v)$	nincs: nincs két azonos alapú literál
$(x \vee y \vee z, \neg y \vee \neg z)$	nincs: két komplement literálpár van
$(\textcolor{red}{x}, \neg\textcolor{red}{x})$	\square

Rezolúció

Rezolúciós levezetés

Egy \mathcal{S} klózhalmazból a C klóz rezolúciós levezetése egy olyan véges K_1, K_2, \dots, K_m ($m \geq 1$) klózsorozat, ahol minden $j = 1, 2, \dots, m$ -re:

- vagy $K_j \in \mathcal{S}$,
- vagy van olyan $1 \leq s, t < j$, hogy $K_j = \text{res}(K_s, K_t)$,

és $K_m = C$.

Tétel

\mathcal{S} klózhalmaz kielégíthetetlen $\iff \mathcal{S}$ -ből levezethető \square .

A bizonyítást nem részletezzük. Az egyik fontos lemma:

Lemma

Minden C_1, C_2 klózra és I interpretációjukra igaz, hogy ha $I \models_0 \{C_1, C_2\}$, akkor $I \models_0 \text{res}(C_1, C_2)$.

Rezolúció

Példa: Rezolúciós levezetéssel igazoljuk, hogy az alábbi klózalmaz kielégíthetetlen!

$$\{y \vee z, \neg x \vee w \vee \neg z, \neg y, y \vee \neg z \vee \neg w, x \vee y\}$$

Megoldás:

$$\mathcal{S} = \{y \vee z, \neg x \vee w \vee \neg z, \neg y, y \vee \neg z \vee \neg w, x \vee y\}$$

1. $\neg y$ ($\in \mathcal{S}$)
2. $y \vee z$ ($\in \mathcal{S}$)
3. z ($= \text{res}(1, 2)$)
4. $\neg x \vee w \vee \neg z$ ($\in \mathcal{S}$)
5. $y \vee \neg z \vee \neg w$ ($\in \mathcal{S}$)
6. $\neg x \vee y \vee \neg z$ ($= \text{res}(4, 5)$)
7. $\neg x \vee y$ ($= \text{res}(3, 6)$)
8. $x \vee y$ ($\in \mathcal{S}$)
9. y ($= \text{res}(7, 8)$)
10. \square ($= \text{res}(1, 9)$)

A számításelmélet alapjai II.

2. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Elsőrendű logika

A nulladrendű logika korlátozottan alkalmas a világ leírására, az egyszerű állítások belső szerkezetét nem vizsgálja. Például a „Minden ember halandó.”, „Szókratész ember.”, „Szókratész halandó.” állítások nulladrendű formalizálása esetén nincs más lehetőségünk, mint x , y és z -ként formalizálni a fenti állításokat, és így a nulladrendű logikában a 3. állítás nem következménye az első 2-nek.

Ugyanakkor jó lenne egy olyan modell, ahol a 3. állítás az első 2 következménye, hiszen az emberek halmaza részhalmaz a halandók halmazának és Szókratész az ember-halmaz egy eleme, így a halandók halmazának is eleme.

Egy elsőrendű logikában (nem véletlen a határozatlan névelő!) az állítások belső szerkezetét is figyelembe tudjuk venni. Tudunk egy halmaz összes elemére illetve legalább egy elemére vonatkozó állításokat formalizálni.

Elsőrendű logika – szintaxis

Definiálni fogunk két nyelvet a termek Term és a formulák Form nyelvét. Ehhez előbb definálunk egy megszámlálhatóan végtelen szimbólumhalmazt, a szavak betűinek a halmazát.

Definíció

Egy elsőrendű logika szimbólumhalmaza a következőkből áll

- ▶ Pred, a **predikátumszimbólumok** véges halmaza,
- ▶ Func, a **függvényyszimbólumok** véges halmaza,
- ▶ Cnst, a **konstansszimbólumok** véges halmaza,
- ▶ $\text{Ind} = \{x_1, x_2, \dots\}$, az **individuumváltozók** megszámlálhatóan végtelen halmaza
- ▶ $\{\neg, \wedge, \vee, \rightarrow, \forall, \exists\}$ műveleti jelek és kvantorok. \forall neve **univerzális kvantor**, míg \exists neve **egzisztenciális kvantor**
- ▶ $(,)$ és $,$ (vessző).

Minden $s \in \text{Pred} \cup \text{Func} \cup \text{Cnst}$ -hez hozzá van rendelve egy $\text{ar}(s) \in \mathbb{N}$ szám, a szimbólum **aritása** (a konstansokhoz mindig 0).

Elsőrendű logika – szintaxis

Definíció

A **termek** Term nyelve az a legszűkebb halmaz, amelyre

- ▶ minden $x \in \text{Ind}$ esetén $x \in \text{Term}$
- ▶ minden $c \in \text{Cnst}$ esetén $c \in \text{Term}$
- ▶ minden $f \in \text{Func}$ és $t_1, \dots, t_{\text{ar}(f)} \in \text{Term}$ esetén $f(t_1, \dots, t_{\text{ar}(f)}) \in \text{Term}$.

Definíció

Az **elsőrendű formulák** Form nyelve az a legszűkebb halmaz, amelyre

- ▶ minden $p \in \text{Pred}$ és $t_1, \dots, t_{\text{ar}(p)} \in \text{Term}$ esetén $p(t_1, \dots, t_{\text{ar}(p)}) \in \text{Form}$. Ezek az **atomi formulák**.
- ▶ Ha $\varphi \in \text{Form}$, akkor $\neg\varphi \in \text{Form}$.
- ▶ Ha $\varphi, \psi \in \text{Form}$, akkor $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi) \in \text{Form}$.
- ▶ Ha $\varphi \in \text{Form}, x \in \text{Ind}$ akkor $\forall x\varphi \in \text{Form}$ és $\exists x\varphi \in \text{Form}$.

Elsőrendű logika – szintaxis

Példa

$\text{Pred} = \{p, q\}$, $\text{Func} = \{f\}$, $\text{Cnst} = \{a\}$.

$\text{ar}(p) = \text{ar}(q) = \text{ar}(f) = 2$.

$x, a, f(x, y), f(x, f(a, x)) \in \text{Term}$.

$f(x) \notin \text{Term}$, mert $\text{ar}(f) = 1$

$p(x, y), q(x, f(a, a)), \neg p(x, f(y, z)),$
 $(\exists x p(x, y) \rightarrow q(x, z)) \in \text{Form}$.

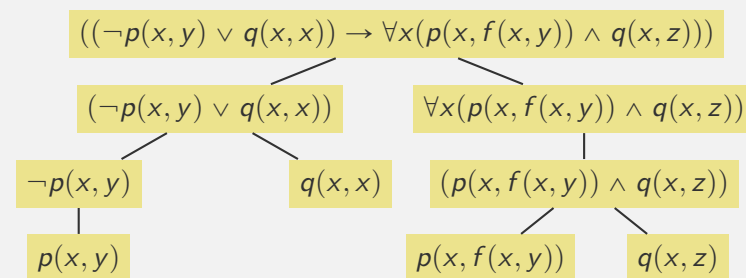
$p(x), \forall x f(x, y), p(x, q(y, z)) \notin \text{Form}$

$\varphi_1 = \forall x p(x, a) \in \text{Form}$,

$\varphi_2 = \forall x \exists y q(f(x, y), a) \in \text{Form}$,

$\varphi_3 = \forall x (\forall y q(f(y, x), y) \rightarrow p(x, a)) \in \text{Form}$.

Szerkezeti fa, részformula, fő logikai összekötő



Egy formula **szerkezeti fája** egy csúcscímkeztet bináris fa. Egy csúcs gyerekei a csúcshoz tartozó formula **közvetlen részformulaíval** címkézettek. ($\neg\varphi$ és $Qx\varphi$ esetén φ -vel címkézett az egyetlen gyerek, ahol $Q \in \{\forall, \exists\}$, $x \in \text{Ind}$. ($\varphi \circ \psi$) esetén két gyerek van, melyek φ -vel és ψ -vel címkézettek $\circ \in \{\wedge, \vee, \rightarrow\}$.)

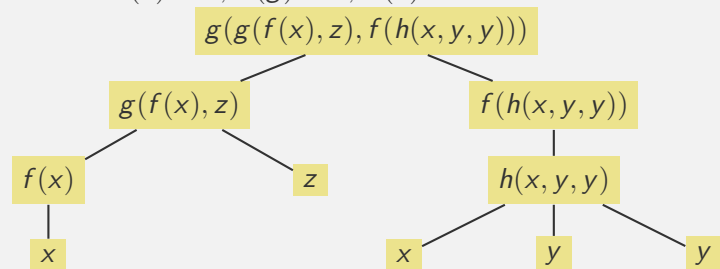
Az előforduló címkék a formula **részformulaí**. (A példában a sárgával megjelölt formulák.) A levelek atomi formulák.

A **fő logikai összekötő** az a logikai művelet vagy kvantor, amelyik csak a gyökérben szerepel. (A példában az \rightarrow ez az összekötő.)

Term szerkezeti fája, zárójelelhagyás

Term szerkezeti fája:

Példa: $\text{ar}(f) = 1, \text{ar}(g) = 2, \text{ar}(h) = 3$



Zárójelelhagyás:

Ugyanúgy, mint a nulladrendű logika esetén.

Precedenciasorrend zárójelelhagyáshoz: $\forall, \exists, \neg, \wedge, \vee, \rightarrow$.

Példa: $((\neg p(x, y) \vee q(x, x)) \rightarrow \forall x(p(x, f(x, y)) \wedge q(x, z)))$.

Mindent elhagyva, amit lehet:

$\neg p(x, y) \vee q(x, x) \rightarrow \forall x(p(x, f(x, y)) \wedge q(x, z))$.

Elsőrendű logika – interpretáció, változókiértékelés

Egy elsőrendű logika szemantikáját a szimbólumainak interpretációja és a változók kiértékelése adja meg.

Definíció

Egy elsőrendű logikai szimbólumainak **interpretációja** alatt egy $I = \langle U, I_{\text{Pred}}, I_{\text{Func}}, I_{\text{Cnst}} \rangle$ rendezett négyest értünk, ahol

- U egy tetszőleges, nemüres halmaz (univerzum),
- I_{Pred} minden $p \in \text{Pred}$ -hez hozzárendel egy $p^I \subseteq U^{\text{ar}(p)}$ $\text{ar}(p)$ -változós relációt U felett,
- I_{Func} minden $f \in \text{Func}$ -hez hozzárendel egy $f^I : U^{\text{ar}(f)} \rightarrow U$ $\text{ar}(f)$ -változós műveletet U -n,
- I_{Cnst} minden $c \in \text{Cnst}$ -hez hozzárendel egy $c^I \in U$ -t.

Definíció

Változókiértékelés alatt egy $\kappa : \text{Ind} \rightarrow U$ leképezést értünk.

Vegyük észre, hogy κ függ az U univerzumtól.

Elsőrendű logika – a termek szemantikája

Példa Az előző példát folytatva legyen $I = \langle \mathbb{N}, I_{\text{Pred}}, I_{\text{Func}}, I_{\text{Cnst}} \rangle$ egy interpretáció, ahol

$$I_{\text{Pred}}(p) = p^I, \quad (m, n) : \in p^I \Leftrightarrow m \geq n$$

$$I_{\text{Pred}}(q) = q^I, \quad (m, n) : \in q^I \Leftrightarrow m = n$$

$$I_{\text{Func}}(f) = f^I, \quad f^I(m, n) := m + n$$

$$I_{\text{Cnst}}(a) := 0,$$

legyen továbbá κ egy változókiértékelés, amelyre

$$\kappa(x) = 5, \kappa(y) = 3.$$

Definíció

Egy $t \in \text{Term}$ **értékét** egy I interpretációban a κ változókiértékelés mellett $|t|^{I, \kappa}$ jelöli és a következőképpen definiáljuk

- Ha $x \in \text{Ind}$, akkor $|x|^{I, \kappa} := \kappa(x)$,
- Ha $c \in \text{Cnst}$, akkor $|c|^{I, \kappa} := c^I$,
- $|f(t_1, t_2, \dots, t_{\text{ar}(f)})|^{I, \kappa} := f^I(|t_1|^{I, \kappa}, |t_2|^{I, \kappa}, \dots, |t_{\text{ar}(f)}|^{I, \kappa})$.

Példa Az előző példát folytatva $|f(f(x, y), y)|^{I, \kappa} = 11$.

Elsőrendű logika – a formulák szemantikája

Definíció

A κ^* változókiértékelés a κ változókiértékelés x -variánsa, ha $\kappa^*(y) = \kappa(y)$ minden $y \in \text{Ind}, y \neq x$ esetén.

Definíció

Egy $\varphi \in \text{Form}$ formula **igazságértékét** egy I interpretációban a κ változókiértékelés mellett $|\varphi|^{I, \kappa}$ jelöli és így definiáljuk:

- $|p(t_1, t_2, \dots, t_{\text{ar}(p)})|^{I, \kappa} = i \Leftrightarrow (|t_1|^{I, \kappa}, |t_2|^{I, \kappa}, \dots, |t_{\text{ar}(p)}|^{I, \kappa}) \in p^I$,
- $|\neg \varphi|^{I, \kappa} := \neg |\varphi|^{I, \kappa}$
- $|\varphi \circ \psi|^{I, \kappa} := |\varphi|^{I, \kappa} \circ |\psi|^{I, \kappa} \quad \circ \in \{\wedge, \vee, \rightarrow\}$
- $|\forall x \varphi|^{I, \kappa} = i \Leftrightarrow$ ha $|\varphi|^{I, \kappa^*} = i$ κ -nak minden κ^* x -variánsára,
- $|\exists x \varphi|^{I, \kappa} = i \Leftrightarrow$ ha $|\varphi|^{I, \kappa^*} = i$ κ -nak legalább egy κ^* x -variánsára.

A $\neg, \wedge, \vee, \rightarrow$ műveletek ugyanazok, mint az ítéletlogikánál.

Elsőrendű logika – a formulák szemantikája

Példa Az előző példát folytatva

$$|p(f(y, y), x)|^{I, \kappa} = i.$$

$$|q(f(y, y), x)|^{I, \kappa} = h.$$

$$|p(x, y) \rightarrow q(x, y)|^{I, \kappa} = h.$$

$$\varphi_1 = \forall x p(x, a),$$

Minden természetes szám ≥ 0 . $|\varphi_1|^{I, \kappa} = i$,

$$\varphi_2 = \forall x \exists y q(f(x, y), a),$$

Minden természetes számhoz hozzá tudjuk adni egy természetes számot úgy, hogy 0-t kapjunk. $|\varphi_2|^{I, \kappa} = h$,

$$\varphi_3 = \forall x (\forall y q(f(y, x), y) \rightarrow p(x, a)),$$

$\forall y q(f(y, x), y)$: az x nullelem, ez $x = 0$ -ra igaz, más x -re hamis. Viszont $p(x, a)$ minden x -re igaz, így $|\varphi_3|^{I, \kappa} = i$.

Ha $U = \mathbb{Z}$ lenne, akkor φ_2 is igaz lenne.

Elsőrendű logika – szabad és kötött előfordulás

Definíció

Legyen φ egy formula, és tekintsük $x \in \text{Ind}$ egy előfordulását φ -ben. (A kvantorokat közvetlenül követő változókat nem tekintjük ezen változó előfordulásának.) Azt mondjuk, hogy x ezen előfordulása **kötött**, ha x a φ egy $\exists x \psi$ vagy $\forall x \psi$ alakú részformulájába esik. Ellenkező esetben x ezen előfordulása **szabad**. Ha φ minden individuumváltozójának minden előfordulása kötött, akkor **zárt** formuláról beszélünk. Egyébként a formula **nyitott**.

Észrevétel: Ha φ zárt, ekkor bármely I interpretáció esetén $|\varphi|^{I, \kappa}$ értéke nem függ κ -tól. Ilyenkor $|\varphi|^{I, \kappa}$ helyett $|\varphi|^I$ írható.

Példa Az előző példában $\varphi_1, \varphi_2, \varphi_3$ zárt formulák, míg $\forall x p(x, x) \rightarrow q(x, x)$ nyitott, mert x 3. és 4. előfordulását nem tartalmazza kvantált részformula. (A formula részformulái: $\forall x p(x, x), \forall x p(x, x), p(x, x), q(x, x)$.)

Az elsőrendű logika szemantikus alapfogalmai

Definíció

- ▶ Egy φ elsőrendű logikai formula **kielégíthető**, ha van olyan I interpretáció és κ változókiértékelés, amelyre $|\varphi|^{I,\kappa} = i$, egyébként **kielégíthetetlen**.
- ▶ φ **logikailag igaz** (vagy **érvényes**), ha minden I, κ -ra, $|\varphi|^{I,\kappa} = i$, ennek jelölése $\models \varphi$.
- ▶ φ és ψ elsőrendű logikai formulák **logikailag ekvivalensek**, ha ha minden I, κ -ra, $|\varphi|^{I,\kappa} = |\psi|^{I,\kappa}$. Jelölése $\varphi \sim \psi$.
- ▶ Az \mathcal{F} formulahalmaz **kielégíthető**, ha van olyan I interpretáció és κ változókiértékelés, amelyre $|\varphi|^{I,\kappa} = i$ teljesül minden $\varphi \in \mathcal{F}$ -re, egyébként **kielégíthetetlen**.
- ▶ Az \mathcal{F} formulahalmaznak φ **logikai következménye** (jelölés: $\mathcal{F} \models \varphi$) ha minden I, κ -ra ha minden $\psi \in \mathcal{F}$ -re $|\psi|^{I,\kappa} = i$ teljesül, akkor $|\varphi|^{I,\kappa} = i$ is teljesül.

Elsőrendű logikai törvények

1. a nulladrendű törvények elsőrendben is érvényesek
2. ha x nem szabad változója A -nak
 $\forall x A \sim A$ és $\exists x A \sim A$,
3. $\forall x \forall y A \sim \forall y \forall x A$ és $\exists x \exists y A \sim \exists y \exists x A$,
4. $\neg \exists x A \sim \forall x \neg A$ és $\neg \forall x A \sim \exists x \neg A$,
5. ha x nem szabad változója A -nak
 $A \wedge \forall x B \sim \forall x (A \wedge B)$ és $A \wedge \exists x B \sim \exists x (A \wedge B)$,
 $A \vee \forall x B \sim \forall x (A \vee B)$ és $A \vee \exists x B \sim \exists x (A \vee B)$,
 $A \rightarrow \forall x B \sim \forall x (A \rightarrow B)$ és $A \rightarrow \exists x B \sim \exists x (A \rightarrow B)$,
 $\forall x B \rightarrow A \sim \exists x (B \rightarrow A)$ és $\exists x B \rightarrow A \sim \forall x (B \rightarrow A)$,
6. $\forall x A \wedge \forall x B \sim \forall x (A \wedge B)$ és $\exists x A \vee \exists x B \sim \exists x (A \vee B)$.

Elsőrendű logikai törvények

Példa: Bizonyítsuk be, hogy $\neg \exists x A \sim \forall x \neg A$!

Megoldás:

$$\begin{aligned} |\neg \exists x A|^{I,\kappa} &= h \\ \Downarrow \\ |\exists x A|^{I,\kappa} &= i \\ \Downarrow \\ \kappa\text{-nak van olyan } \kappa^* \text{ x-variánsa, amelyre } |A|^{I,\kappa^*} &= i \\ \Downarrow \\ \kappa\text{-nak van olyan } \kappa^* \text{ x-variánsa, amelyre } |\neg A|^{I,\kappa^*} &= h \\ \Downarrow \\ |\forall x \neg A|^{I,\kappa} &= h \end{aligned}$$

Ugyanazon (I, κ) (interpretáció, változókiértékelés)-párokra hamis a két formula, tehát valóban logikailag ekvivalensek. \square

Megjegyzés: A bizonyítás egyik nehézsége: **végtelen sok (I, κ) pár van.**

Elsőrendű következmény

Példa: Igazoljuk formálisan a bevezetőben említett következtetést!

Megoldás:

Először is, a formalizáláshoz legyenek

$E(x)$: x ember

$H(x)$: x halandó

s : Szókratész (konstans)

„Minden ember halandó.”

$\forall x (E(x) \rightarrow H(x))$

„Szókratész ember.”

$E(s)$

„Szókratész halandó.”

$H(s)$

Azt kell belátni, hogy $\{\forall x (E(x) \rightarrow H(x)), E(s)\} \models H(s)$, azaz hogy minden I, κ -ra amelyre $|\forall x (E(x) \rightarrow H(x))|^{I,\kappa} = i$ és $|E(s)|^{I,\kappa} = i$ teljesül $|H(s)|^{I,\kappa} = i$ is igaz.

Elsőrendű következmény

Azt kell belátni, hogy $\{\forall x(E(x) \rightarrow H(x)), E(s)\} \models H(s)$.

Legyen I tetszőleges interpretáció és κ ebben tetszőleges változókiértékelés és tegyük fel, hogy $|\forall x(E(x) \rightarrow H(x))|^{I,\kappa} = i$ és $|E(s)|^{I,\kappa} = i$.

Előbbi miatt κ^* -nak minden κ^* x -variánsára $|E(x) \rightarrow H(x)|^{I,\kappa^*} = i$. Vegyük ezek közül azt, amelyre $\kappa^*(x) = s^I$. Ekkor $|E(x)|^{I,\kappa^*} = |E(s)|^{I,\kappa} = i$, hiszen mindkettő épp akkor igaz, ha $(s^I) \in E^I$.

Tehát $|H(x)|^{I,\kappa^*} = i$, és így $(s^I) \in H^I$, ami épp azt jelenti, hogy $|H(s)|^{I,\kappa} = i$. \square

Megjegyzés: Léteznek a végtelen a keresési teret szűkítő bizonyítási módszerek (pl. elsőrendű rezolúció), de ezek nem adnak egy minden esetben véges sok lépésben termináló algoritmust.

Függvények aszimptotikus nagyságrendje

Legyenek $f, g : \mathbb{N} \rightarrow \mathbb{R}_0^+$ függvények, ahol \mathbb{N} a természetes számok, \mathbb{R}_0^+ pedig a nemnegatív valós számok halmaza.

- ▶ f -nek g aszimptotikus felső korlátja (jelölése: $f(n) = O(g(n))$); ejtsd: $f(n)$ = nagydó $g(n)$) ha létezik olyan $c > 0$ konstans és $N \in \mathbb{N}$ küszöbindex, hogy $f(n) \leq c \cdot g(n)$ minden $n \geq N$ -re.
- ▶ f -nek g aszimptotikus alsó korlátja (jelölése: $f(n) = \Omega(g(n))$) ha létezik olyan $c > 0$ konstans és $N \in \mathbb{N}$ küszöbindex, hogy $f(n) \geq c \cdot g(n)$ minden $n \geq N$ -re.
- ▶ f -nek g aszimptotikus éles korlátja (jelölése: $f(n) = \Theta(g(n))$) ha léteznek olyan $c_1, c_2 > 0$ konstansok és $N \in \mathbb{N}$ küszöbindex, hogy $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ minden $n \geq N$ -re.

Megjegyzés: a definíció könnyen kiterjeszthető aszimptotikusan nemnegatív, azaz egy korlát után nemnegatív értékű függvényekre. Ilyenek például a pozitív főegyütthatójú polinomok.

Függvények aszimptotikus nagyságrendje

O, Ω, Θ 2-aritású relációnak is tekinthető az $\mathbb{N} \rightarrow \mathbb{R}_0^+$ függvények univerzumán, ekkor

- ▶ O, Ω, Θ tranzitív (pl. $f = O(g), g = O(h) \Rightarrow f = O(h)$)
- ▶ O, Ω, Θ reflexív
- ▶ Θ szimmetrikus
- ▶ O, Ω fordítottan szimmetrikus ($f = O(g) \Leftrightarrow g = \Omega(f)$)
- ▶ (köv.) Θ ekvivalenciareláció, az $\mathbb{N} \rightarrow \mathbb{R}_0^+$ függvények egy osztályozását adja. Az egyes függvényosztályokat általában "legegyszerűbb" tagjukkal reprezentáljuk. Pl. 1 (korlátos függvények), n (lineáris függvények), n^2 (négyzetes függvények), stb. Persze a négyzetes függvények osztálya nem csak másodfokú polinomokat tartalmaz. Pl. $2n^2 + 3 \log_2 n = \Theta(n^2)$.

Függvények aszimptotikus nagyságrendje

- ▶ $f, g = O(h) \Rightarrow f + g = O(h)$, hasonlóan Ω -ra, Θ -ra. (Összeadásra való zártság)
- ▶ Legyen $c > 0$ konstans $f = O(g) \Rightarrow c \cdot f = O(g)$, hasonlóan Ω -ra, Θ -ra. (Pozitív konstanssal szorzásra való zártság)
- ▶ $f + g = \Theta(\max\{f, g\})$ (szekvencia tétele). A domináns tag határozza meg egy összeg aszimptotikus nagyságrendjét.
- ▶ Ha létezik az f/g határérték
 - ha $f(n)/g(n) \rightarrow +\infty \Rightarrow f(n) = \Omega(g(n))$ és $f(n) \neq O(g(n))$
 - ha $f(n)/g(n) \rightarrow c \quad (c > 0) \Rightarrow f(n) = \Theta(g(n))$
 - ha $f(n)/g(n) \rightarrow 0 \Rightarrow f(n) = O(g(n))$ és $f(n) \neq \Omega(g(n))$

Függvények aszimptotikus nagyságrendje

- ▶ $p(n) = a_k n^k + \dots + a_1 n + a_0$ ($a_k > 0$), ekkor $p(n) = \Theta(n^k)$,
- ▶ Minden $p(n)$ polinomra és $c > 1$ konstansra $p(n) = O(c^n)$, de $p(n) \neq \Omega(c^n)$,
- ▶ Minden $c > d > 1$ konstansokra $d^n = O(c^n)$, de $d^n \neq \Omega(c^n)$,
- ▶ Minden $a, b > 1$ -re $\log_a n = \Theta(\log_b n)$,
- ▶ Minden $c > 0$ -ra $\log n = O(n^c)$, de $\log n \neq \Omega(n^c)$.

Megjegyzés:

A jelölés Edmund Landau német matematikustól származik.

Matematikailag precízebb például $f = O(g)$ helyett a következő:

$$O(g) := \{f \mid \exists c > 0 \exists N \in \mathbb{N} \forall n \geq N : f(n) \leq c \cdot g(n)\}.$$

Ilyenkor ha f -nek g aszimptotikus felső korlátja $f \in O(g)$ -t írhatunk.

A számításelmélet alapjai II.

3. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Alapfogalmak, jelölések – Ismétlés

Ábécé: Egy véges, nemüres halmaz.
Az ábécé elemeit **betűk**nek nevezzük.

Egy V ábécé elemeiből képzett véges sorozatokat **V feletti szavaknak** vagy sztringeknek nevezzük. Egy $u = t_1 \cdots t_n$ szóban lévő betűk számát (n) a szó **hosszának** nevezzük. Jelölés: $|u| = n$. A 0 hosszú sorozat jelölése ε , ezt **üres szónak** nevezzük ($|\varepsilon| = 0$). V^* jelöli a **V ábécé feletti szavak halmazát**, beleértve az üres szót is.

$V^+ = V^* \setminus \{\varepsilon\}$ a **V ábécé feletti, nemüres szavak halmazát** jelöli.

Példa: $V = \{a, b\}$, ekkor
 $V^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$.

Alapfogalmak, jelölések – Ismétlés

Legyen V egy ábécé, V^* egy L részhalmazát V feletti **nyelvnek** nevezzük.

Az üres nyelv (nyelv, amely egyetlen szót sem tartalmaz) jelölése \emptyset . Egy V ábécé feletti nyelv véges nyelv, ha véges számú szót tartalmaz, ellenkező esetben végtelen.

Ha X egy halmaz, jelölje $\mathcal{P}(X)$ az X halmaz hatványhalmazát, azaz $\mathcal{P}(X) = \{A \mid A \subseteq X\}$.

Nyelvcsalád (vagy nyelvosztály) alatt nyelveknek egy halmazát értjük.

Tehát ha V egy ábécé:

- $a \in V$: betű
- $u \in V^*$: szó
- $L \subseteq V^*$ vagy $L \in \mathcal{P}(V^*)$: nyelv
- $\mathcal{L} \subseteq \mathcal{P}(V^*)$ vagy $\mathcal{L} \in \mathcal{P}(\mathcal{P}(V^*))$: nyelvcsalád

Grammatikák – Ismétlés

Definíció

Egy $G = \langle N, T, P, S \rangle$ rendezett négyest **grammatikának** nevezünk ha

- ▶ N és T diszjunkt véges ábécék (azaz $N \cap T = \emptyset$). N elemeit **nemterminális**, T elemeit pedig **terminális** szimbólumoknak nevezzük.
- ▶ $S \in N$ a grammatika **kezdőszimbóluma**.
- ▶ A P **szabályrendszer** $x \rightarrow y$ alakú szabályok véges halmaza, ahol $x \in (N \cup T)^* N (N \cup T)^*$, $y \in (N \cup T)^*$.

Példa: A $G = \langle \{S\}, \{a\}, \{S \rightarrow aaS, S \rightarrow \varepsilon\}, S \rangle$ grammatikában $S \Rightarrow aaS \Rightarrow aaaaS \Rightarrow aaaa$ egy **levezetés**. Egy G grammatika kezdőszimbólumából levezethető, csak terminálisokból álló szavak halmazát a grammatika által **generált nyelvnek** nevezzük és $L(G)$ -vel jelöljük. Az adott példában $L(G) = \{a^{2n} \mid n \in \mathbb{N}\}$.

Grammatikák osztályozása – Ismétlés

Legyen $G = \langle N, T, P, S \rangle$ egy grammatika. A G **grammatika i -típusú** ($i = 0, 1, 2, 3$), ha P szabályhalmazára teljesülnek a következők:

- ▶ $i = 0$ eset: nincs korlátozás,
- ▶ $i = 1$ eset:
 - (1) P minden szabálya $u_1 A u_2 \rightarrow u_1 v u_2$ alakú, ahol $u_1, u_2, v \in (N \cup T)^*$, $A \in N$, és $v \neq \varepsilon$,
 - (2) Egyetlen kivétel megengedünk: P tartalmazhatja az $S \rightarrow \varepsilon$ szabályt, de csak abban az esetben, ha S nem fordul elő P egyetlen szabályának jobb oldalán sem.
("Korlátozott ε szabály" vagy röviden "KES")
- ▶ $i = 2$ eset: P minden szabálya $A \rightarrow v$ alakú, ahol $A \in N$ és $v \in (N \cup T)^*$,
- ▶ $i = 3$ eset: P minden szabálya vagy $A \rightarrow uB$ vagy $A \rightarrow u$, alakú, ahol $A, B \in N$ és $u \in T^*$.

Nyelvek Chomsky féle osztályozása – Ismétlés

$\mathcal{L}_i := \{L \mid \exists G \in \mathcal{G}_i, \text{ hogy } L = L(G)\}$ jelöli az i -típusú nyelvek nyelvosztályát, elemei az **i -típusú nyelvek**. ($i = 0, 1, 2, 3$).

A 0,1,2,3-típusú grammatikákat rendre **mondatszerkezetű**, **környezetfüggő**, **környezetfüggetlen**, valamint **reguláris** grammatikának is mondjuk.

A 0,1,2,3-típusú nyelvek osztályait rendre **rekurzíven felsorolható**, **környezetfüggő**, **környezetfüggetlen**, valamint **reguláris** nyelvosztálynak is mondjuk.

Chomsky nyelvhierarchia tétele

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0.$$

Emlékeztető: Az \mathcal{L}_2 és az \mathcal{L}_1 nyelvosztályok közötti tartalmazási reláció nem adódik azonnal a grammatikaosztályok definíciójából. Tartalmazás valódisága: pumpálási (Bar Hillel) lemmákkal.

Formális nyelvek megadása

- (Generatív) grammatikákkal

A grammatikák **szintetizáló** eszközök, egyetlen szimbólumból egy szabályrendszer segítségével szavakat lehet felépíteni. Azon szavak halmaza, melyeket fel lehet építeni egy nyelvet határoz meg, tehát a grammatika szabályrendszere meghatároz egy nyelvet.

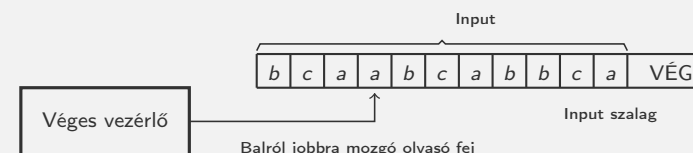
- Matematikai gépek, automaták segítségével

Az automaták elemző, **analitikus** eszközök. Az automaták bemenete egy szó, kimenete egy bináris érték („igen”/„nem”). Az automata működési szabályai szerint feldolgozza a szavakat. Csak bizonyos szavak esetén ad „igen” választ, ezen szavak egy nyelvet alkotnak, melyet az automata működési szabályai határoz meg.

- egyéb módon: felsorolás, reguláris kifejezés, ...

Figyelem! Nem feltétlen lehet minden eszközzel minden nyelvet megadni. Például reguláris kifejezéssel kevesebb nyelvet lehet leírni mint egy általános grammatikával, de az se elég az összes $\{0, 1\}$ ábécé feletti nyelv leírásához.

Véges automata, intuitív kép – Ismétlés



- ▶ A véges automata a kezdőállapotából indul, az inputszó az inputszalagon helyezkedik el, az olvasófej pedig az inputszó legbaloldali szimbólumán áll.
- ▶ Az automata, miután elolvasott egy szimbólumot, az olvasófejet egy pozícióval jobbra mozgatja, majd állapotot vált az állapot-átmenet függvénye szerint.
- ▶ Amennyiben az automata még nem olvasta végig a teljes inputot és elfogadó állapotba ér akkor nem dönt még az elfogadásról/elutasításról, tovább működik. Ha végigolvasta az inputot, akkor megáll és aktuális állapota alapján válaszol, hogy elfogadja vagy elutasítja-e a bemenetet.

Véges automata – Ismétlés

Definíció

A (nemdeterminisztikus) **véges automata** egy rendezett ötös, $A = \langle Q, T, \delta, Q_0, F \rangle$, ahol

- ▶ Q az állapotok egy véges, nemüres halmaza,
- ▶ T az inputszimbólumok véges ábécéje,
- ▶ $\delta : Q \times T \rightarrow \mathcal{P}(Q)$ ún. állapot-átmenet függvény,
- ▶ $Q_0 \subseteq Q$ a kezdőállapotok halmaza,
- ▶ $F \subseteq Q$ az elfogadó állapotok halmaza.

Definíció

Ha $\forall (q, a) \in Q \times T$ esetén $|\delta(q, a)| = 1$ és $|Q_0| = 1$, akkor **determinisztikus véges automatáról** beszélünk.

Az $A = \langle Q, T, \delta, Q_0, F \rangle$ véges automata által **elfogadott nyelv** T^* azon elemei, amelyekre a szó teljes feldolgozása után F -beli állapotba juthatunk. VDA esetén ez egyértelmű.

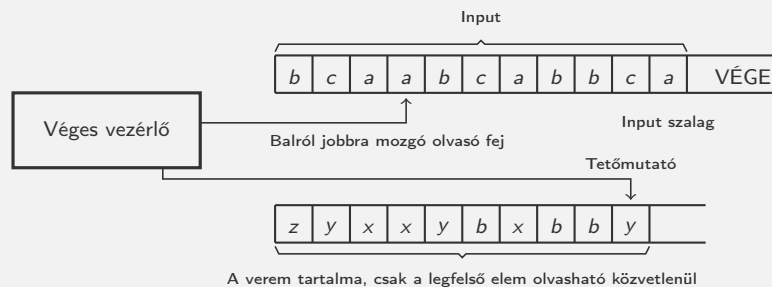
3-as típusú nyelvek – Ismétlés

Tétel

Az alábbi formális eszközök mindegyikével pontosan a 3-as típusú nyelvek írhatók le

- ▶ 3-as típusú grammatikák
- ▶ 3-as normálformájú grammatikák
- ▶ reguláris kifejezések
- ▶ determinisztikus véges automaták (az automata átmenetfüggvénye minden állapot-betű párhoz egyértelműen rendel új állapotot)
- ▶ nemdeterminisztikus véges automaták (az automata átmenetfüggvénye nem feltétlenül egyértelműen rendel az egyes állapot-betű párokhoz új állapotot, több kezdőállapot is megengedett)

Veremautomata – Ismétlés



- ▶ A veremautomata a véges automata általánosítása potenciálisan végtelen veremmel és véges kontrollal.
- ▶ A verem esetében az új adat mindig a már meglévő veremtartalom tetejéhez adódik, kivétele fordított sorrendben történik.
- ▶ alapértelmezetten nemdeterminisztikus

Veremautomata – Ismétlés

Jelölés: ha X egy halmaz, jelölje $\mathcal{P}_{\text{véges}}(X)$ az X véges részhalmazainak halmazát.

Definíció

A **veremautomata** egy $A = \langle Z, Q, T, \delta, z_0, q_0, F \rangle$, rendezett hetes, ahol

- ▶ Z a veremszimbólumok véges halmaza (veremábécé),
- ▶ Q az állapotok véges halmaza,
- ▶ T az inputszimbólumok véges halmaza (inputábécé),
- ▶ $\delta : Z \times Q \times (T \cup \{\varepsilon\}) \rightarrow \mathcal{P}_{\text{véges}}(Z^* \times Q)$, az ún. átmeneti függvény,
- ▶ $z_0 \in Z$ a kezdeti (kezdő) veremszimbólum,
- ▶ $q_0 \in Q$ a kezdeti állapot (kezdőállapot),
- ▶ $F \subseteq Q$ az elfogadó állapotok vagy végállapotok halmaza.

Veremautomata – Ismétlés

- ▶ A verem tetején lévő szimbólum, az aktuális állapot és az inputszimbólum együttesen határozzák meg a következő átmenetet.
- ▶ Minden lépésben mindenképpen kiveszünk egyetlen egy elemet a verem tetejéről és beteszünk helyette néhányat. $(0, 1, 2, \dots$ darabot)
- ▶ Ha $\delta(z, q, \varepsilon)$ nem üres, akkor ún. **ε -átmenet** (ε -lépés, ε -mozgás) hajtható végre, ami lehetővé teszi, hogy a veremautomata anélkül változtassa meg az állapotát, hogy valamilyen szimbólumot olvasson az inputszalagról.
- ▶ ε -mozgásra lehetőség van már az első inputszimbólum elolvasása előtt is illetve még az utolsó inputszimbólum elolvasása után is.

Veremautomata – Ismétlés

Az $A = \langle Z, Q, T, \delta, z_0, q_0, F \rangle$ veremautomata által **elfogadó állapottal (végállapottal) elfogadott nyelv** azon T feletti w szavakból áll, amelyeknek a teljes feldolgozása (végigolvasása) után a veremautomata el tud jutni valamely végállapotába.

(A működés nemdeterminisztikus, a feldolgás végén a verem tartalma nem érdekes.) Az A által végállapottal elfogadott nyelvet $L(A)$ jelöli.

Az $A = \langle Z, Q, T, \delta, z_0, q_0, F \rangle$ veremautomata által **üres veremmel elfogadott nyelv** azon T feletti w szavakból áll, amelyeknek a teljes feldolgozása (végigolvasása) után a veremautomata olyan konfigurációba tud eljutni, ahol a verem üres.

(A működés nemdeterminisztikus, a feldolgás végén az aktuális állapot nem érdekes.) Az A által üres veremmel elfogadott nyelvet $N(A)$ jelöli.

Veremautomata – Ismétlés

Determinisztikus veremautomata

Definíció

Az $A = \langle Z, Q, T, \delta, z_0, q_0, F \rangle$ veremautomatát **determinisztikusnak** nevezzük, ha minden $(z, q, a) \in Z \times Q \times T$ esetén $|\delta(z, q, a)| + |\delta(z, q, \varepsilon)| = 1$.

Tehát minden $q \in Q$ és $z \in Z$ esetén

- ▶ vagy $\delta(z, q, a)$ pontosan egy elemet tartalmaz minden $a \in T$ inputszimbólumra és $\delta(z, q, \varepsilon) = \emptyset$,
- ▶ vagy $\delta(z, q, \varepsilon)$ pontosan egy elemet tartalmaz és $\delta(z, q, a) = \emptyset$ minden $a \in T$ inputszimbólumra.

Észrevétel: Ha minden $(z, q, a) \in Z \times Q \times T$ esetén $|\delta(z, q, a)| + |\delta(z, q, \varepsilon)| \leq 1$ akkor a veremautomata a felismert nyelv módosulása nélkül kiegészíthető determinisztikus veremautomatává. Így tágabb értelemben az ezt a feltételt teljesítő veremautomatákat is tekinthetjük determinisztikus veremautomatának.

Veremautomata – Ismétlés

Veremautomaták és a 2-es típusú nyelvek

Tétel

Bármely L nyelvre ekvivalensek a következő állítások

- ▶ L környezetfüggetlen, azaz környezetfüggetlen (2-es típusú) grammatikával generálható
- ▶ L (nemdeterminisztikus) veremautomatával végállapottal felismerhető
- ▶ L (nemdeterminisztikus) veremautomatával üres veremmel felismerhető

Tétel

Minden reguláris (3-as típusú) nyelv felismerhető determinisztikus veremautomatával, de létezik olyan (2-es típusú) környezetfüggetlen nyelv, ami nem ismerhető fel determinisztikus veremautomatával.

Néhány nyelv szigorú típusa – Ismétlés

Jelölés: $|u|_t$: a t betűk száma u -ban

Példák:

$\in \mathcal{L}_3$	$\in \mathcal{L}_2 - \mathcal{L}_3$	$\in \mathcal{L}_1 - \mathcal{L}_2$
$\{u \mid abbab \subseteq u\}$	$\{u \in \{a, b\}^* \mid u = u^{-1}\}$	$\{uu \mid u \in \{a, b\}^*\}$
$\{u \mid abbab \not\subseteq u\}$	$\{a^n b^n \mid n \in \mathbb{N}\}$	$\{a^n b^n c^n \mid n \in \mathbb{N}\}$
7-tel osztható számok	$\{u \in \{a, b\}^* \mid u _a = u _b\}$	$\{a^{n^2} \mid n \in \mathbb{N}\}$
$((a + bb)^* + ab)^*$	helyes ()-k nyelve	$\{a^{2^n} \mid n \in \mathbb{N}\}$

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.

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.
- ▶ $\mathcal{I} = \{G \mid G \text{ környezetfüggetlen grammatika}\} \times T^*$, $\mathcal{O} = \{„igen”, „nem”\}$, ahol T egy ábécé. \mathcal{P} a szóprobléma. Algoritmikus megoldás: CYK algoritmus
- ▶ $\mathcal{I} = \{\text{elsőrendű formulák}\}$, $\mathcal{O} = \{„igen”, „nem”\}$, $\mathcal{P}(\varphi) = „igen”$ 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.)

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

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 imidig 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 veremautomatával 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 nyelveleíró 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 biztató 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.

Már a 30-as években megfogalmazásra került a következő:

Church-Turing tézis

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

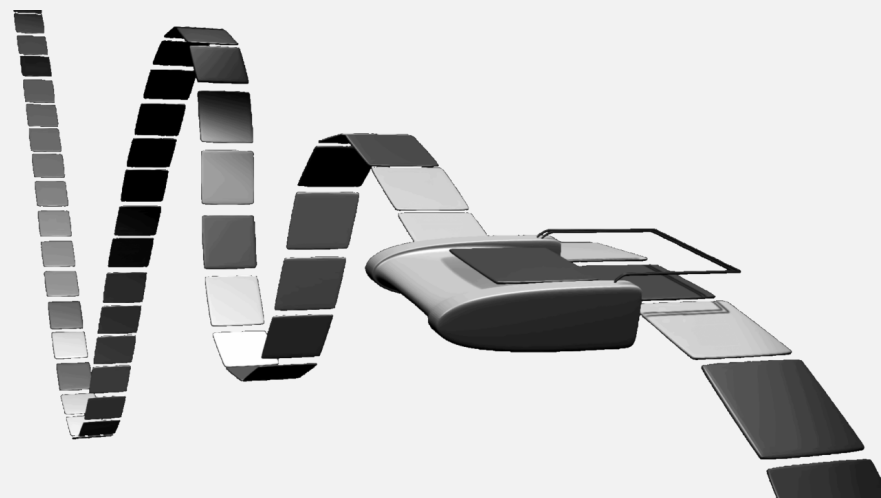
(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

NEM TÉTEL!!!

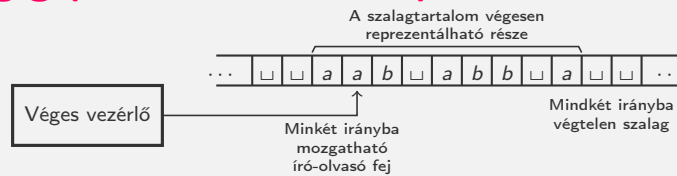
A Church-Turing tézis nem bizonyítható, hiszen nem egy formális matematikai állítás, az algoritmus intuitív fogalmát használja.

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

Turing gépek

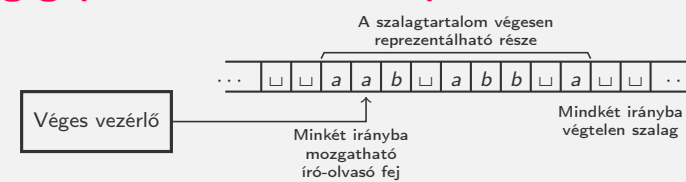


Turing gépek – Informális kép



- ▶ 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



- ▶ 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, 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 $\sqcup \in \Gamma \setminus \Sigma$.
- ▶ $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény. δ az egész $(Q \setminus \{q_i, q_n\}) \times \Gamma$ -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 konfigurációi

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 konfigurációi

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 konfigurációátmenet

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 $\exists n > 0 \wedge C_1, C_2, \dots, C_n \in C_M$, hogy $\forall 1 \leq i \leq n-1$ -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

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

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.

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 *félíg 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:

Definíció

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

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

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 ($t \geq 0$), 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

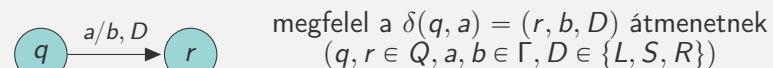
Legyen $f : \mathbb{N} \rightarrow \mathbb{N}$ egy függvény. Azt mondjuk, hogy M egy **$f(n)$ időkorlátos** gép (vagy M $f(n)$ időigényű), ha minden $u \in \Sigma^*$ input szóra M futási ideje az u szón legfeljebb $f(|u|)$.

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

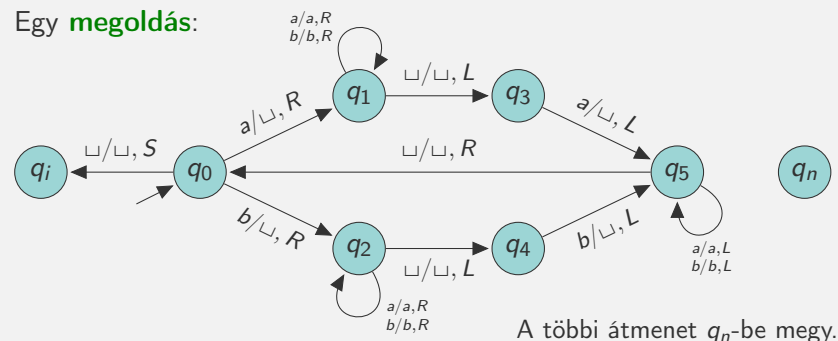
Turing gépek – Példa

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

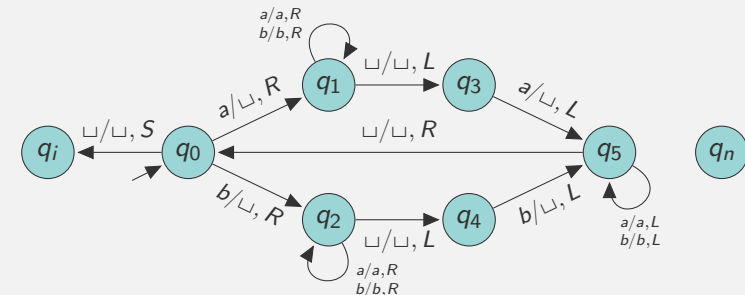
Az **átmenetdiagram**.



Egy **megoldás**:



Turing gépek – Példa (folyt.)

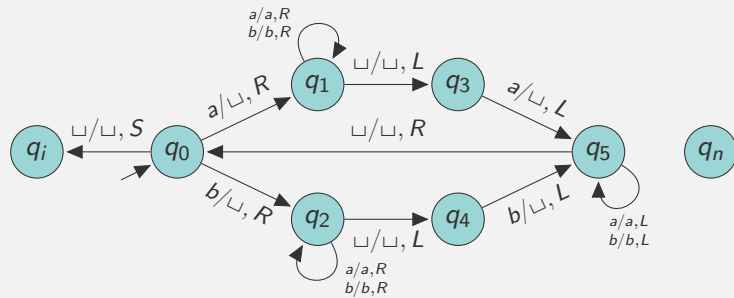


Példa. Konfigurációátmenetek sorozata az aba inputra:

$q_0aba \vdash q_1ba \vdash bq_1a \vdash baq_1 \vdash bq_3a \vdash q_5b \vdash q_5 \vdash q_0b \vdash q_2 \vdash q_4 \vdash q_n$.

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.

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**.

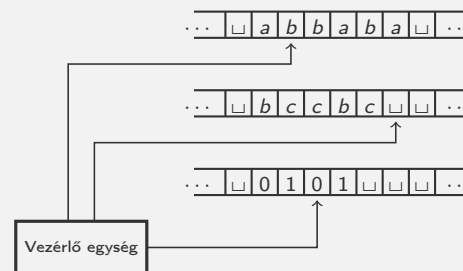
Van-e olyan TG, ami nem dönti el, de azért felismeri L -et? **Igen**, a q_n -be menő átmeneteket vezessük végtelen ciklusba.

A számításelmélet alapjai II.

4. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Többszalagos Turing gép – Informális kép



- ▶ Véges vezérlő egység, $k (\geq 1)$ 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 $k \geq 1$ 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 $\sqcup \in \Gamma \setminus \Sigma$,
- ▶ $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, S, R\}^k$ az átmenet függvény.

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

k -szalagos Turing gépek konfigurációi

Definíció

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

Ez azt reprezentálja, hogy

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

Definíció

Az u szóhoz tartozó **kezdőkonfiguráció**: $(q_0, u_1, v_1, \dots, u_k, v_k)$, ahol $u_i = \varepsilon$ ($1 \leq i \leq k$), $v_1 = u \sqcup$, és $v_i = \sqcup$ ($2 \leq i \leq k$).

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

- ▶ **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.

Legyen $C = (q, u_1, a_1 v_1, \dots, u_k, a_k v_k)$ egy konfiguráció, ahol $a_i \in \Gamma$, $u_i, v_i \in \Gamma^*$ ($1 \leq i \leq k$). Legyen továbbá

$\delta(q, a_1, \dots, a_k) = (r, b_1, \dots, b_k, D_1, \dots, D_k)$, ahol $q, r \in Q$, $b_i \in \Gamma$, $D_i \in \{L, S, R\}$ ($1 \leq i \leq k$). Ekkor

$C \vdash (r, u'_1, v'_1, \dots, u'_k, v'_k)$, ahol minden $1 \leq i \leq k$ -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.

Példa:

Legyen $k=2$ és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1 v_1, u_2, a_2 v_2) \vdash (r, u_1 b_1, v'_1, u_2, b_2 v_2)$, 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**: $L(M) = \{u \in \Sigma^* \mid (q_0, \varepsilon, u \sqcup, \varepsilon, \sqcup, \dots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \dots, x_k, y_k), \text{ valamely } x_1, y_1, \dots, x_k, y_k \in \Gamma^*, y_1, \dots, y_k \neq \varepsilon\}$.

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.

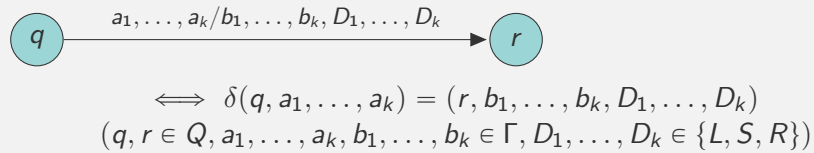
Definíció

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

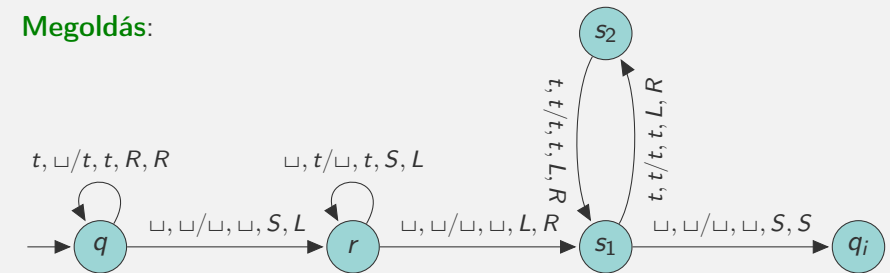
A k -szalagos TG-ek **átmenetdiagramja** egy csúcs- és élcímkezett irányított gráf, melyre



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.

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

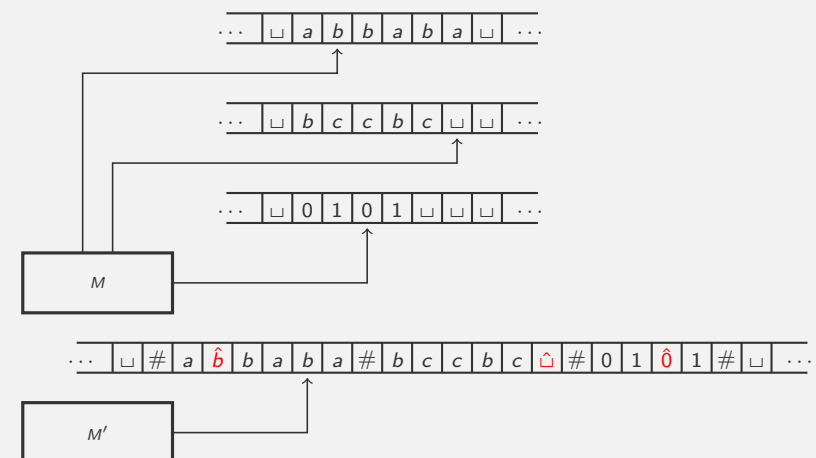
Két TG **ekvivalens**, ha ugyanazt a nyelvet ismerik fel.

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 $q'_0 \# \hat{a}_1 a_2 \cdots a_n \# \hat{\sqcup} \# \cdots \hat{\sqcup} \#$
2. M' először végigmegy a szalagon (számolja a $\#$ -okat) és eltárolja a $\hat{\cdot}$ -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' még egyszer 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 $O(\text{mozgató betűk száma})$ 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

- ▶ 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 $O(\text{felhasznált cellaterület})$ lépésben megoldható.)
- ▶ a felhasznált cellaterület $O(1)$ -el nőtt. ($\leq k$ -val, hiszen $\leq k$ -szor kellhet egy \sqcup -t beszúrni.)

Az M' által felhasznált cellaterület mérete kezdetben $\Theta(n)$, lépésenként $O(1)$ -gyel nőhet, így $\leq f(n)$ 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 $\leq f(n)$ 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.

- ▶ 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.

Nilvánvalóan 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.

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 tükröképe van)
2. Szimuláljuk M' -t egy egyirányban végtelen szalagos M'' Turing géppel (az előző tételben látott bizonyításhoz hasonlóan)

Nemdeterminisztikus Turing gép

Jelölés: $\mathcal{P}(X) = \{Y \mid Y \subseteq X\}$ az X halmaz hatványhalmaza.

Nemdeterminisztikus Turing gép (NTG)

Az egyszalagos **nemdeterminisztikus Turing gép** (továbbiakban röviden NTG) 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 $\sqcup \in \Gamma \setminus \Sigma$,
- ▶ $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, S, R\})$.

Azaz míg a **determinisztikus** esetben a δ átmenetfüggvény minden egyes $(Q \setminus \{q_i, q_n\}) \times \Gamma$ -beli párhoz **pontosan egy**, addig egy **nemdeterminisztikus** TG **akárhány** (pl. 0,1,5,100) darab $Q \times \Gamma \times \{L, S, R\}$ -beli rendezett hármast rendelhet hozzá.

NTG egy lépéses konfigurációátmenete

A **konfiguráció** fogalma azonos, jelölje most is C_M az M NTG lehetséges konfigurációinak halmazát.

Definíció

Egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egyszalagos nemdeterminisztikus 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 $(r, b, R) \in \delta(q, a)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$, különben $v' = \sqcup$,
- ▶ ha $(r, b, S) \in \delta(q, a)$, akkor $uqav \vdash urbv$,
- ▶ ha $(r, b, L) \in \delta(q, a)$, 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 $\delta(q_2, a) = \{(q_5, b, L), (q_1, d, R)\}$ Legyen továbbá $C_1 = bcq_2a \sqcup b$, $C_2 = bq_5cb \sqcup b$, $C_3 = bcdq_1 \sqcup b$. Ekkor $C_1 \vdash C_2$ és $C_1 \vdash C_3$.

Nemdeterminisztikus Turing gép

Vegyük észre, hogy míg a **determinisztikus** esetben minden nem-megállási C konfigurációhoz **pontosan egy** C' konfiguráció létezett, melyre $C \vdash C'$, addig a **nemdeterminisztikus** esetben **több** ilyen is létezhet. Pl. 0,1,5,100 darab. Persze csak véges sok, hiszen $|Q \times \Gamma \times \{L, S, R\}|$ véges!

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

Példa: Tegyük fel, hogy $C_1 \vdash C_2$, $C_1 \vdash C_3$, $C_2 \vdash C_4$. Ekkor $C_1 \vdash^* C_1$, $C_1 \vdash^* C_2$, $C_1 \vdash^* C_3$ és $C_1 \vdash^* C_4$ is teljesül.

Nemdeterminisztikus Turing gép

Definíció

Az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ nemdeterminisztikus Turing gép által **felismert nyelv**

$L(M) = \{u \in \Sigma^* \mid q_0u \sqcup \vdash^* xq_iy \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon\}$.

Bár a definíció formálisan megegyezik a determinisztikus TG által felismert nyelv definíciójával az egy lépéses átmenet fogalmának módosulása miatt újra érdemes átgondolni mit jelent ez.

Determinisztikus esetben csupán egyetlen számítása létezik a gépnek adott kezdőkonfigurációból, így ha elfogadó konfigurációba jut, akkor nincs elutasító konfigurációba jutó számítása és viszont.

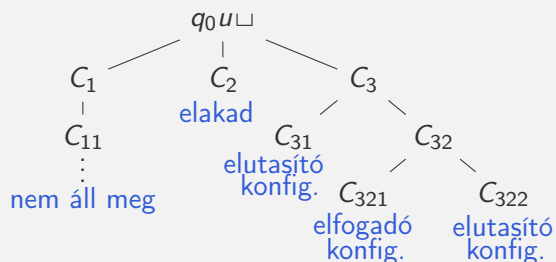
Egy NTG-nek azonban **több számítása is lehet ugyanarra a szóra**. Ezek között lehetnek elfogadó és elutasító (sőt nem termináló!) számítások is. Egy NTG akkor fogad el egy szót, ha az adott szóra **legalább egy számítása q_i -ben ér véget** (hiszen ekkor a kezdőkonfiguráció és ez az elfogadó konfiguráció \vdash^* relációban áll).

Nemdeterminisztikus számítási fa

Definíció

Egy M TG egy $u \in \Sigma^*$ inputjához tartozó **nemdeterminisztikus számítási fa** egy gyökeres fa, melynek csúcsai M konfigurációival címkézettek. $q_0 u \sqcup$ a gyökér címkéje. Ha C egy csúcs címkéje, akkor $|\{C' \mid C \vdash C'\}|$ gyereke van és ezek címkéi éppen $\{C' \mid C \vdash C'\}$ elemei.

Példa:



M elfogadja u -t, hiszen $q_0 u \sqcup \vdash C_3 \vdash C_{32} \vdash C_{321}$ elfogadó számítás. Az elfogadáshoz **egyetlen** elfogadó számítás is elég!

Nemdeterminisztikus számítási fa

Tehát adott inputra több számítás is lehetséges, ezek lehetnek elfogadóak, elutasítóak, **elakadóak** (ha olyan C -be jut, melyre $\{C' \mid C \vdash C'\} = \emptyset$), illetve végtelenek.

Észrevétel: $u \in L(M) \Leftrightarrow$ az u -hoz tartozó nemdeterminisztikus számítási fának van olyan levele, ami elfogadó konfiguráció.

Megjegyzés: a nemdeterminisztikus Turing gép definíciója értelemszerűen kiterjeszthető k -szalagos gépekre is, így beszélhetünk k -szalagos nemdeterminisztikus Turing gépekről is.

NTG-vel való eldönthetőség, időigény

Definíció

Az M NTG **felismeri** az $L \subseteq \Sigma^*$ nyelvet, ha $L(M) = L$.

Az M NTG **eldönti** az $L \subseteq \Sigma^*$ nyelvet, ha felismeri továbbá minden $u \in \Sigma^*$ input szóhoz tartozó nemdeterminisztikus számítási fa véges és a fa minden levele elfogadó vagy elutasító konfiguráció.

Definíció

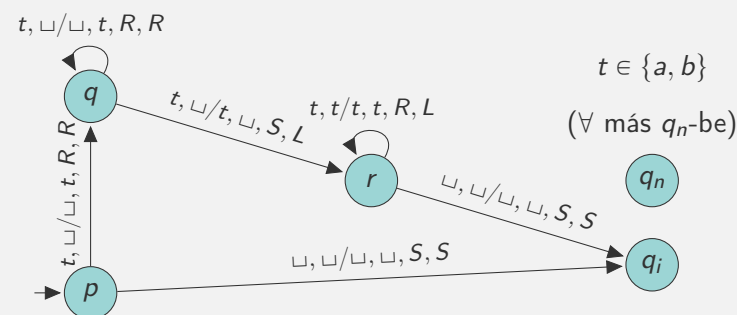
Az M NTG **$f(n)$ időkorlátos** (időigényű), ha minden $u \in \Sigma^*$ n hosszú szóra u számítási fája legfeljebb $f(n)$ magas.

Tehát, ha M $f(n)$ időkorlátos, akkor nincs végtelen számítása és minden n -re a legfeljebb n méretű bemeneteken a számításai (nemcsak az elfogadó, hanem az elutasító és elakadó számításai is) legfeljebb $f(n)$ lépésben véget érnek.

Nemdeterminisztikus Turing gép

Példa

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



$(p, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, \varepsilon, bba, a, \sqcup) \vdash (r, \varepsilon, bba, \varepsilon, a) \vdash (q_n, \varepsilon, bba, \varepsilon, a)$

$(p, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, \varepsilon, bba, a, \sqcup) \vdash (q, \varepsilon, ba, ab, \sqcup) \vdash (r, \varepsilon, ba, a, b) \vdash (r, b, a, \varepsilon, ab) \vdash (r, ba, \sqcup, \varepsilon, \sqcup ab) \vdash (q_i, ba, \sqcup, \varepsilon, \sqcup ab)$

Hosszlexikografikus rendezés

Definíció

Legyen $X = \{x_1 < x_2 < \dots < x_s\}$ egy rendezett ábécé. Ekkor X^* szavainak **hossz-lexikografikus** (shortlex) rendezése alatt azt a $<_{\text{shortlex}}$ rendezést értjük, melyre a következők teljesülnek. Minden $u_1 \dots u_n, v_1 \dots v_m \in X^*$ -ra
 $u_1 \dots u_n <_{\text{shortlex}} v_1 \dots v_m \Leftrightarrow (n < m) \vee ((n = m) \wedge (u_k < v_k))$, ahol k a legkisebb olyan i , melyre $u_i \neq v_i$.

1. Példa: Ha $X = \{a, b\}$ és $a < b$, akkor X^* szavainak hossz-lexikografikus sorrendje:

$\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \dots$

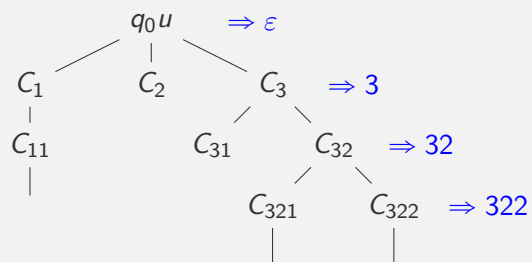
2. Példa: Tekintsük a természetes számokat (azaz 0 számjeggyel nem kezdődhetnek, a 0 kivételével), mint számjegysorozatokat.

Ekkor $n < m$ pontosan akkor igaz, ha az

$X = \{0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9\}$ rendezett ábécé feletti szavaknak tekintve őket $n <_{\text{shortlex}} m$ teljesül.

NTG szimulálása determinisztikus TG-pel

A számítási fa minden csúcsához egyértelműen hozzárendelhető egy T^* -beli szó, az adott konfigurációhoz tartozó parciális számítás (konfigurációátmenet-sorozat) ún. *szelektora*.



A gyökér szelektora ε . Tekintsük a gyökértől egy x csúcsig vezető egyértelmű utat, ha a szülő konfigurációnak x az i -edik gyereke és a szülő szelektora $w \in T^*$, akkor x szelektora $wi \in T^*$.

Nemdeterminisztikus Turing gép

Szimulálás determinisztikus TG-pel

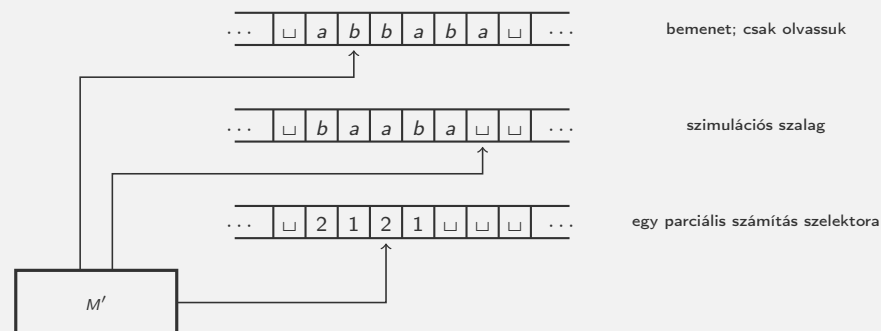
Tétel

Minden $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ $f(n)$ időkorlátos NTG-hez megadható egy ekvivalens, $2^{O(f(n))}$ időkorlátos M' determinisztikus TG.

Bizonyítás (vázlat): Ötlet: Vegyük észre, hogy minden $u \in \Sigma^*$ -ra u számítási fájának csúcsai éppen u parciális (nem feltétlen befejezett) számításainak felelnek meg. M' egy adott $u \in \Sigma^*$ bemeneten tehát szimulálni tudja u M -beli összes parciális számítását a számítási fájának szélességi bejárása által.

- Legyen d az M átmenetfüggvényének jobb oldalán szereplő halmazok számosságának a maximuma, azaz $d = \max_{(q,a) \in Q \times T} |\delta(q,a)|$.
- Legyen $T = \{1, 2, \dots, d\}$ egy (rendezett) ábécé.
- minden $(q, a) \in Q \times \Gamma$ esetén rögzítsük le $\delta(q, a)$ elemeinek egy sorrendjét

NTG szimulálása determinisztikus TG-pel



M' működése:

NTG szimulálása determinisztikus TG-pel

- ▶ M' kezdőkonfigurációja: az 1-es szalag tartalmazza a bemenetet, a 2-es és 3-as szalagok üresek.
- ▶ Amíg nincs elfogadás
 - M' rámásolja az 1-es szalag tartalmát a 2-esre
 - Amíg a 3-ik szalagon a fej nem \sqcup -re mutat
 - Legyen k a 3-ik szalagon a fej pozíciójában lévő betű
 - Legyen a 2-ik szalagon a fej pozíciójában lévő betű a és a szimulált M aktuális állapota q
 - Ha $\delta(q, a)$ -nak \exists k -adik eleme, akkor
 - M' szimulálja M egy lépését ezen elem szerint
 - Ha ez q_i -be vezet, akkor M' is elfogad
 - Ha ez q_n -be vezet, akkor M' kilép ebből a ciklusból különben (ha $\delta(q, a)$ -nak \nexists k -adik eleme) kilép a ciklusból
 - M' a 3-ik szalagon eggyel jobbra lép
 - M' törli a 2. szalagot és előállítja a 3. szalagon a hossz-lexikografikus (shortlex) rendezés szerinti következő szót T felett (a fejet a szó elejére állítva)

NTG szimulálása determinisztikus TG-pel

- ▶ M' akkor és csak akkor lép elfogadó állapotba, ha a szimulált M elfogadó állapotba lép, azaz a két gép ekvivalens
- ▶ M' -nek $f(n)$ -ben exponenciálisan sok számítást kell megvizsgálnia, legfeljebb annyit, mint amennyi egy $f(n)$ magasságú teljes d -áris fa csúcsainak száma. Ez

$$\sum_{i=0}^{f(n)} d^i = \frac{d^{f(n)+1} - 1}{d - 1} = O(d^{f(n)}).$$

- ▶ a parciális számítások szimulálása $O(n + f(n))$ időkorlátos,
- ▶ így M' $O(n + f(n))O(d^{f(n)}) = 2^{O(f(n))}$ időkorlátos.

Megjegyzés:

- ▶ Abból, hogy a bizonyításban alkalmazott szimuláció exponenciális időigényű még nem következik, hogy nincs hatékonyabb szimuláció.
- ▶ Az a sejtés, hogy nem lehet NTG-t az időigény drasztikus romlása nélkül determinisztikus TG-pel szimulálni.

A számításelmélet alapjai II.

5. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Számosság

A véges halmazok fontos jellemzője a méretük (\rightarrow **természetes számok** fogalma). Cél: ennek kiterjesztése végtelen halmazokra. Ez vezetett a **számosság** fogalmához (*G. Cantor, 1845-1918*).

Definíció

- ▶ A és B halmazoknak **megegyezik a számosságuk**, ha \exists bijekció köztük. Jelölése: $|A| = |B|$.
- ▶ A -nak **legalább annyi a számossága**, mint B -nek, ha \exists B -ből injekció A -ba. Jelölése: $|A| \geq |B|$.
- ▶ A -nak **nagyobb a számossága, mint B -nek**, ha \exists B -ből A -ba injekció, de \nexists bijekció. Jelölése: $|A| > |B|$.

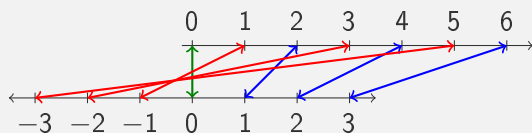
Cantor-Bernstein-Schröder tétel

Ha \exists injekció A -ból B -be és B -ből A -ba is, akkor \exists bijekció A és B között, azaz ha $|A| \leq |B|$ és $|A| \geq |B|$, akkor $|A| = |B|$.

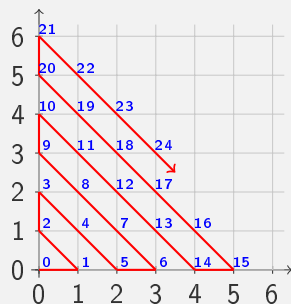
(Nem nyilvánvaló, itt nem bizonyítjuk.)

Számosság – példák

1. Példa: $|\mathbb{N}| = |\mathbb{Z}|$.



2. példa: $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$.



A megszámlálhatóan végtelen számosság

3. példa: $|\mathbb{N}| = |\mathbb{Q}|$.

Bizonyítás:

$\mathbb{N} \subset \mathbb{Q}$, ezért $|\mathbb{N}| \leq |\mathbb{Q}|$.

$\mathbb{Q}^+ := \{\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+, \text{ a tört nem egyszerűsíthető}\}$.

$\mathbb{Q}^- := \{-\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+, \text{ a tört nem egyszerűsíthető}\}$.

$\frac{p}{q} \in \mathbb{Q}^+ \mapsto (p, q) \in \mathbb{N} \times \mathbb{N}$ injektív, tehát $|\mathbb{Q}^+| \leq |\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$.

Legyen $\mathbb{Q}^+ = \{a_0, a_1, \dots\}$, $\mathbb{Q}^- = \{b_0, b_1, \dots\}$, ekkor

$\mathbb{Q} = \{0, a_0, b_0, a_1, b_1, \dots\}$.

Definíció

Egy A halmaz **megszámlálhatóan végtelen számosságú**, ha létezik A és \mathbb{N} között bijekció.

Azaz egy A halmaz számossága megszámlálhatóan végtelen, ha elemei megindexelhetők a természetes számokkal (vagy \mathbb{N}^+ elemeivel; az indexek eggyel való eltolása bijekció \mathbb{N}^+ és \mathbb{N} között).

A continuum számosság

Egy halmaz **megszámlálható**, ha számossága véges vagy megszámlálhatóan végtelen.

Tétel: Megszámlálható sok megszámlálható halmaz uniója megszámlálható.

Bizonyítás (vázlat) Konstrukció: mint $|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$ bizonyításánál.

Definíció

Egy A halmaz **continuum számosságú**, ha létezik A és \mathbb{R} között bijekció.

Be fogjuk látni, hogy $|\mathbb{R}| > |\mathbb{N}|$.

4. példa: $|\mathbb{R}| = |(0, 1)|$.

Bizonyítás: $\text{tg}(\pi(x - \frac{1}{2}))|_{(0,1)} : (0, 1) \rightarrow \mathbb{R}$ bijekció $(0, 1)$ és \mathbb{R} között.

Megjegyzés: $|\mathbb{R}| = |(a, b)| = |[c, d]|$ és $|\mathbb{R}| = |\mathbb{R}^n|$.

Szavakkal kapcsolatos halmazok számossága

5. Példa: $|\{0, 1\}^*| = |\mathbb{N}|$.

A hossz-lexikografikus (shortlex) rendezés egy bijekciót ad:
 $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \dots$

Jelöljük a megszámlálhatóan ∞ hosszúságú $\{0, 1\}$ -sorozatok halmazát $\{0, 1\}^{\mathbb{N}}$ -nel, azaz

$$\{0, 1\}^{\mathbb{N}} := \{(b_1, \dots, b_i, \dots) \mid b_i \in \{0, 1\}, i \in \mathbb{N}\}.$$

6. Példa: $|\{L \mid L \subseteq \{0, 1\}^*\}| = |\{0, 1\}^{\mathbb{N}}|$

Bizonyítás: Jelölje w_i $\{0, 1\}^*$ hossz-lexikografikus rendezésének i . szavát ($i \in \mathbb{N}$).

Egy L nyelvhez rendeljük hozzá azt a megszámlálhatóan végtelen hosszúságú $\mathbf{b}_L = (b_1, \dots, b_i, \dots)$ bitsorozatot, amelyre $b_i = 1 \Leftrightarrow w_i \in L$.

Ez nyilván bijekció, \mathbf{b}_L -t nevezhetjük is az L nyelv **karakterisztikus sorozatának**.

Szavakkal kapcsolatos halmazok számossága

7. Példa: $|\{0, 1\}^{\mathbb{N}}| = |[0, 1]|$.

Bizonyítás (vázlat):

Minden $x \in [0, 1]$ -hez rendeljük hozzá x kettedestört alakjának "0." utáni részét. Ez nem feltétlen egyértelmű, hiszen a véges kettedestörteknek két végtelen kettedestört alakja is van. (Például $0,01=0,0100\dots=0,0011\dots$)

Válasszuk ilyenkor a ∞ 0-ra végződő alakot. Ez a leképezés így nem bijekció, de injektív, azaz $|[0, 1]| \leq |\{0, 1\}^{\mathbb{N}}|$.

Fordítva, $z \in \{0, 1\}^{\mathbb{N}}$ minden 1-esét helyettesítsük 2-essel, írjunk elé "0."-t és tekintsük végtelen harmadostörtnak. Meggondolható, hogy csak 0-ásokat és 2-eseket tartalmazó harmadostört alakja egy valós számnak legfeljebb 1 lehet (azaz a véges harmadostörtek két alakja közül legalább az egyik tartalmaz 1-et). Tehát $|\{0, 1\}^{\mathbb{N}}| \leq |[0, 1]|$.

A Cantor-Bernstein-Schröder tétel alapján $|\{0, 1\}^{\mathbb{N}}| = |[0, 1]|$.

Megszámlálhatóan végtelen vs continuum számosság

Tétel

$$|\mathbb{R}| > |\mathbb{N}|$$

Bizonyítás:

Mivel $|\{0, 1\}^{\mathbb{N}}| = |\mathbb{R}|$, ezért elég belátni, hogy $|\{0, 1\}^{\mathbb{N}}| > |\mathbb{N}|$.

$$|\{0, 1\}^{\mathbb{N}}| \geq |\mathbb{N}|:$$

$$H_0 := \{(1, 0, 0, 0, \dots), (0, 1, 0, 0, \dots), (0, 0, 1, 0, \dots), \dots\}$$

$$H_0 \subset \{0, 1\}^{\mathbb{N}}, \text{ és } |H_0| = |\mathbb{N}|.$$

$$|\{0, 1\}^{\mathbb{N}}| \neq |\mathbb{N}|:$$

Indirekt tegyük fel, hogy bijekcióba lehet állítani $\{0, 1\}^{\mathbb{N}}$ elemeit \mathbb{N}^+ elemeivel, azaz $\{0, 1\}^{\mathbb{N}} = \{u_i \mid i \in \mathbb{N}\} = \{u_1, u_2, \dots\}$ a $\{0, 1\}^{\mathbb{N}}$ elemeinek egy felsorolása (a pozitív természetes számokkal való indexelése).

A Cantor-féle átlós módszer

Jelölje $u_{i,j}$ u_i j . bitjét ($i, j \in \mathbb{N}^+, u_{i,j} \in \{0, 1\}$), azaz

$$u_i = (u_{i,1}, u_{i,2}, \dots, u_{i,j}, \dots).$$

Tekintsük az $u = (\overline{u_{1,1}}, \overline{u_{2,2}}, \dots, \overline{u_{i,i}}, \dots)$ megszámlálhatóan végtelen hosszúságú bináris (azaz $\{0, 1\}^{\mathbb{N}}$ -beli) szót, ahol $\overline{b} = 0$, ha $b = 1$ és $\overline{b} = 1$, ha $b = 0$.

Mivel, minden megszámlálhatóan végtelen hosszúságú bináris szó fel van sorolva, ezért létezik olyan $k \in \mathbb{N}^+$, melyre $u = u_k$.

Ekkor u k .bitje $u_{k,k}$ (így jelöltük u_k k . bitjét), másrészt $\overline{u_{k,k}}$ (így definiáltuk u -t).

De egy bit nem lehet 0 és 1 egyszerre, tehát az indirekt feltevésünk, azaz hogy $\{0, 1\}^{\mathbb{N}}$ és \mathbb{N}^+ között \exists bijekció helytelen volt.

Megjegyzés: A bizonyítás módszerét **Cantor-féle átlós módszernek** nevezik.

Túl sok a nyelv

Következmény

A $\{0, 1\}$ feletti nyelvek halmazának számossága nagyobb, mint a $\{0, 1\}$ feletti szavak számossága.

Ezekhez csak foglalkozzunk össze amit tudunk:

$$|\mathbb{R}| = |[0, 1]| = |\{0, 1\}^{\mathbb{N}}| = |\{L \mid L \subseteq \{0, 1\}^*\}| > |\mathbb{N}| = |\{0, 1\}^*|.$$

Észrevétel: $\{L \mid L \subseteq \{0, 1\}^*\} = \mathcal{P}(\{0, 1\}^*)$.

Igaz-e általában, hogy $|\mathcal{P}(H)| > |H|$?

Hatványhalmaz számossága

Tétel

Minden H halmazra $|\mathcal{P}(H)| > |H|$.

Bizonyítás: [Cantor-féle átlós módszerrel]

$|\mathcal{P}(H)| \geq |H|$, hiszen $\{\{h\} \mid h \in H\} \subseteq \mathcal{P}(H)$.

$|\mathcal{P}(H)| \neq |H|$: Indirekt $\exists f : \mathcal{P}(H) \leftrightarrow H$ bijekció. Definálunk egy $A \subseteq H$ halmazt: $\forall x \in H : x \in A \Leftrightarrow x \notin f^{-1}(x)$

$f(A) \in A$ igaz-e? Ha igaz, $f(A) \notin A$, ha nem igaz $f(A) \in A$ következik A definíciójából. Tehát $f(A) \in A$ se igaz, se hamis nem lehet, ellentmondás.

Következmény

Minden számosságnál van nagyobb számosság, tehát végtelen sok számosság van.

$$\aleph_0 := |\mathbb{N}|, \mathfrak{c} := |\mathcal{P}(\mathbb{N})| = |\mathbb{R}|.$$

A Turing gépek egy elkódolása

Tegyük fel, hogy $\Sigma = \{0, 1\}$. Ez feltehető, mivel egy nagyobb input ábécé feletti szavak hatékonyan átkódolhatóak $\{0, 1\}$ felettiékké.

Definíció

Egy M Turing-gép **kódja** (jelölése $\langle M \rangle$) a következő:

Legyen $M = (Q, \{0, 1\}, \Gamma, \delta, q_0, q_i, q_n)$, ahol

- ▶ $Q = \{p_1, \dots, p_k\}$, $\Gamma = \{X_1, \dots, X_m\}$, $D_1 = R$, $D_2 = S$, $D_3 = L$
- ▶ $k \geq 3$, $p_1 = q_0$, $p_{k-1} = q_i$, $p_k = q_n$,
- ▶ $m \geq 3$, $X_1 = 0$, $X_2 = 1$, $X_3 = \sqcup$.
- ▶ Egy $\delta(p_i, X_j) = (p_r, X_s, D_t)$ átmenet kódja $0^i 10^j 10^r 10^s 10^t$.
- ▶ $\langle M \rangle$ az átmenetek kódjainak felsorolása 11-el elválasztva.

Észrevétel: $\langle M \rangle$ 0-val kezdődik és végződik, nem tartalmaz 3 darab 1-t egymás után.

Jelölés: $\langle M, w \rangle := \langle M \rangle 111w$

Létezik nem Turing-felismerhető nyelv

Tétel

Létezik nem Turing-felismerhető nyelv.

Bizonyítás: A TG-ek számossága megszámlálható (a fenti kódolás injekció $\{0,1\}^*$ -ba, amiről tudjuk, hogy megszámlálható). Másrészt azt is tudjuk, hogy a $\{0,1\}$ feletti nyelvek számossága continuum. Tehát nem jut minden nyelvre öt felismerő TG (minden TG egyetlen nyelvet ismer fel).

Megjegyzés: Tehát valójában a nyelvek „többsége” $\notin RE$.
Tudnánk-e konkrét nyelvet mondani?

Jelölés: Minden $i \geq 1$ -re,

- ▶ jelölje w_i a $\{0,1\}^*$ halmaz i -ik elemét a hossz-lexikografikus rendezés szerint.
- ▶ jelölje M_i a w_i által kódolt TG-t (ha w_i nem kódol TG-t, akkor M_i egy tetszőleges olyan TG, ami nem fogad el semmit)

$L_{\text{átló}}$ Turing-felismerhetetlen

Tétel

$L_{\text{átló}} := \{w_i \mid w_i \notin L(M_i)\} \notin RE$.

Bizonyítás: [Cantor-féle átlós módszerrel]

Tekintsük azt a mindkét dimenziójában megszámlálhatóan végtelen T bittáblázatot, melyre $T(i,j) = 1 \Leftrightarrow w_j \in L(M_i)$ ($i,j \geq 1$).

Legyen $z = (T(1,1), \dots, T(i,i), \dots)$ a T átlójában olvasható megszámlálhatóan végtelen hosszú bitsztring és \bar{z} a z bitenkénti komplementere. Ekkor:

- ▶ minden $i \geq 1$ -re, T i -ik sora az $L(M_i)$ nyelv karakterisztikus sorozata
- ▶ \bar{z} az $L_{\text{átló}}$ karakterisztikus sorozata.
- ▶ Minden TG-pel felismerhető, azaz RE-beli nyelv karakterisztikus sorozata megegyezik T valamelyik sorával.
- ▶ \bar{z} különbözik T minden sorától.
- ▶ Tehát $L_{\text{átló}}$ különbözik az összes RE-beli nyelvtől.

Az univerzális TG

Felismerhetőség

Univerzális nyelv: $L_U = \{\langle M, w \rangle \mid w \in L(M)\}$.

Tétel

$L_U \in RE$

Bizonyítás: Konstruálunk egy 4 szalagos U „univerzális” TG-et, ami minden M TG minden bementére szimulálja annak működését.

Feltehető, hogy M egyszalagos.

1. **szalag:** U ezt csak olvassa, itt olvasható végig $\langle M, w \rangle$.
2. **szalag:** M aktuális szalagtartalma és a fej helyzete (elkódolva a fentiek szerint)
3. **szalag:** M aktuális állapota (elkódolva a fentiek szerint)
4. **szalag:** segédszalag

Az univerzális TG

Felismerhetőség

U működése vázlatosan:

1. Megnézi, hogy a bemenetén szereplő szó első része kódol-e TG-t; ha nem \Rightarrow elutasítja a bemenetet
2. ha igen \Rightarrow felmásolja w -t a 2., q_0 kódját a 3. szalagra
3. Szimulálja M egy lépését:
 - Leolvassa a második szalagról M aktuálisan olvasott szalagszimbólumát.
 - Leolvassa a harmadik szalagról M aktuális állapotát.
 - Szimulálja M egy lépését M első szalagon található leírása alapján. Ehhez U számára ehhez minden információ rendelkezésre áll. A 2. szalagon elő kell állítania az új szalagtartalmat a fej helyzetével és a 3. szalagon az új állapotot. Ehhez, ha szükséges használja a 4. szalagot. A megvalósítás átmenetszintű részletezésétől eltekintünk.
4. Ha M aktuális állapota elfogadó/elutasító, akkor U is belép a saját elfogadó/elutasító állapotába. Különben goto 3.

Az univerzális TG

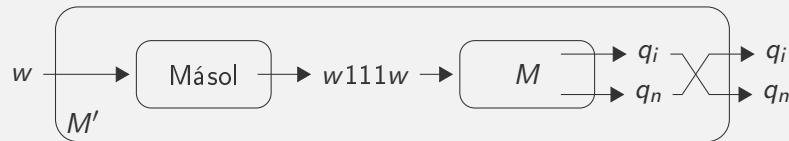
Eldönthetlenség

Megjegyzés: Ha M nem áll meg w -n, akkor U se áll meg $\langle M, w \rangle$ -n, így U nem dönti el L_U -t, csak felismeri.

Tétel

$L_U \notin R$.

Bizonyítás: Indirekt, tegyük fel, hogy létezik L_U -t eldöntő M TG. M -et felhasználva készítünk egy $L_{\text{átló}}$ -t felismerő M' TG-et.



$w \in L(M') \Leftrightarrow w111w \notin L(M) \Leftrightarrow$ a w által kódolt TG nem fogadja el w -t $\Leftrightarrow w \in L_{\text{átló}}$.

Tehát $L(M') = L_{\text{átló}}$, ami lehetetlen egy előző tétel miatt.

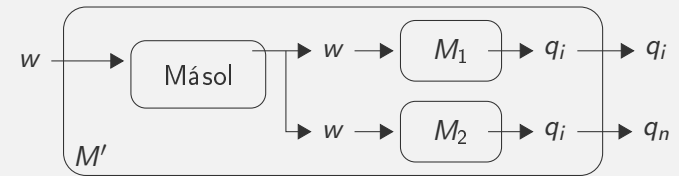
RE és R tulajdonságai

Jelölés: Ha $L \subseteq \Sigma^*$, akkor jelölje $\bar{L} = \{u \in \Sigma^* \mid u \notin L\}$.

Tétel

Ha L és $\bar{L} \in RE$, akkor $L \in R$.

Bizonyítás: Legyen M_1 és M_2 rendre az L -t és \bar{L} -t felismerő TG. Konstruáljuk meg az M' kétszalagos TG-t:



M' lemásolja w -t a második szalagjára, majd felváltva szimulálja M_1 és M_2 egy-egy lépését addig, amíg valamelyik elfogadó állapotba lép.

Így M' az L -et ismeri fel, és minden bemeneten meg is áll, azaz $L \in R$.

RE és R tulajdonságai

Következmény

RE nem zárt a komplementer-képzésre.

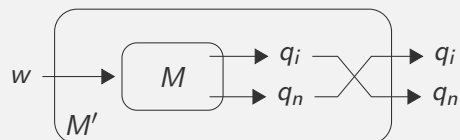
Bizonyítás:

Legyen $L \in RE \setminus R$ (L_U pl. egy ilyen nyelv) Ekkor $\bar{L} \notin RE$, hiszen ha $\bar{L} \in RE$ lenne, akkor ebből az előző tétel miatt $L \in R$ következne, ami ellentmondás.

Tétel

Ha $L \in R$, akkor $\bar{L} \in R$. (Azaz R zárt a komplementer-képzésre.)

Bizonyítás: Legyen $L \in R$ és M egy TG, ami az L -t dönti el. Akkor az alábbi M' \bar{L} -t dönti el:



Számítási feladatok megoldása TG-pel

Az eldöntési (igen/nem kimenetű) problémák általánosításai a (ki)számítási problémák. Ilyenkor a kimenet bármi lehet. A kiszámítási problémákra is algoritmikus megoldást keresünk.

Feltehetjük (megfelelő kódolás alkalmazásával), hogy f értelmezési tartománya Σ^* , értékkészlete Δ^* valamely Σ, Δ ábécékre.

Definíció

Azt mondjuk, hogy az $M = \langle Q, \Sigma, \Delta, \delta, q_0, q_i, (q_n) \rangle$ TG **kiszámítja** az $f : \Sigma^* \rightarrow \Delta^*$ szöfüggvényt, ha minden $u \in \Sigma^*$ -beli szóra megáll, és ekkor $f(u) \in \Delta^*$ olvasható az utolsó szalagján.

Megjegyzés: A definíció értelmében nincs szükség q_i és q_n megkülönböztetésére, elég lenne egyetlen megállási állapot. [Ezért van q_n ()-ben.]

Példa:

$f(u) = uc$ ($u \in \{a, b\}^*$).
($\Sigma = \{a, b\}$, $\Delta = \{a, b, c\}$.)



Visszavezetés

Definíció

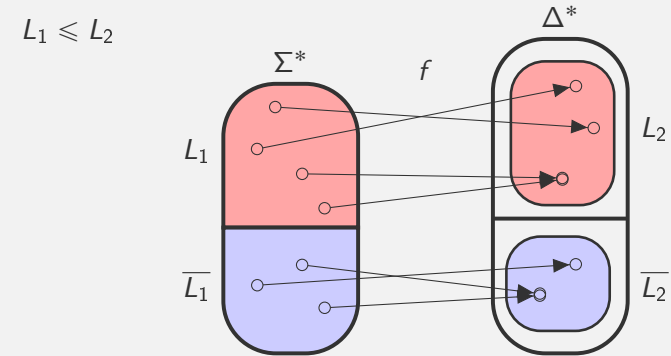
Az $f : \Sigma^* \rightarrow \Delta^*$ szófüggvény **kiszámítható**, ha van olyan Turing-gép, ami kiszámítja. [lásd szófüggvényt kiszámító TG]

Definíció

$L_1 \subseteq \Sigma^*$ **visszavezethető** $L_2 \subseteq \Delta^*$ -ra, ha van olyan $f : \Sigma^* \rightarrow \Delta^*$ kiszámítható szófüggvény, hogy $w \in L_1 \Leftrightarrow f(w) \in L_2$. Jelölés: $L_1 \leq L_2$

Megjegyzés: A fogalom Emil Posttól származik, angol nyelvű szakirodalomban: many-one reducibility

Visszavezetés



f kiszámítható, az egész Σ^* -on értelmezett, $f(L_1) \subseteq L_2$ valamint $f(\bar{L}_1) \subseteq \bar{L}_2$. f nem kell hogy injektív legyen és az se kell, hogy szürjektív.

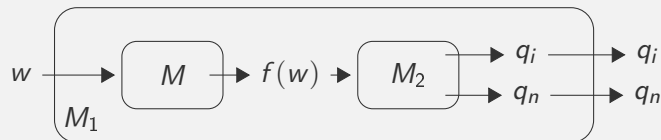
Tétel

- ▶ Ha $L_1 \leq L_2$ és $L_2 \in RE$, akkor $L_1 \in RE$.
- ▶ Ha $L_1 \leq L_2$ és $L_2 \in R$, akkor $L_1 \in R$.

Visszavezetés

Bizonyítás:

Legyen $L_2 \in RE$ ($\in R$) és tegyük fel, hogy $L_1 \leq L_2$. Legyen M_2 az L_2 -t felismerő (eldöntő), M pedig a visszavezetést kiszámító TG. Konstruáljuk meg M_1 -et:



Ha M_2 felismeri L_2 -t M_1 is fel fogja ismerni L_1 -t, ha el is dönti, akkor M_1 is el fogja dönteni.

Következmény

- ▶ Ha $L_1 \leq L_2$ és $L_1 \notin RE$, akkor $L_2 \notin RE$.
- ▶ Ha $L_1 \leq L_2$ és $L_1 \notin R$, akkor $L_2 \notin R$.

Bizonyítás: Indirekt bizonyítással azonnal adódik a fenti tételből.

A Turing gépek megállási problémája

Megállási probléma: megáll-e M w -n?

$$L_h = \{\langle M, w \rangle \mid M \text{ megáll a } w \text{ bemeneten}\}.$$

Megjegyzés: más jegyzetekben L_{halt} néven is előfordulhat.

Észrevétel: $L_u \subseteq L_h$

Kérdés: Igaz-e ha $A \subseteq B$, és A eldönthetetlen akkor B is az? Nem.

Tétel

$L_h \notin R$.

Bizonyítás: Az előző tétel következménye alapján elég megmutatni, hogy $L_u \leq L_h$, hiszen tudjuk, hogy $L_u \notin R$.

Tetszőleges M TG-re, legyen M' az alábbi TG:

M' tetszőleges u bemeneten a következőket teszi:

1. Futtatja M -et u -n
2. Ha M q_i -be lép, akkor M' is q_i -be lép
3. Ha M q_n -be lép, akkor M' végtelen ciklusba kerül

A Turing gépek megállási problémája

Belátható, hogy

- ▶ $f : \langle M, w \rangle \rightarrow \langle M', w \rangle$ kiszámítható függvény
- ▶ Tetszőleges (M, w) (TG,input)-párra $\langle M, w \rangle \in L_u \Leftrightarrow M$ elfogadja w -t $\Leftrightarrow M'$ megáll w -n $\Leftrightarrow \langle M', w \rangle \in L_h$

Tehát f által L_u visszavezethető L_h -ra. Így $L_h \notin R$.

Megjegyzés: Visszavezetések megadásakor jellemzően csak azon szavakra térünk ki, amelyek ténylegesen kódolnak valamilyen nyelvbeli objektumot (TG-t, (TG,szó) párt, stb.)

Pl. a fenti esetben nem foglalkoztunk azzal, hogy f mit rendeljen olyan szavakhoz, melyek nem kódolnak (TG, szó) párt. Ez általában egy könnyen kezelhető eset, most:

$$f(x) = \begin{cases} \langle M', w \rangle & \text{ha } \exists M \text{ TG, hogy } x = \langle M, w \rangle \\ \varepsilon & \text{egyébként,} \end{cases} \quad (x \in \{0, 1\}^*)$$

hiszen ε nem kódol (TG,szó) párt (L_h elemei (TG,szó) párok).

A Turing gépek megállási problémája

Tétel

$L_h \in RE$.

Bizonyítás: Az előző tétel alapján elég megmutatni, hogy $L_h \leq L_u$, hiszen tudjuk, hogy $L_u \in RE$. Tetszőleges M Turing-gépre, legyen M' az alábbi TG: M' tetszőleges u bemeneten a következőket teszi:

1. Futtatja M -et u -n
2. Ha M q_i -be lép, akkor M' is q_i -be lép
3. Ha M q_n -be lép, akkor M' q_i -be lép

Belátható, hogy

- ▶ $f : \langle M, w \rangle \rightarrow \langle M', w \rangle$ kiszámítható függvény
- ▶ Tetszőleges (M, w) (TG,input)-párra $\langle M, w \rangle \in L_h \Leftrightarrow M$ megáll w -n $\Leftrightarrow M'$ elfogadja w -t $\Leftrightarrow \langle M', w \rangle \in L_u$

Tehát f által L_h visszavezethető L_u -ra.

A számításelmélet alapjai II.

6. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Rice tétel

Definíció

Tetszőleges $\mathcal{P} \subseteq RE$ halmazt a rekurzívan felsorolható nyelvek egy tulajdonságának nevezzük. \mathcal{P} **triviális**, ha $\mathcal{P} = \emptyset$ vagy $\mathcal{P} = RE$.

$$L_{\mathcal{P}} = \{\langle M \rangle \mid L(M) \in \mathcal{P}\}.$$

Rice tétele

Ha $\mathcal{P} \subseteq RE$ egy nem triviális tulajdonság, akkor $L_{\mathcal{P}} \notin R$.

Rice tétel

Bizonyítás:

1. eset $\emptyset \notin \mathcal{P}$.

Mivel tudjuk, hogy $L_u \notin R$, elég belátni, hogy $L_u \leq L_{\mathcal{P}}$.

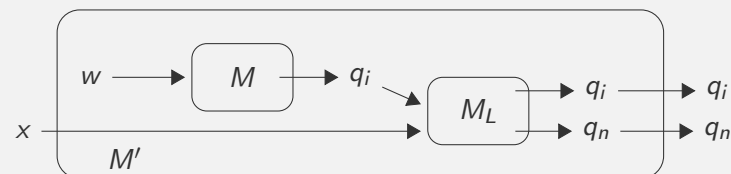
Mivel \mathcal{P} nem triviális, ezért létezik $L \in \mathcal{P}$. ($L \neq \emptyset$).

$L \in RE$, ezért van olyan M_L TG, melyre $L(M_L) = L$.

Egy tetszőleges $\langle M, w \rangle$ TG – bemenet pároshoz elkészítünk egy M' kétszalagos TG-t, mely egy x bemenetén a következőképpen működik:

1. Bemenetétől függetlenül először szimulálja M -et w -re a második szalagján.
2. Így, ha M nem áll meg w -n, akkor M' nem áll meg egyetlen inputjára sem. Ez esetben $L(M') = \emptyset$.
3. Ha M elutasítja w -t, akkor M' q_n -be lép és leáll (azaz nem fogadja el x -et). Ez esetben is $L(M') = \emptyset$.
4. Ha M elfogadja w -t, akkor M' szimulálja M_L -et x -en. Ekkor M_L definíciója miatt $L(M') = L$.

Rice tétel



Összefoglalva

- ▶ $\langle M, w \rangle \in L_u \Rightarrow L(M') = L \Rightarrow L(M') \in \mathcal{P} \Rightarrow \langle M' \rangle \in L_{\mathcal{P}}$.
- ▶ $\langle M, w \rangle \notin L_u \Rightarrow L(M') = \emptyset \Rightarrow L(M') \notin \mathcal{P} \Rightarrow \langle M' \rangle \notin L_{\mathcal{P}}$.

Azaz:

$\langle M, w \rangle \in L_u \Leftrightarrow \langle M' \rangle \in L_{\mathcal{P}}$, tehát $L_u \leq L_{\mathcal{P}}$ és így $L_{\mathcal{P}} \notin R$.

Rice tétel

2. eset $\emptyset \in \mathcal{P}$.

- ▶ Alkalmazhatjuk az 1. eset eredményét $\overline{\mathcal{P}} = RE \setminus \mathcal{P}$ -re, hiszen ekkor $\overline{\mathcal{P}}$ szintén nem triviális és $\emptyset \notin \overline{\mathcal{P}}$.
- ▶ Azt kapjuk, hogy $L_{\overline{\mathcal{P}}} \notin R$.
- ▶ $\overline{L_{\mathcal{P}}}$: azon $\{0,1\}$ feletti szavak, amelyek nem kódjai egy olyan M TG-nek, amelyre $L(M) \in \mathcal{P}$ tulajdonságú.
 $L_{\overline{\mathcal{P}}}$: olyan M TG-ek kódjai, melyre $L(M)$ nem \mathcal{P} tulajdonságú.
Megállapodásunk szerint azonban minden $\{0,1\}$ feletti szó TG kód, a nem kellő alakú szavak egy rögzített, egyetlen szót sem elfogadó TG-et kódolnak, így $\overline{L_{\mathcal{P}}} = L_{\overline{\mathcal{P}}}$.
- ▶ $\overline{L_{\mathcal{P}}} \notin R \Rightarrow L_{\mathcal{P}} \notin R$ (tétel volt). \square

Rice tétel

Alkalmazások

Következmények:

Eldönthetetlen, hogy egy M TG

- ▶ az üres nyelvet ismer-e fel. ($\mathcal{P} = \{\emptyset\}$)
- ▶ véges nyelvet ismer-e fel ($\mathcal{P} = \{L \mid L \text{ véges} \}$)
- ▶ környezetfüggetlen nyelvet ismer-e fel ($\mathcal{P} = \{L \mid L \text{ környezetfüggetlen} \}$)
- ▶ elfogadja-e az üres szót ($\mathcal{P} = \{L \in RE \mid \varepsilon \in L\}$)

Post Megfelelkezési Probléma

Definíció

Legyen Σ egy ábécé és legyenek $u_1, \dots, u_n, v_1, \dots, v_n \in \Sigma^+ \ (n \geq 1)$.

A $D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_n}{v_n} \right\}$ halmazt **dominókészletnek** nevezzük.

(Valójában az i . dominó egy az u_i és v_i szavakból álló rendezett pár. u_i -t a dominó felső, míg v_i -t a dominó alsó szavának nevezzük.)

Definíció

Az $\frac{u_1}{v_1} \dots \frac{u_m}{v_m}$ dominósorozat ($m \geq 1, 1 \leq i_1, \dots, i_m \leq n$) a

$D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_n}{v_n} \right\}$ dominókészlet egy **megoldása**, ha
 $u_{i_1} \dots u_{i_m} = v_{i_1} \dots v_{i_m}$.

Post Megfelelkezési Probléma

Példa: A $\left\{ \frac{b}{ca}, \frac{dd}{e}, \frac{a}{ab}, \frac{ca}{a}, \frac{abc}{c} \right\}$ készlet egy lehetséges megoldása

$$\frac{a}{ab} \frac{b}{ca} \frac{ca}{a} \frac{a}{ab} \frac{abc}{c}.$$

Egy másik megoldás: $\frac{a}{ab} \frac{b}{ca} \frac{ca}{a} \frac{a}{ab} \frac{abc}{c} \frac{a}{ab} \frac{b}{ca} \frac{ca}{a} \frac{a}{ab} \frac{abc}{c}.$

Megjegyzés: Tehát egy megoldáshoz a dominók többször felhasználhatók és nem kell minden dominót felhasználni. Egy dominókészletnek több megoldása is lehet.

Megoldás alatt véges (de akármekkora) hosszúságú kirakást értünk.

Vegyük észre, hogy hiába véges maga a készlet, végtelen sok féleképpen lehet a készlet dominóit véges sorozatba egymás után rakni, így megoldás keresésekor egy végtelen keresési térrel van dolgunk.

Post Megfelelkezési Probléma (PMP):

$L_{\text{PMP}} = \{ \langle D \rangle \mid D\text{-nek van megoldása} \}.$

Post Megfelelkezési Probléma

Tétel

$L_{PMP} \in RE$.

Bizonyítás: Ha D -t egy ábécének tekintjük, akkor éppen a D feletti szavak a potenciális megoldások.

Egy olyan TG, mely a D feletti szavakat hosszlexikografikus sorrendben sorra kipróbálja és ha megoldást talál q_i -ben leáll éppen L_{PMP} -t ismeri fel.

Tétel

$L_{PMP} \notin R$.

Bizonyítás:

Definiáljuk a PMP egy módosított változatát, MPMP-t. Az MPMP probléma igen-példányai olyan (D, d) (dominókészlet, dominó) párok, melyre D -nek van d -vel kezdődő megoldása.

$L_{MPMP} = \{ \langle D, d \rangle \mid d \in D \wedge D\text{-nek van } d\text{-vel kezdődő megoldása} \}$.

Post Megfelelkezési Probléma

$L_{PMP} = \{ \langle D \rangle \mid D\text{-nek van megoldása} \}$,

$L_{MPMP} = \{ \langle D, d \rangle \mid d \in D \wedge D\text{-nek van } d\text{-vel kezdődő megoldása} \}$.

Először megmutatjuk, hogy $L_{MPMP} \leq L_{PMP}$.

Jelölés: ha $u = a_1 \cdots a_n \in \Sigma^+$ és $*$ $\notin \Sigma$ akkor legyen

$\text{balcsillag}(u) := * a_1 * a_2 \cdots * a_n$

$\text{jobbcsillag}(u) := a_1 * a_2 * \cdots * a_n *$

$\text{baljobbcsillag}(u) := * a_1 * a_2 * \cdots * a_n *$.

Legyen $D = \{d_1, \dots, d_n\}$ egy tetszőleges dominókészlet, ahol $d_i = \frac{u_i}{v_i}$ ($1 \leq i \leq n$).

D' legyen a következő $|D| + 2$ méretű készlet ($\# \notin \Sigma$):

$$d'_i = \frac{\text{balcsillag}(u_i)}{\text{jobbcsillag}(v_i)} \quad (1 \leq i \leq n)$$

$$d'_0 = \frac{\text{balcsillag}(u_1)}{\text{baljobbcsillag}(v_1)}, \quad d'_{n+1} = \frac{* \#}{\#}$$

Post Megfelelkezési Probléma

Példa: Ha

$$D = \left\{ \frac{ab}{a}, \frac{c}{bc} \right\},$$

akkor

$$D' = \left\{ \frac{* a * b}{* a *}, \frac{* a * b}{a *}, \frac{* c}{b * c *}, \frac{* \#}{\#} \right\}$$

Állítás: $\langle D, d_1 \rangle \in L_{MPMP} \iff \langle D' \rangle \in L_{PMP}$.

Az állítás bizonyítása:

- ha $d_{i_1} \cdots d_{i_m}$ MPMP egy (D, d_1) bemenetének egy megoldása, akkor $d'_0 d'_{i_2} \cdots d'_{i_m} d'_{n+1}$ megoldása a D' PMP inputnak.
- ha $d'_{i_1} \cdots d'_{i_m}$ D' -nek, mint PMP inputnak egy megoldása, akkor az első illetve az utolsó betű egyezése miatt ez csak úgy lehetséges, hogy $d'_{i_1} = d'_0$ és $d'_{i_m} = d'_{n+1}$. Ekkor viszont $d_{i_1} \cdots d_{i_{m-1}}$ megoldása a (D, d_1) MPMP bemenetnek.

Ezzel az állítást bizonyítottuk. Mivel ez a megfeleltetés nyilván TG-pel kiszámítható, ezért $L_{MPMP} \leq L_{PMP}$.

Post Megfelelkezési Probléma

Most megmutatjuk, hogy $L_u \leq L_{MPMP}$.

Minden $\langle M, w \rangle$ (TG, szó) párhoz megadunk egy $\langle D, d \rangle$ (dominókészlet, kezdődominó) párt, úgy hogy

$w \in L(M) \iff D\text{-nek van } d\text{-vel kezdődő megoldása}$.

Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ és $w = a_1 \cdots a_n \in \Sigma^*$. (D, d) konstrukciója:

- $d := \frac{\#}{\# q_0 a_1 \cdots a_n \#}$ (ahol $\# \notin \Sigma$) $d \in D$
- ha $\delta(p, a) = (q, b, R)$, akkor $\frac{p a}{b q} \in D$
- ha $\delta(p, a) = (q, b, L)$, akkor $(\forall c \in \Gamma :) \frac{c p a}{q c b} \in D$
- ha $\delta(p, a) = (q, b, S)$, akkor $\frac{p a}{q b} \in D$
- $(\forall a \in \Gamma :) \frac{a}{a} \in D$
- $\frac{\#}{\#}, \frac{\#}{\sqcup \#}, \frac{\#}{\# \sqcup} \in D$
- $(\forall a \in \Gamma :) \frac{a q_i}{q_i}, \frac{q_i a}{q_i} \in D$
- $\frac{q_i \# \#}{\#} \in D$.

Post Megfelelkezési Probléma

Példa:

Ha M -nek $\delta(q_0, b) = (q_2, a, R)$ és $\delta(q_2, a) = (q_i, b, S)$ átmenetei, akkor $q_0bab \vdash aq_2ab \vdash aq_i b b$ egy bab -ot elfogadó konfigurációátmenet.

Az $\langle M, bab \rangle$ -hoz tartozó dominókészlet tartalmazza többek között a

$\frac{\#}{\#q_0bab\#}$ kezdő-, $\frac{q_0b}{aq_2}$ és $\frac{q_2a}{q_i b}$ átmenet-, $\frac{a}{a}$, $\frac{b}{b}$, $\frac{\sqcup}{\sqcup}$ és $\frac{\#}{\#}$ identikus dominókat valamint a befejezéshez szükséges $\frac{aq_i}{q_i}$, $\frac{q_i b}{q_i}$ és $\frac{q_i \# \#}{\#}$ dominókat.

Ekkor egy kirakás ($|$ -al blokkokra osztva):

$$\frac{\#}{\#q_0bab\#} \mid \frac{q_0b \ a \ b \ \#}{aq_2 \ a \ b \ \#} \mid \frac{a \ q_2a \ b \ \#}{a \ q_i b \ b \ \#} \mid \frac{aq_i \ b \ b \ \#}{q_i \ b \ b \ \#} \mid \frac{q_i b \ b \ \#}{q_i \ b \ \#} \mid \frac{q_i b \ \#}{q_i \ \#} \mid \frac{q_i \# \#}{\#}$$

Post Megfelelkezési Probléma

$$\frac{\#}{\#q_0bab\#} \mid \frac{q_0b \ a \ b \ \#}{aq_2 \ a \ b \ \#} \mid \frac{a \ q_2a \ b \ \#}{a \ q_i b \ b \ \#} \mid \frac{aq_i \ b \ b \ \#}{q_i \ b \ b \ \#} \mid \frac{q_i b \ b \ \#}{q_i \ b \ \#} \mid \frac{q_i b \ \#}{q_i \ \#} \mid \frac{q_i \# \#}{\#}$$

A példán szemléltetjük, hogy $w \in L(M) \Rightarrow \exists \langle D, d \rangle$ -nek megoldása.

Az első blokk csak a $d = \frac{\#}{\#q_0bab\#}$ kezdődominóból áll.

A következő két blokkban alul és felül is konfigurációk következnek, felül mindig eggyel "lemaradva".

A 4.-6. blokkokban a $\frac{aq_i}{q_i}$ (és $\frac{q_i a}{q_i}$) típusú dominókkal egyesével „behozható” a felső szó lemaradása, egészen addig, amíg az alsó rész már csak $q_i \#$ -al hosszabb.

Végül a 7. blokkban csak egy (záró)dominó szerepel, melynek az a szerepe, hogy „behozza” a még megmaradt lemaradást.

Ugyanilyen módon megkonstruálható egy megoldás minden olyan esetben, ha $w \in L(M)$. Azaz, $w \in L(M) \Rightarrow \exists \langle D, d \rangle$ -nek megoldása.

Post Megfelelkezési Probléma

Másrészt, ha van $\langle D, d \rangle$ -nek megoldása, akkor ebben a megoldás alsó és a felső szava egyenlő hosszú (sőt ugyanaz a szó!), így tartalmaznia kell q_i -t tartalmazó dominót, hiszen csak ezekben hosszabb a felső szó, d -ben viszont az alsó az.

Meggondolható, hogy minden kirakás alsó szava az első q_i -t követő $\#$ -ig egy $\#$ -ekkel elválasztott elfogadó konfigurációban végződő konfigurációsorozat kell legyen a w -hez tartozó kezdőkonfigurációból. Tehát a w szóhoz tartozó kezdőkonfigurációból M el tud jutni elfogadó konfigurációba, azaz $w \in L(M)$.

Nyilván $\langle D, d \rangle \langle M, w \rangle$ -ből TG-pel kiszámítható, így a visszavezetéshez használt függvény kiszámítható. Tehát beláttuk, hogy $L_u \leq L_{MPMP}$.

Post Megfelelkezési Probléma

Állítás: A visszavezetés tranzitív.

Az állítás bizonyítása: Legyen $L_i \subseteq \Sigma_i^*$ ($i = 1, 2, 3$), $L_1 \leq L_2$ és $L_2 \leq L_3$. Legyen továbbá $f : \Sigma_1^* \rightarrow \Sigma_2^*$ és $g : \Sigma_2^* \rightarrow \Sigma_3^*$ a visszavezetés definíciója alapján létező két kiszámítható szófüggvény, amelyekre $f(L_1) \subseteq L_2$, $f(\bar{L}_1) \subseteq \bar{L}_2$, $g(L_2) \subseteq L_3$, $g(\bar{L}_2) \subseteq \bar{L}_3$.

Ekkor $g \circ f$ nyilván kiszámítható és visszavezeti L_1 -et L_3 -ra, hiszen $g \circ f(L_1) = g(f(L_1)) \subseteq g(L_2) \subseteq L_3$ és $g \circ f(\bar{L}_1) = g(f(\bar{L}_1)) \subseteq g(\bar{L}_2) \subseteq \bar{L}_3$. \square

Innen a tétel bizonyítása: $L_u \leq L_{MPMP}$, $L_{MPMP} \leq L_{PMP}$ és tudjuk már, hogy $L_u \notin R$. Ebből a visszavezetés tranzitivitása és korábbi tételünk alapján $L_{PMP} \notin R$. \square

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

(Volt:) Egy G környezetfüggetlen (CF, 2-es típusú) grammatikát **egyértelműnek** neveztünk, ha minden $L(G)$ -beli szónak pontosan egy baloldali levezetése van G -ben. (**Baloldali levezetés:** mindig a legbaloldalibb nemterminálist írjuk át a mondatformában.)

$L_{ECF} := \{ \langle G \rangle \mid G \text{ egyértelmű CF grammatika} \}$.

Tétel

$L_{ECF} \notin R$

Bizonyítás: Megmutatjuk, hogy $L_{PMP} \leq \overline{L_{ECF}}$.

Legyen $D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_n}{v_n} \right\}$ egy tetszőleges dominókészlet a Σ ábécé felett.

$\Delta := \{a_1, \dots, a_n\}$ úgy, hogy $\Sigma \cap \Delta = \emptyset$.

$P_A := \{A \rightarrow u_1 A a_1, \dots, A \rightarrow u_n A a_n, A \rightarrow \varepsilon\}$.

$P_B := \{B \rightarrow v_1 B a_1, \dots, B \rightarrow v_n B a_n, B \rightarrow \varepsilon\}$.

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

$G_A = \langle A, \{A\}, \Sigma \cup \Delta, P_A \rangle$. $G_B = \langle B, \{B\}, \Sigma \cup \Delta, P_B \rangle$.

$G_D = \langle S, \{S, A, B\}, \Sigma \cup \Delta, \{S \rightarrow A, S \rightarrow B\} \cup P_A \cup P_B \rangle$.

$f : \langle D \rangle \rightarrow \langle G_D \rangle$ visszavezetés, mert:

- ha $\frac{u_1}{v_1} \dots \frac{u_m}{v_m}$ megoldása D -nek, akkor $u_1 \dots u_m = v_1 \dots v_m$.
De ekkor $u_1 \dots u_m a_{i_1} \dots a_{i_m} = v_1 \dots v_m a_{i_1} \dots a_{i_m}$
kétféleképpen is levezethető, így G_D nem egyértelmű.
- ha G_D nem egyértelmű, akkor van olyan szó, aminek két baloldali levezetése van. De ezek $S \rightarrow A$ -val illetve $S \rightarrow B$ -vel kell kezdődjenek, hiszen G_A és G_B egyértelmű. A generált szavak xy , $x \in \Sigma^*$, $y \in \Delta^*$ alakúak, így ugyanaz a generált Σ feletti prefix is. Így a két levezetés D egy megoldását adja.

f nyilván TG-pel kiszámítható. Mivel $L_{PMP} \notin R$, következik, hogy $\overline{L_{ECF}} \notin R$, amiből kapjuk, hogy $L_{ECF} \notin R$. \square

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

Lemma

Az előző tétel bizonyításában definiált G_A és G_B grammatikák esetén $\overline{L(G_A)}$ és $\overline{L(G_B)}$ környezetfüggetlen.

Bizonyítás: Az állítás nem nyilvánvaló, mivel a környezetfüggetlen nyelvek nem zártak a komplementer képzésre. Elég G_A -ra belátni az állítást, G_B -re ugyanígy bizonyítható.

Legyen $n_i := |u_i|$ ($1 \leq i \leq |D|$). $L(G_A)$ -hoz adható determinisztikus veremautomata.

Ötlet: Amíg Σ -beli betűk jönnek az inputon pakoljuk őket bele a verembe. Ha $a_i \in \Delta$ -beli betű jön, akkor próbáljuk meg kivenni u_i^{-1} -et a veremből. Megvalósítás:

$A = \langle \Sigma \cup \{\#\}, Q, \Sigma \cup \Delta, \delta, q_0, \#, \{s\} \rangle$, ahol

$$Q = \{q_0, r, s\} \cup \bigcup_{i=1}^{|D|} \{q_{i1}, \dots, q_{i(n_i-1)}\}$$

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

és M_δ :

$\#q_0 t \rightarrow \#t q_0 \quad (t \in \Sigma)$

$t_1 q_0 t_2 \rightarrow t_1 t_2 q_0 \quad (t_1, t_2 \in \Sigma)$

$t_{n_i} x a_i \rightarrow q_{i(n_i-1)} \quad (1 \leq i \leq |D|, x \in \{q_0, r\}, u_i = t_1 \dots t_{n_i}, n_i \geq 2)$

$t_j q_{ij} \rightarrow q_{i(j-1)} \quad (1 \leq i \leq |D|, 2 \leq j \leq n_i-1, u_i = t_1 \dots t_{n_i}, n_i \geq 2)$

$t_1 q_{i1} \rightarrow r \quad (1 \leq i \leq |D|, u_i = t_1 \dots t_{n_i}, n_i \geq 2)$

$tra_i \rightarrow r \quad (1 \leq i \leq |D|, u_i = t, t \in \Sigma)$

$\#r \rightarrow \#s$

A veremautomata $L(G_A)$ -t ismeri fel és determinisztikus.

(Pontosabban a szokásos módon kiegészíthető determinisztikussá egy csapdaállapottal és a hiányzó átmenetek odavezetésével.)

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

Állítás: A determinisztikus veremautomatával felismerhető nyelvek osztálya zárt a komplementerképzésre.

Az állítás bizonyítása: $Q \setminus F$ -re változtatva az elfogadó állapothalmazt a determinisztikus veremautomata épp a komplementer nyelvet ismeri fel.

Megjegyzés: Hasonlóan, bármilyen determinisztikus géptípus által eldönthető nyelvek osztálya zárt a komplementerképzésre.

Innen a lemma bizonyítása: Láttuk, hogy létezik $L(G_A)$ -t felismerő determinisztikus veremautomata. Az állítás szerint olyan determinisztikus veremautomata is van, amelyik $\overline{L(G_A)}$ -t ismeri fel. Minden veremautomata által felismert nyelv 2-típusú, így $\overline{L(G_A)}$ is.

□

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

Tétel

Eldönthetetlenek az alábbi, G_1 és G_2 környezetfüggetlen grammatikákkal kapcsolatos kérdések.

- (1) $L(G_1) \cap L(G_2) \stackrel{?}{=} \emptyset$
- (2) $L(G_1) \stackrel{?}{=} L(G_2)$
- (3) $L(G_1) \stackrel{?}{=} \Gamma^*$ valamely Γ ábécére
- (4) $L(G_1) \stackrel{?}{\subseteq} L(G_2)$

Bizonyítás:

(1) L_{PMP} -t vezethetjük vissza rá. Legyen $D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_n}{v_n} \right\}$ a dominókészlet. Készítsük el a fenti G_A és G_B grammatikákat. Könnyen látható, hogy D -nek akkor és csak akkor van megoldása, ha $L(G_A)$ -nak és $L(G_B)$ -nek a metszete nemüres.

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

(2) L_{PMP} -t vezethetjük vissza rá, ehhez tekintsük ismét G_A -t és G_B -t.

$L := \overline{L(G_A) \cap L(G_B)} = \overline{L(G_A)} \cup \overline{L(G_B)} \in \mathcal{L}_2$, mivel az előző Lemma szerint $\overline{L(G_A)} \in \mathcal{L}_2$ és $\overline{L(G_B)} \in \mathcal{L}_2$ az, és \mathcal{L}_2 zárt az unióra.

Legyenek G_1 és G_2 olyan környezetfüggetlen grammatikák, amelyekre $L(G_1) = L$ és $L(G_2) = (\Sigma \cup \Delta)^*$.

$L(G_1) = L(G_2) \Leftrightarrow L(G_A) \cap L(G_B) = \emptyset$, így ha (2) eldönthető volna, akkor az (1)-beli érvelés alapján L_{PMP} is az lenne, de láttuk, hogy nem az.

(3) Legyen G_1 ugyanaz, mint (2)-ben és $\Gamma = \Sigma \cup \Delta$. Pontosan az előbbi érveléssel (3) eldönthetősége L_{PMP} eldönthetőségét implikálná.

(4) Mivel $L(G_1) = L(G_2) \Leftrightarrow L(G_1) \subseteq L(G_2) \wedge L(G_2) \subseteq L(G_1)$, ezért a tartalmazás eldönthetősége (2) eldönthetőségét implikálná.

□

A számításelmélet alapjai II.

7. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Eldönthető problémák a nulladrendű logikában

Állítás: Eldönthetők az ítéletkalkulus alábbi algoritmikus kérdései:

- ▶ egy φ ítéletkalkulusbeli formula kielégíthető-e,
- ▶ egy φ ítéletkalkulusbeli formula kielégíthetetlen-e,
- ▶ egy φ ítéletkalkulusbeli formula tautológia-e,
- ▶ φ és ψ ítéletkalkulusbeli formulákra $\varphi \sim_0 \psi$ fennáll-e,
- ▶ egy \mathcal{F} véges ítéletkalkulusbeli formulahalmaz és egy φ formula esetén $\mathcal{F} \models_0 \varphi$ fennáll-e.

Bizonyítás: Készítsük el az ítéletábrák(ka)t a szóban forgó formulá(k)ra és olvassuk le belőlük. \square

Megjegyzés: A fenti algoritmikus kérdések eldönthetősége azon múlik, hogy véges sok interpretáció lehetséges, ezek egyesével megvizsgálhatóak. Mivel n ítéletváltozó esetén az ítéletábrának 2^n , azaz exponenciális sora van, ez a „brute force” módszer persze nem hatékony. Ugyan ismeretesek az ítéletábránál praktikusabban működő módszerek, azonban ezek mindegyike a legrosszabb esetben szintén exponenciális műveletigényű.

Eldönthetetlen problémák az elsőrendű logikában

A következőkben belátjuk, hogy az elsőrendű logikában olyan alapvető kérdések, mint hogy egy formula logikailag igaz, kielégíthető, kielégíthetetlen, illetve hogy egy formula egy formulahalmaz logikai következménye-e (algoritmikusan) eldönthetetlen.

Definíció

$\text{VALIDITYPRED} := \{ \langle \varphi \rangle \mid \varphi \text{ logikailag igaz elsőrendű formula} \}.$

$\text{UNSATPRED} := \{ \langle \varphi \rangle \mid \varphi \text{ kielégíthetetlen elsőrendű formula} \}.$

$\text{SATPRED} := \{ \langle \varphi \rangle \mid \varphi \text{ kielégíthető elsőrendű formula} \}.$

$\text{EQIVPRED} := \{ \langle \varphi, \psi \rangle \mid \varphi, \psi \text{ elsőrendű formulák, melyekre } \varphi \sim \psi \}.$

$\text{CONSPRED} := \{ \langle \mathcal{F}, \varphi \rangle \mid \mathcal{F} \text{ véges elsőrendű formulahalmaz, } \varphi \text{ elsőrendű formula, } \mathcal{F} \models \varphi \}.$

Megjegyzés: Itt $\langle \varphi \rangle$ a φ formula egy $\{0, 1\}$ feletti kódolása.

A TG-ek kódolásánál látott módon a nem-kódokhoz hozzárendelhetjük pl. \perp -t, a konstans kielégíthetetlen formulát, így feltehető, hogy $\overline{\text{UNSATPRED}} = \text{SATPRED}$.

Eldönthetetlen problémák az elsőrendű logikában

Tétel

$\text{VALIDITYPRED} \notin \text{R}$

Bizonyítás: L_{PMP} -t vezetjük vissza VALIDITYPRED -re, korábbi tételünk alapján ebből a tétel állítása következik. Minden D dominókészlethez megadunk egy φ_D elsőrendű formulát, amelyre teljesül, hogy D -nek akkor és csak akkor van megoldása, ha $\models \varphi_D$.

Legyen tehát $D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_k}{v_k} \right\}$ ($k \geq 1$) egy $\Sigma = \{a_1, \dots, a_n\}$ feletti dominókészlet.

Tekintsük azt az elsőrendű logikai nyelvet ahol

$\text{Pred} = \{p\}$, $\text{ar}(p) = 2$, $\text{Func} = \{f_{a_1}, \dots, f_{a_n}\}$,
 $\text{ar}(f_{a_i}) = 1$ ($\forall 1 \leq i \leq n$), $\text{Cnst} = \{c\}$.

Jelölés: $f_{b_1 \dots b_m}(t) := f_{b_1}(f_{b_2}(\dots(f_{b_m}(t))\dots))$ ahol $b_1 \dots b_m \in \Sigma$, t pedig egy term.

Eldönthetetlen problémák az elsőrendű logikában

Ekkor $\varphi_D := \varphi_1 \wedge \varphi_2 \rightarrow \varphi_3$, ahol

$$\varphi_1 = p(f_{u_1}(c), f_{v_1}(c)) \wedge \cdots \wedge p(f_{u_k}(c), f_{v_k}(c)),$$

$$\varphi_2 = \forall x \forall y (p(x, y) \rightarrow p(f_{u_1}(x), f_{v_1}(y)) \wedge \cdots \wedge p(f_{u_k}(x), f_{v_k}(y))),$$

$$\varphi_3 = \exists z p(z, z).$$

Először tegyük fel, hogy φ_D logikailag igaz. Legyen I a következő interpretáció. (φ_D zárt formula, így igazságértéke csak az interpretációtól függ.)

I alaphalmaza legyen Σ^* . $f_{a_i}^I(u) := a_i u$ ($1 \leq i \leq k, u \in \Sigma^*$), $c^I := \varepsilon$, p interpretációja pedig az alábbi. Tetszőleges $u, v \in \Sigma^*$ esetén $p^I(u, v) :=$ igaz akkor és csak akkor, ha az alábbi feltétel teljesül:

$$\text{van olyan } m \geq 1 \text{ és } 1 \leq i_1, \dots, i_m \leq k, \text{ hogy} \\ u_{i_1} \cdots u_{i_m} = u \text{ és } v_{i_1} \cdots v_{i_m} = v. \quad (*)$$

Vegyük észre, hogy a feltétel pontosan akkor teljesül, ha D néhány dominója egymás után tehető úgy hogy felül u , alul v olvasható.

Eldönthetetlen problémák az elsőrendű logikában

A u hosszára vonatkozó teljes indukcióval könnyen látható, hogy minden $u \in \Sigma^*$ esetén $|f_u(c)|^I = u$. Így minden $1 \leq i \leq k$ -ra $|f_{u_i}(c)|^I = u_i$ és $|f_{v_i}(c)|^I = v_i$ és így $p^I(u_i, v_i) =$ igaz. Tehát $I \models \varphi_1$.

Tegyük fel, hogy $p^I(u, v) =$ igaz valamely $u, v \in \Sigma^*$ -ra. Ekkor p^I definíciója szerint van olyan $m \geq 1$ és $1 \leq i_1, \dots, i_m \leq k$, hogy $u_{i_1} \cdots u_{i_m} = u$ és $v_{i_1} \cdots v_{i_m} = v$.

Ekkor nyilván minden $1 \leq i \leq k$ -ra az $u_i u$ és $v_i v$ szavakra is teljesül a (*) feltétel (az i, i_1, \dots, i_m index sorozat jó), tehát $p^I(u_i u, v_i v) =$ igaz.

Mivel $|f_{u_i}(u)|^I = u_i u$ és $|f_{v_i}(v)|^I = v_i v$, ezért bármely $u, v \in \Sigma^*$ -ra ha $p^I(u, v) =$ igaz, akkor $p^I(|f_{u_1}(u)|^I, |f_{v_1}(v)|^I) \wedge \cdots \wedge p^I(|f_{u_k}(u)|^I, |f_{v_k}(v)|^I) =$ igaz. Tehát $I \models \varphi_2$.

Mivel φ_D logikailag igaz, ezért $|\varphi_D|^I =$ igaz. Mivel $|\varphi_1|^I =$ igaz és $|\varphi_2|^I =$ igaz, ezért ez csak úgy lehet hogy $|\varphi_3|^I =$ igaz. φ_3 I -ben viszont akkor és csak akkor igaz, ha D -nek van megoldása.

Eldönthetetlen problémák az elsőrendű logikában

A másik irány bizonyításához tegyük fel most, hogy D -nek van megoldása és legyen $I = \langle U, I_{\text{Pred}}, I_{\text{Func}}, I_{\text{Cnst}} \rangle$ egy tetszőleges interpretáció. Be kell látni, hogy $I \models \varphi_D$.

Ha $I \not\models \varphi_1 \wedge \varphi_2$, akkor $I \models \varphi_D$ fennáll az implikáció igazságértékének definíciója miatt. Feltehető tehát, hogy $I \models \varphi_1$ és $I \models \varphi_2$.

Legyen $m \geq 1$ és $1 \leq i_1, \dots, i_m \leq k$ olyanok, hogy $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$ (ilyen létezik, mert D -nek van megoldása). $I \models \varphi_1$ -ből $I \models p(f_{u_{i_m}}(c), f_{v_{i_m}}(c))$ adódik.

Ebből $I \models \varphi_2$ miatt sorra (teljes indukcióval) adódik, hogy

$$\begin{aligned} I &\models p(f_{u_{i_m}}(c), f_{v_{i_m}}(c)) \\ I &\models p(f_{u_{i_{m-1}u_{i_m}}}(c), f_{v_{i_{m-1}v_{i_m}}}(c)) \\ &\vdots \\ I &\models p(f_{u_{i_1} \cdots u_{i_{m-1}u_{i_m}}}(c), f_{v_{i_1} \cdots v_{i_{m-1}v_{i_m}}}(c)) \end{aligned}$$

Eldönthetetlen problémák az elsőrendű logikában

Mivel $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$, ezért a $w = |f_{u_{i_1} \cdots u_{i_{m-1}u_{i_m}}}(c)|^I$ U -beli elemre $p^I(w, w) =$ igaz teljesül.

Tehát $I \models \varphi_3$ és így $I \models \varphi_D$.

D alapján a φ_D formula nyilván kiszámítható, így

$L_{\text{PMP}} \leq \text{VALIDITYPRED}$. A tétel állítása következik egy a visszavezetésről tanult tételből és abból, hogy $L_{\text{PMP}} \notin \text{R}$. \square

Következmény

$\text{UNSATPRED}, \text{SATPRED}, \text{EQUIVPRED}, \text{CONSPRED} \notin \text{R}$

Bizonyítás: φ kielégíthetetlen $\Leftrightarrow \models \neg \varphi$. Tehát az előző tétel alapján $\text{UNSATPRED} \notin \text{R}$.

Eldönthetetlen nyelv komplementere eldönthetetlen, tehát $\text{SATPRED} \notin \text{R}$.

φ kielégíthetetlen $\Leftrightarrow \varphi \sim \perp$, így $\text{EQUIVPRED} \notin \text{R}$.

$\emptyset \models \varphi \Leftrightarrow \varphi$ logikailag igaz, így $\text{CONSPRED} \notin \text{R}$. \square

Eldönthetetlen problémák az elsőrendű logikában

Megjegyzés: Van olyan parciális algoritmus, amely egy tetszőleges φ elsőrendű formulára pontosan akkor áll meg „igen” válasszal, ha φ kielégíthetetlen (ilyen például az elsőrendű logika rezolúciós algoritmus).

Az elsőrendű rezolúció ismertetése meghaladja az előadás kereteit így bizonyítás nélkül kimondjuk a következőt:

Tétel

$\text{UNSATPRED} \in \text{RE}$.

Következmény

$\text{SATPRED} \notin \text{RE}$

Bizonyítás: Korábbi tételünk volt hogy $L, \bar{L} \in \text{RE} \Rightarrow L \in \text{R}$. Mivel $\bar{\text{UNSATPRED}} = \text{SATPRED}$ és $\text{UNSATPRED} \in \text{RE} \setminus \text{R}$, ezért $\text{SATPRED} \notin \text{RE}$. \square

\mathcal{L}_0 és RE kapcsolata

Tétel

Minden G grammatikához megadható egy $L(G)$ -t felismerő NTG.

Bizonyítás: Legyen M -nek 3 szalagja, az első a TG bemenetét, a második a G grammatika szabályait tartalmazza. Ezeket a működés során csak olvassuk.

A harmadik szalagon mindig egy α mondatforma áll (kezdetben G kezdőszimbóluma).

A Turing gép nemdeterminisztikusan választ egy $p \rightarrow q$ szabályt és α -ban egy pozíciót. Ha az adott pozícióban éppen p kezdődik, azaz $\alpha = xpy$, akkor p -t q -ra cseréli, az új mondatforma xqy lesz.

Ha az 1. és a 3. szalag tartalma megegyezik a gép q_i -ben megáll. M ezt minden iteráció előtt ellenőrzi. Így $L(M) = L(G)$. \square

Következmény: Egy korábbi tételünk alapján persze determinisztikus TG is megadható G -hez.

\mathcal{L}_0 és RE kapcsolata

Tétel

Minden $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ determinisztikus TG-hez megadható egy $L(M)$ -et generáló G grammatika.

Bizonyítás: G mondatformái M konfigurációit fogják kódolni. A G grammatika éppen fordítottan fog haladni. Nemdeterminisztikusan előállít egy elfogadó konfigurációt, majd ebből megpróbál egy kezdőkonfigurációt levezetni.

\mathcal{L}_0 és RE kapcsolata

Legyen $G = \langle (\Gamma \setminus \Sigma) \cup Q \cup \{S, A, \triangleright, \triangleleft\}, \Sigma, P, S \rangle$.
 P szabályai:

1. $S \rightarrow \triangleright A q_i A \triangleleft$
2. $A \rightarrow aA \mid \varepsilon$ ($\forall a \in \Gamma$)
3. $bq' \rightarrow qa$, ha $\delta(q, a) = (q', b, R)$
4. $q'b \rightarrow qa$, ha $\delta(q, a) = (q', b, S)$
5. $q'cb \rightarrow cqa$, ha $\delta(q, a) = (q', b, L)$ ($\forall c \in \Gamma$)
6. $\sqcup \triangleleft \rightarrow \triangleleft, \triangleleft \rightarrow \varepsilon, \triangleright \sqcup \rightarrow \triangleright, \triangleright q_0 \rightarrow \varepsilon$

1-2. generálunk egy tetszőleges elfogadó konfigurációt

3-5. a konfigurációátmeneteket fordított irányban szimuláljuk. Pl. ha $\alpha cqa\beta \vdash \alpha q'cb\beta$ egy $\delta(q, a) = (q', b, L)$ szabály szerint, akkor most a grammatikában az 5-ös pont szerint $q'cb$ íródhat át cqa -ra.

6. Ha a mondatformánk egy kezdőkonfiguráció (esetleg néhány extra \sqcup -el), akkor ezek takarítják el a már felesleges jeleket.

\mathcal{L}_0 és RE kapcsolata

A konfigurációátmenet hosszára (n) vonatkozó indukcióval könnyen megmutatható, hogy

$upv \vdash^* u'qv' \ (p, q \in Q, u, u', v, v' \in \Gamma^*)$ akkor és csak akkor ha $\triangleright \sqcup^{i'} u'qv' \sqcup^{j'} \triangleleft \Rightarrow^* \triangleright \sqcup^i upv \sqcup^j \triangleleft$ valamely $i, i', j, j' \in \mathbb{N}$ -re.

$n = 0$ -ra nyilvánvaló. n -ről $n + 1$ -re nézzük meg a jobbrólépés példáján. Ha $upv \vdash^* u'qav' \vdash u'brv''$ valamely $a, b \in \Gamma, r \in Q$ -ra ahol $v'' = v'$ ha $v' \neq \varepsilon$, $v'' = \sqcup$ különben, akkor $\triangleright \sqcup^{i'} u'qav' \sqcup^{j'} \triangleleft \Rightarrow^* \triangleright \sqcup^i upv \sqcup^j \triangleleft$ valamely $i, i', j, j' \in \mathbb{N}$ -re az indukciós feltevés miatt és $\triangleright \sqcup^{i'} u'brv'' \sqcup^{j'} \triangleleft \Rightarrow \triangleright \sqcup^{i'} u'qav' \sqcup^{j'} \triangleleft$ mivel $bs \rightarrow qa \in P$. Tehát $\triangleright \sqcup^{i'} u'brv'' \sqcup^{j'} \triangleleft \Rightarrow^* \triangleright \sqcup^i upv \sqcup^j \triangleleft$, azaz a jobbrólépéssel végződő $n + 1$ hosszú konfigurációátmenetekre is igaz az állítás. Hasonlóan megy a bizonyítás az S és L irányokra illetve az állítás megfordítására.

Tehát $q_0w \vdash^* \alpha q_i \beta$ valamely $\alpha, \beta \in \Gamma^*$ -ra akkor és csak akkor, ha $S \Rightarrow^* \triangleright \alpha q_i \beta \triangleleft \Rightarrow^* \triangleright \sqcup^i q_0w \sqcup^j \triangleleft \Rightarrow^* w$. \square

\mathcal{L}_1 és R kapcsolata

Definíció

A **lineárisan korlátozott automata** (LKA) olyan **nemdeterminisztikus** TG, melynek Σ bemeneti ábécéje két speciális szimbólumot tartalmaz \triangleright -et (baloldali végejel/endmarker) és \triangleleft -et (jobboldali végejel/endmarkert). Ezen felül

- ▶ a bemenetek $\triangleright(\Sigma \setminus \{\triangleright, \triangleleft\})^* \triangleleft$ -beliek,
- ▶ \triangleright és \triangleleft nem írhatók felül
- ▶ \triangleright -tól balra illetve \triangleleft -tól jobbra nem állhat a fej.
- ▶ a fej kezdőpozíciója a \triangleright tartalmú cella jobb-szomszédja

Magyarán az LKA egy korlátozott munkaterülettel rendelkező NTG.

Megjegyzés: Nevét egy vele ekvivalens modellről kapta, amelyben a rendelkezésre álló tár az input hosszának konstansszorosa (lineáris függvénye). (Megmutatható, hogy egy konstans szorzó a megengedett munkaterület méretére nem növeli meg a gép számítási erejét.)

\mathcal{L}_1 és R kapcsolata

Tétel

- (1) Minden G 1-es típusú grammatikához megadható egy A LKA, melyre $L(A) = L(G)$.
- (2) Minden A LKA-hoz megadható egy G 1-es típusú grammatika, melyre $L(G) = L(A)$.

Bizonyítás:

- (1) Az előző előtti tételben láttuk, hogy minden 0. típusú grammatikához lehet konstruálni $L(G)$ -t felismerő NTG-t. A konstrukció a 3. szalagján nemdeterminisztikusan szimulált egy G -beli levezetést, az iterációk végén ellenőrizte, hogy a 3. szalagon lévő mondatforma megegyezik-e az első szalagon lévő u inputtal. Amennyiben G 1-es típusú, azaz hossz-nemcsökkentőek a szabályai, akkor a 3. szalagon lévő mondatforma hossza nem haladhatja meg $|u|$ -t, így ez az NTG egy LKA.

\mathcal{L}_1 és R kapcsolata

- (2) Módosítsuk egy kicsit az előző tétel konstrukcióját. Legyen $\Gamma' := \Gamma \setminus \{\triangleright, \triangleleft\}$ és $G = \langle (\Gamma \setminus \Sigma) \cup Q \times \Gamma' \cup \{S, A\}, \Sigma, P, S \rangle$.

1. $S \rightarrow \triangleright A(q_i, a) A \triangleleft \mid \triangleright (q_i, a) A \triangleleft \mid \triangleright (q_i, a) A \triangleleft \mid \triangleright (q_i, a) \triangleleft$ ($\forall a \in \Gamma'$)
2. $A \rightarrow aA \mid a$ ($\forall a \in \Gamma'$)
3. $b(q', c) \rightarrow (q, a)c$, ha $(q', b, R) \in \delta(q, a)$ ($\forall c \in \Gamma'$)
4. $(q', b) \rightarrow (q, a)$, ha $(q', b, S) \in \delta(q, a)$
5. $(q', c)b \rightarrow c(q, a)$, ha $(q', b, L) \in \delta(q, a)$ ($\forall c \in \Gamma'$)
6. $\triangleright (q_0, a) \rightarrow \triangleright a$ ($\forall a \in \Gamma'$)

Vegyük észre, hogy ez a grammatika úgynevezett hossz-nemcsökkentő grammatika, azaz minden szabály jobboldala legalább olyan hosszú, mint a baloldala. Ismeretes, hogy a hossz-nemcsökkentő grammatikák osztálya pontosan a környezetfüggő nyelveket képes generálni és ezek a grammatikák ekvivalens 1-típusú grammatikákká alakíthatóak.

\mathcal{L}_1 és R kapcsolata

Éppen a hossz-nemcsökkentő tulajdonság teljesüléseért van szükségünk a pusztán technikai $Q \times \Gamma'$ -beli jelekre. Ezekből az utolsó lépésig mindig pontos egy van a mondatformában.

1-2. generálunk egy tetszőleges elfogadó konfigurációt. Mivel A LKA, ezért u elfogadásához elég olyan hosszút, mint u . Ezután a mondatforma hossza állandó.

3-5. a konfigurációátmeneteket a G -ben fordítottan szimuláljuk.

6. A $Q \times \Gamma'$ -beli jel (azaz az állapot) eltüntetésé.

Minden $a \in \Sigma \setminus \{\triangleright, \triangleleft\}$, $w \in (\Sigma \setminus \{\triangleright, \triangleleft\})^*$ -ra vagy $a = \sqcup$, $w = \varepsilon$ -ra a levezetés hosszára vonatkozó indukcióval megmutatható, hogy valamely $x \in \Gamma'$, $\alpha, \beta \in (\Gamma')^*$ -ra $\triangleright q_0 a w \triangleleft \vdash^* \triangleright \alpha q_i x \beta \triangleleft$ akkor és csak akkor ha $S \Rightarrow^* \triangleright \alpha(q_i, x) \beta \triangleleft \Rightarrow^* \triangleright (q_0, a) w \triangleleft \Rightarrow \triangleright a w \triangleleft$. (Hasonlóan, mint az előző tétel bizonyításában.)

Tehát $L(A)$ hossz-nemcsökkentő grammatikával generálható, és így 1-es típusúval is. \square

\mathcal{L}_1 és R kapcsolata

Tétel

Ha A LKA, akkor $L(A)$ eldönthető.

Bizonyítás: A lineáris korlátoltság miatt A lehetséges konfigurációinak száma egy u bemenetre legfeljebb $m(u) = |Q| \cdot |u| \cdot |\Gamma|^{|u|}$, ahol Q az A állapothalmaza és Γ a szalagábécéje. Ha A -nak van elfogadó számítása, akkor van legfeljebb $m(u)$ hosszú elfogadó számítása is (a számítások két azonos konfiguráció közötti része kihagyható).

Működjön az M Turing gép pontosan úgy, mint A , de minden u bemenetre számolja a lépéseit $m(u)$ -ig. Ekkor állítsuk le M -et q_n -ben. Nyilván $L(M) = L(A)$ és M minden bemenetre megáll. \square

Következmény

$\mathcal{L}_1 \subseteq R$.

\mathcal{L}_1 és R kapcsolata

Tétel

$\mathcal{L}_1 \subset R$.

Bizonyítás: Az előző következmény alapján $\mathcal{L}_1 \subseteq R$.

Legyen $L_{LKA\text{-átló}} = \{\langle M \rangle \mid M \text{ LKA és } \langle M \rangle \notin L(M)\}$.

• $L_{LKA\text{-átló}}$ eldönthető. ($\Rightarrow L_{LKA\text{-átló}} \in R$)

Egy S TG ugyanis egy M LKA bemenetére menjen q_i -be, ha $\langle M \rangle \notin L(M)$ illetve menjen q_n -be, ha $\langle M \rangle \in L(M)$. Mivel $L(M)$ eldönthető ezért S mindig terminál.

• $L_{LKA\text{-átló}}$ felismerhetetlen LKA-val ($\Rightarrow L_{LKA\text{-átló}} \notin \mathcal{L}_1$)
(Cantor féle átlós módszerrel)

Tegyük fel, indirekt, hogy $L_{LKA\text{-átló}}$ -t egy S LKA felismeri.

* ha $\langle S \rangle \in L_{LKA\text{-átló}} = L(S)$, akkor S felismeri $\langle S \rangle$ -et, így $\langle S \rangle \notin L_{LKA\text{-átló}}$, ellentmondás,

* ha $\langle S \rangle \notin L_{LKA\text{-átló}} = L(S)$, akkor S nem ismeri fel $\langle S \rangle$ -et, így $\langle S \rangle \in L_{LKA\text{-átló}}$, ellentmondás. \square

R, RE és a Chomsky nyelvosztályok – Összefoglaló

A Chomsky nyelvhierarchiát és a most tanultakat összefoglalhatjuk. Az alábbi táblázatban adott sorban minden nyelvleíró eszköz egyforma erejű és erősebb a korábbi sorokban felsoroltaknál.

\mathcal{L}_3	3-típusú grammatika determinisztikus véges automata nemdeterminisztikus véges automata reguláris kifejezés
	determinisztikus veremautomata
\mathcal{L}_2	2-típusú grammatika veremautomata
\mathcal{L}_1	1-típusú grammatika lineárisan korlátolt automata
R	minden inputra megálló Turing gép
RE	Turing gép
=	nemdeterminisztikus Turing gép
\mathcal{L}_0	0-típusú grammatika

A számításelmélet alapjai II.

8. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

BONYOLULTSÁGELMÉLET

A továbbiakban eldönthető (R-beli) problémákkal foglalkozunk, ilyenkor az a kérdés, hogy valamilyen erőforrás (leggyakrabban idő vagy tár) tekintetében mennyire hatékonyan oldható meg az adott probléma.

A problémákat a legtakarékosabb megoldásuk erőforrásigénye alapján osztályozhatjuk.

A bonyolultságelmélet (angolul: complexity theory) ezen idő- és tárbonyolultsági osztályok egymáshoz való viszonyával foglalkozik.

Időbonyolultsági osztályok, $P \stackrel{?}{=} NP$

Definíció

- ▶ $TIME(f(n)) = \{L \mid L \text{ eldönthető } O(f(n)) \text{ időkorlátos determinisztikus TG-pel}\}$
- ▶ $NTIME(f(n)) = \{L \mid L \text{ eldönthető } O(f(n)) \text{ időkorlátos NTG-pel}\}$
- ▶ $P = \bigcup_{k \geq 1} TIME(n^k)$.
- ▶ $NP = \bigcup_{k \geq 1} NTIME(n^k)$.

Példa: Korábbi tételünk szerint $NTIME(f(n)) \subseteq TIME(2^{O(f(n))})$.

Észrevétel: $P \subseteq NP$, mivel a determinisztikus TG-ek tekinthetők a NTG-ek speciális esetének.

Sejtés: $P \neq NP$ (sejtjük, hogy igaz, de bizonyítani nem tudjuk).

A Clay Matematikai Intézet 2000-ben 7 probléma megoldására egyenként 1M\$-t tűzött ki (Milleniumi problémák), ezek egyike a $P \stackrel{?}{=} NP$ probléma.

NP

P -re úgy gondolunk, hogy ez tartalmazza a gyakorlatban is hatékonyan megoldható problémákat. (Nem teljesen igaz.)

Milyen problémákat tartalmaz NP?

Egy L NP-beli problémához definíció szerint létezik öt polinom időben eldöntő NTG ami gyakran a következőképpen működik: a probléma minden I bemenetére polinom időben „megsejt” (azaz nemdeterminisztikusan generál) egy T egy bizonyítékot (vagy „tanút”), majd polinom időben leellenőrzi (már determinisztikusan), hogy T alapján $I \in L$ teljesül-e. A NTG definíciója alapján elég ha a számításai közül egy produkál egy ilyen „tanút”.

Precíz tétellé is tehető, miszerint akkor és csak akkor NP-beli egy eldöntési probléma, ha minden „igen”-inputhoz megadható polinom méretű és polinom időben ellenőrizhető tanú (azaz, ami igazolja, hogy ő valóban „igen”-input).

A következőkben a P és NP bonyolultsági osztályok közötti kapcsolatot vizsgáljuk.

Polinom idejű visszavezetés

Definíció

Az $f : \Sigma^* \rightarrow \Delta^*$ szófüggvény **polinom időben kiszámítható**, ha van olyan polinom időkorlátos Turing gép, amelyik kiszámítja.

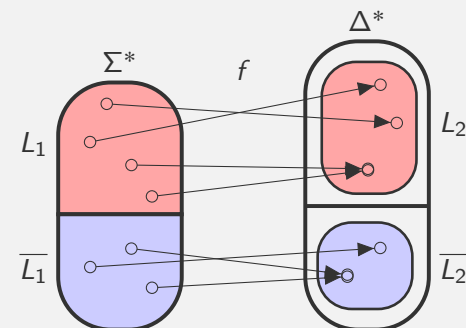
Definíció

$L_1 \subseteq \Sigma^*$ **polinom időben visszavezethető** $L_2 \subseteq \Delta^*$ -ra, ha van olyan $f : \Sigma^* \rightarrow \Delta^*$ polinom időben kiszámítható szófüggvény, hogy $w \in L_1 \Leftrightarrow f(w) \in L_2$. Jelölés: $L_1 \leq_p L_2$.

Megjegyzés: A polinom idejű visszavezetést Richard Karpról elnevezve **Karp-visszavezetésnek** vagy **Karp-redukciónak** is nevezik. Angolul: polynomial-time many-one reduction vagy Karp reduction.

Polinom idejű visszavezetés

$$L_1 \leq_p L_2$$



f **polinom időben** kiszámítható, az egész Σ^* -on értelmezett, $f(L_1) \subseteq L_2$ valamint $f(\overline{L_1}) \subseteq \overline{L_2}$.

f nem kell hogy injektív legyen és az se, hogy szürjektív.

Tétel

- Ha $L_1 \leq_p L_2$ és $L_2 \in P$, akkor $L_1 \in P$.
- Ha $L_1 \leq_p L_2$ és $L_2 \in NP$, akkor $L_1 \in NP$.

Polinom idejű visszavezetés

Bizonyítás:

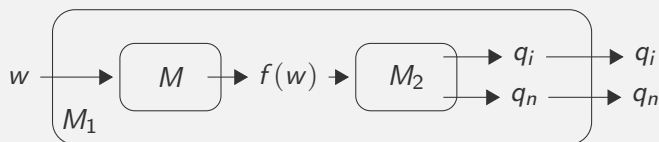
Az elsőt bizonyítjuk, a második analóg.

Legyen $L_2 \in P$ és tegyük fel, hogy $L_1 \leq_p L_2$.

Legyen M_2 az L_2 -t eldöntő, míg M a visszavezetést kiszámító TG.

Feltehetjük, hogy M $p(n)$ és M_2 $p_2(n)$ polinom idejű TG-ek.

Konstruáljuk meg M_1 -et:



- M_1 eldönti az L_1 nyelvet.
- ha w n hosszú, akkor $f(w)$ legfeljebb $n + p(n)$ hosszú lehet. (M minden lépése legfeljebb 1-gyel növelheti a hosszt.)
- M_1 tehát $p_2(n + p(n))$ időkorlátos, ami szintén polinom. \square

C-teljesség

Intuitíve, ha egy problémára visszavezetünk egy másikat, az azt jelenti, hogy az a probléma legalább olyan nehéz, mint amit visszavezettünk rá. Azaz ebben az értelemben a legnehezebb problémák azok, melyekre minden probléma visszavezethető.

Definíció

Legyen C egy bonyolultsági osztály. Egy L nyelv **C-nehéz** (a polinom idejű visszavezetésre nézve), ha minden $L' \in C$ esetén $L' \leq_p L$.

Definíció

Legyen C egy bonyolultsági osztály. Egy L nyelv **C-teljes**, ha $L \in C$ és L C-nehéz.

Ilyen bonyolultsági osztályok például, P , NP , EXP (exponenciális időben eldönthető problémák osztálya), vagy a későbbiekben tanult tárbonyolultsági osztályok.

- a konfigurációátmenet definíciója miatt bármely két sor közötti különbség belefér egy 2×3 -as "ablakba"
- T magassága akkora, hogy minden, $\leq p(n)$ lépéses átmenetet tartalmazhasson. A \sqcup -ek számát ($\Rightarrow T$ szélességét) pedig úgy, hogy az ablakok biztosan "ne eshessenek le" egyik oldalon se.

A Cook-Levin tétel bizonyítása

- φ_w ítéletváltozói $x_{i,j,s}$ alakúak, melynek jelentése: T i -ik sorának j -ik cellájában az s szimbólum van, ahol $s \in \Delta = Q \cup \Gamma \cup \{\#\}$.
- φ_w a w bemenetre M minden lehetséges legfeljebb $p(n)$ lépésű működését leírja. Felépítése: $\varphi_w = \varphi_0 \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$.
- φ_0 akkor és csak akkor legyen igaz, ha minden cellában pontosan 1 betű van:

$$\varphi_0 := \bigwedge_{\substack{1 \leq i \leq p(n)+1 \\ 1 \leq j \leq 2p(n)+3}} \left(\left(\bigvee_{s \in \Delta} x_{i,j,s} \right) \wedge \bigwedge_{s,t \in \Delta, s \neq t} (\neg x_{i,j,s} \vee \neg x_{i,j,t}) \right)$$

- φ_{start} akkor és csak akkor legyen igaz, ha T első sora a \sqcup -ekkel és $\#$ -ekkel a fent említett módon adott hosszúságúra kiegészített kezdőkonfiguráció.

$$\varphi_{\text{start}} := x_{1,1,\#} \wedge x_{1,2,\sqcup} \wedge \dots \wedge x_{1,2p(n)+2,\sqcup} \wedge x_{1,2p(n)+3,\#}$$

A Cook-Levin tétel bizonyítása

- φ_{move} akkor és csak akkor legyen igaz, ha minden ablak legális, azaz δ szerinti átmenetet ír le:

$$\varphi_{\text{move}} := \bigwedge_{\substack{1 \leq i \leq p(n) \\ 2 \leq j \leq 2p(n)+2}} \psi_{i,j},$$

$$\text{ahol } \psi_{i,j} \sim \bigvee_{\substack{(b_1, \dots, b_6) \\ \text{legális ablak}}} x_{i,j-1,b_1} \wedge x_{i,j,b_2} \wedge x_{i,j+1,b_3} \wedge x_{i+1,j-1,b_4} \wedge x_{i+1,j,b_5} \wedge x_{i+1,j+1,b_6}$$

b_1	b_2	b_3
b_4	b_5	b_6

De: $\psi_{i,j}$ sajnos nem KNF alakú!!! Ezért e helyett:

$$\psi_{i,j} := \bigwedge_{\substack{(b_1, \dots, b_6) \\ \text{illegális ablak}}} (\neg x_{i,j-1,b_1} \vee \neg x_{i,j,b_2} \vee \neg x_{i,j+1,b_3} \vee \neg x_{i+1,j-1,b_4} \vee \neg x_{i+1,j,b_5} \vee \neg x_{i+1,j+1,b_6})$$

A Cook-Levin tétel bizonyítása

- végezetül: φ_{accept} akkor és csak akkor legyen igaz, ha az utolsó sorban van q_i :

$$\varphi_{\text{accept}} = \bigvee_{j=2}^{2p(n)+2} x_{p(n)+1,j,q_i}$$

- $w \in L \Leftrightarrow$ az M NTG-nek van w -t elfogadó számítása $\Leftrightarrow T$ kitölthető úgy, hogy φ_w igaz $\Leftrightarrow \varphi_w$ kielégíthető $\Leftrightarrow \langle \varphi_w \rangle \in \text{SAT}$,
- hány literált tartalmaz a φ_w formula? Legyen $k = |\Delta|$.
- φ_0 : $(p(n)+1)(2p(n)+3)(k+k(k-1)) = O(p^2(n))$,
- φ_{start} : $2p(n)+3 = O(p(n))$,
- φ_{move} : $\leq p(n)(2p(n)+1)k^6 \cdot 6 = O(p^2(n))$,
- φ_{accept} : $2p(n)+1 = O(p(n))$,
- azaz φ_w $O(p^2(n))$ méretű, így polinom időben megkonstruálható
- tehát $w \mapsto \langle \varphi_w \rangle$ pol. idejű visszavezetés, így $L \leq_p \text{SAT}$.
- Ez tetszőleges $L \in \text{NP}$ nyelvre elmondható. Így SAT NP-nehéz. Mivel NP-beli, ezért NP-teljes is. \square

Polinom idejű visszavezetés tranzitivitása

Állítás: $L_1 \leq_p L_2, L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$.

Bizonyítás:

Tartozzon az első visszavezetéshez egy f szófüggvény és legyen M_1 egy $p_1(n)$ idejű TG ami ezt kiszámítja. A másodikhoz tartozzon egy g függvény, melyet egy M_2 TG kiszámít $p_2(n)$ időben ($p_1(n)$ és $p_2(n)$ polinomok).

$w \in L_1 \Leftrightarrow f(w) \in L_2 \Leftrightarrow g(f(w)) \in L_3$, tehát $g \circ f$ visszavezetés.

$|f(w)| \leq n + p_1(n)$, ha $|w| = n$, ugyanis M_1 legfeljebb $p_1(n)$ darab lépést lesz, lépésenként ≤ 1 -gyel nőhet a hossz.

Így $M_2 \circ M_1$ legfeljebb $h(n) := p_2(n + p_1(n))$ időben kiszámítja az $L_1 \leq L_3$ -t bizonyító $g \circ f$ -et.

Mivel $h(n)$ polinom, ezért $L_1 \leq_p L_3$. \square

További NP-teljes problémák

Az alábbi tétel alapján további nyelvek NP-teljességének bizonyítására nyílik lehetőség.

Tétel

Ha L NP-teljes, $L \leq_p L'$ és $L' \in \text{NP}$, akkor L' NP-teljes.

Bizonyítás:

Legyen $L'' \in \text{NP}$ tetszőleges. Mivel L NP-teljes, ezért $L'' \leq_p L$. Mivel a feltételek szerint $L \leq_p L'$, ezért a polinom idejű visszavezetések tranzitivitása miatt $L'' \leq_p L'$. Ebből és a 3. feltételből következik az állítás. \square

kSAT

Tehát a polinom idejű visszavezetés fogalmának segítségével további NP-beli nyelvek NP-teljessége bizonyítható. Erre nézzünk példákat.

KNF: konjunktív normálformájú nulladrendű formula

Volt: $\text{SAT} = \{ \langle \varphi \rangle \mid \varphi \text{ kielégíthető KNF} \}$ NP-teljes.

Definíció

kKNF-nek nevezünk egy olyan KNF-t, ahol minden klóz pontosan k darab páronként különböző alapú literál diszjunkciója.

Példák 4KNF:

$(\neg x_1 \vee x_3 \vee x_5 \vee \neg x_6) \wedge (\neg x_1 \vee \neg x_3 \vee x_4 \vee \neg x_6) \wedge (x_1 \vee x_2 \vee \neg x_4 \vee \neg x_6)$.

2KNF: $(\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$.

Definíció:

$k\text{SAT} = \{ \langle \varphi \rangle \mid \varphi \text{ kielégíthető } k\text{KNF} \}$

3SAT NP-teljessége

Tétel

3SAT NP-teljes.

Bizonyítás:

▶ 3SAT NP-beli. Lásd az érvelést SAT-nál.

▶ $\text{SAT} \leq_p 3\text{SAT}$

Kell $f : \varphi \mapsto \varphi'$, φ KNF, φ' 3KNF, φ' kielégíthető $\Leftrightarrow \varphi$ kielégíthető, f polinom időben kiszámolható.

$\varphi \mapsto \varphi'$:

ℓ	$\ell \vee x \vee y, \ell \vee x \vee \neg y, \ell \vee \neg x \vee y, \ell \vee \neg x \vee \neg y$
$\ell_1 \vee \ell_2$	$\ell_1 \vee \ell_2 \vee x, \ell_1 \vee \ell_2 \vee \neg x$
$\ell_1 \vee \ell_2 \vee \ell_3$	$\ell_1 \vee \ell_2 \vee \ell_3$
$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$	$\ell_1 \vee \ell_2 \vee x, \neg x \vee \ell_3 \vee \ell_4$
$\ell_1 \vee \dots \vee \ell_n \ (n \geq 5)$	$\ell_1 \vee \ell_2 \vee x_1, \neg x_1 \vee \ell_3 \vee x_2, \dots, \neg x_{n-3} \vee \ell_{n-1} \vee \ell_n$

$x, y, x_1, \dots, x_{n-3}$ új ítéletváltozók.

Minden tagra elvégezzük a fenti helyettesítést. φ' ezek konjunkciója.

3SAT NP-teljessége

ℓ	$\ell \vee x \vee y, \ell \vee x \vee \neg y, \ell \vee \neg x \vee y, \ell \vee \neg x \vee \neg y$
$\ell_1 \vee \ell_2$	$\ell_1 \vee \ell_2 \vee x, \ell_1 \vee \ell_2 \vee \neg x$
$\ell_1 \vee \ell_2 \vee \ell_3$	$\ell_1 \vee \ell_2 \vee \ell_3$
$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$	$\ell_1 \vee \ell_2 \vee x, \neg x \vee \ell_3 \vee \ell_4$
$\ell_1 \vee \dots \vee \ell_n \ (n \geq 5)$	$\ell_1 \vee \ell_2 \vee x_1, \neg x_1 \vee \ell_3 \vee x_2, \dots, \neg x_{n-3} \vee \ell_{n-1} \vee \ell_n$

Belátjuk, hogy ha egy I interpretáció kielégíti φ -t, akkor az új változók megfelelő kiértékelésével megadható egy I' φ' -t kielégítő interpretáció.

És fordítva, ha adott egy I' φ' -t kielégítő interpretáció, akkor ennek a régi változókra való I megszorítása kielégíti φ -t.

Az állításokat tagonként gondoljuk meg. Tekintsük φ egy n literálból álló tagját.

$n = 3$: nincs bizonyítani való

$n = 2$: (\Rightarrow) : ha legalább az egyik literál igaz, nyilván mindkét jobboldali tag igaz (\Leftarrow) : x és $\neg x$ közül az egyik hamis, így ha mindkét jobboldali tag igaz, akkor ℓ_1 vagy ℓ_2 igaz.

3SAT NP-teljesége

ℓ	$\ell \vee x \vee y, \ell \vee x \vee \neg y, \ell \vee \neg x \vee y, \ell \vee \neg x \vee \neg y$
$\ell_1 \vee \ell_2$	$\ell_1 \vee \ell_2 \vee x, \ell_1 \vee \ell_2 \vee \neg x$
$\ell_1 \vee \ell_2 \vee \ell_3$	$\ell_1 \vee \ell_2 \vee \ell_3$
$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$	$\ell_1 \vee \ell_2 \vee x, \neg x \vee \ell_3 \vee \ell_4$
$\ell_1 \vee \dots \vee \ell_n \ (n \geq 5)$	$\ell_1 \vee \ell_2 \vee x_1, \neg x_1 \vee \ell_3 \vee x_2, \dots, \neg x_{n-3} \vee \ell_{n-1} \vee \ell_n$

$n = 1$: (\Rightarrow): ha ℓ igaz, nyilván minden jobboldali tag igaz (\Leftarrow): " $\ell \vee$ " nélkül nem lehet mindegyik egyszerre igaz. Így ha minden jobboldali tag igaz, akkor ℓ igaz.

$n = 4$: (\Rightarrow): ha a 4 közül valamelyik literál igaz, akkor igaz az egyik jobboldali tag. x igazságértékét válasszuk úgy, hogy a másik tag is igaz legyen. (\Leftarrow): x és $\neg x$ közül az egyik hamis, így ha mindkét jobboldali tag igaz, akkor $\ell_1, \ell_2, \ell_3, \ell_4$ közül legalább egy igaz.

$n \geq 5$: (\Rightarrow): Tegyük fel, hogy ℓ_i igaz. Ekkor legyen x_1, \dots, x_{i-2} igaz x_{i-1}, \dots, x_{n-3} hamis. Átgondolható, hogy minden tagban lesz igaz literál.

3SAT NP-teljesége

ℓ	$\ell \vee x \vee y, \ell \vee x \vee \neg y, \ell \vee \neg x \vee y, \ell \vee \neg x \vee \neg y$
$\ell_1 \vee \ell_2$	$\ell_1 \vee \ell_2 \vee x, \ell_1 \vee \ell_2 \vee \neg x$
$\ell_1 \vee \ell_2 \vee \ell_3$	$\ell_1 \vee \ell_2 \vee \ell_3$
$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$	$\ell_1 \vee \ell_2 \vee x, \neg x \vee \ell_3 \vee \ell_4$
$\ell_1 \vee \dots \vee \ell_n \ (n \geq 5)$	$\ell_1 \vee \ell_2 \vee x_1, \neg x_1 \vee \ell_3 \vee x_2, \dots, \neg x_{n-3} \vee \ell_{n-1} \vee \ell_n$

$n \geq 5$: (\Leftarrow): Tegyük fel, hogy jobboldalon minden tag igaz és indirekt tegyük fel, hogy ℓ_1, \dots, ℓ_n hamis. Ekkor az új tagokon balról jobbra végighaladva sorra kapjuk, hogy x_1, \dots, x_{n-3} igaz kell legyen, de ekkor az utolsó tag mégiscsak hamis, ellentmondás.

Tehát φ kielégíthető $\Leftrightarrow \varphi'$ kielégíthető. φ' φ -ből polinom időben elkészíthető és mérete az eredeti méret polinomja, tehát $\text{SAT} \leq_p 3\text{SAT}$. \square

2SAT P-beli

Tétel

$2\text{SAT} \in \text{P}$.

Bizonyítás Legyen φ egy x_1, \dots, x_n változót tartalmazó 2KNF formula m klózzal.

Konstruálunk egy G_φ $2n$ csúcsú irányított gráfot. G_φ csúcsai legyenek a $2n$ literál és minden $\ell_i \vee \ell_j$ klóz esetén adjuk hozzá a $(\neg \ell_i, \ell_j)$ és a $(\neg \ell_j, \ell_i)$ irányított éleket a gráf élhalmazához. Ezt az motiválja, hogy $\ell_i \vee \ell_j \sim_0 \neg \ell_i \rightarrow \ell_j \sim_0 \neg \ell_j \rightarrow \ell_i$.

Be fogjuk látni a következő állítást:

Állítás: φ akkor és csak akkor kielégíthető, ha G_φ egyetlen erősen összefüggő komponense se tartalmaz komplement literálpárt.

(Emlékeztető: egy irányított gráf erősen összefüggő, ha bármely két csúcsa között van mindkét irányban irányított út. Minden irányított gráf csúcshalmaza erősen összefüggő komponensekre particionálható.)

2SAT P-beli

Az állításból következik a tétel, hiszen ismeretes (lásd pl. Algoritmusok és Adatszerkezetek II.), hogy egy $G = (V, E)$ gráf erősen összefüggő komponensei $O(|V| + |E|)$ időben meghatározhatóak, és most $|V| = 2n, |E| = 2m$, azaz az algoritmus $\max\{n, m\}$ -ben polinomiális.

Az állítás bizonyítása: Vegyük észre, hogy ha egy I interpretáció kielégíti φ -t, akkor ha egy literál igaz I -ben, akkor minden belőle kiinduló él végpontja is igaz. Így az erősen összefüggő komponensek literáljainak ugyanaz az igazságértéke.

Ebből azonnal következik az állítás egyik iránya, hiszen ha G_φ valamelyik erősen összefüggő komponense tartalmaz komplement literálpárt, akkor ezen literálpárnak ugyanaz lenne az igazságértéke, ami lehetetlen. Így φ kielégíthetetlen.

2SAT P-beli

Az állítás másik irányához meg kell adnunk egy φ -t kielégítő I interpretációt ha G_φ erősen összefüggő komponensei nem tartalmaznak komplement literálpárt.

Legyen x_i tetszőleges ítéletváltozó. A feltétel szerint vagy x_i -ből $\neg x_i$ -be vagy $\neg x_i$ -ből x_i -be nincs irányított út. Ha egyik sincs, akkor adjuk hozzá G_φ -hez az $e = (x_i, \neg x_i)$ élt.

Ettől nem sérül a feltétel, hiszen ha ezzel valamely j -re x_j és $\neg x_j$ egy komponensbe kerülne, akkor ide kerülne az e él is és így x_i és $\neg x_i$ is. Azonban ez nem lehet, hiszen nincs $\neg x_i$ -ből x_i -be út.

Ezt addig folytatjuk, amíg nem lesz G_φ -ben minden komplement literálpár között pontosan az egyik irányba út. Minden i -re

$$I(x_i) := \begin{cases} i, & \text{ha } G_\varphi\text{-ben van } \neg x_i\text{-ből } x_i\text{-be irányított út} \\ h & \text{ha } G_\varphi\text{-ben van } x_i\text{-ből } \neg x_i\text{-be irányított út} \end{cases}$$

Így minden hamis literálból van a komplement párjába irányított út.

2SAT P-beli

Ez az I interpretáció minden klózt igazra értékeli.

Ugyanis indirekt tegyük fel, hogy φ -nek az $\ell_i \vee \ell_j$ klóza I -ben hamis. Ekkor ℓ_i és ℓ_j is hamis. Tehát az I definíciója utáni észrevétel miatt

(1) van irányított út ℓ_i -ből $\neg \ell_i$ -be és ℓ_j -ből $\neg \ell_j$ -be.

Másrészt G_φ definíciója miatt

(2) $(\neg \ell_i, \ell_j)$ és $(\neg \ell_j, \ell_i)$ éle G_φ -nek.

(1)-ből és (2)-ből következik, hogy ℓ_i és $\neg \ell_i$ G_φ -nek ugyanabban az erősen összefüggő komponensében van, ami feltételünk szerint nem lehet.

Ezzel az állítás és így a tétel bizonyítását is befejeztük. \square

HORNSAT P-beli

Definíció

Horn formula: olyan KNF, amelynek minden tagja legfeljebb egy pozitív (azaz negátlan) literált tartalmaz.

Példa: $(\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_4 \vee \neg x_6) \wedge (\neg x_2 \vee \neg x_4 \vee \neg x_6)$

Definíció

$\text{HORNSAT} = \{ \langle \varphi \rangle \mid \varphi \text{ kielégíthető Horn formula} \}$

HORNSAT P-beli

Tétel

$\text{HORNSAT} \in P$.

Bizonyítás: Legyen φ egy x_1, \dots, x_n változókat tartalmazó Horn formula m klózzal. Feltehető, hogy az egyes klózokban szereplő változók páronként különbözőek. 3 fajta klóz fordulhat elő:

1. x_k ($1 \leq k \leq n$),
2. $x_k \vee \neg x_{i_1} \vee \dots \vee \neg x_{i_j}$ ($j \geq 1, 1 \leq k, i_1, \dots, i_j \leq n$),
3. $\neg x_{i_1} \vee \dots \vee \neg x_{i_j}$ ($j \geq 1, 1 \leq i_1, \dots, i_j \leq n$).

Definiáljuk az I_{\min} interpretációt a következő algoritmus:

- ▶ Kezdetben legyen minden ítéletváltozó hamisra értékelve.
- ▶ Az 1. típusú klózokban szereplő változókat állítsuk át igazra.
- ▶ Amíg van olyan 2. típusú, $x_k \vee \neg x_{i_1} \vee \dots \vee \neg x_{i_j}$ klóz, amelyre $I_{\min}(x_k) = h$ és minden $\ell \in \{i_1, \dots, i_j\}$ esetén $I_{\min}(x_\ell) = i$ addig csináljuk a következőt. Billentsük át x_k igazságértékét hamisról igazra.

HORNSAT P-beli

Állítás: φ kielégíthető $\Leftrightarrow I_{\min} \models_0 \varphi$.

Teljes indukcióval könnyen látható ugyanis, hogy a fenti algoritmus minden iterációja után az aktuálisan igaz változókat minden változókiértékelésnek igazra kell értékelnie. Így minden φ -t kielégíthető interpretációnak igazra kell értékelnie minden egyes I_{\min} által igazra értékelt változót.

Tehát I_{\min} az az interpretáció, amely a lehető legkevesebb 3. típusú klózt értékeli hamisra.

Ha N a formula hossza, akkor I_{\min} $O(mN) = O(N^2)$ időben kiszámítható, majd $O(N)$ időben ellenőrizhető, hogy minden klózt kielégít-e. \square

A számításelmélet alapjai II.

9. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

3 színezhetőség

Definíció

Legyen $k \geq 1$ egész szám. Egy (irányítatlan) gráf **k -színezhető**, ha kiszínezhetők a csúcsai k színnel úgy, hogy bármely két szomszédos csúcsnak a színe különböző.

Formálisan: $G = (V, E)$ k -színezhető, ha $\exists f : V \rightarrow \{1, \dots, k\}$ leképezés, melyre $\forall x, y \in V : f(x) = f(y) \Rightarrow \{x, y\} \notin E$.

1. Példa: Jelölje K_n a teljes n csúcsú gráfot. Ekkor K_n k -színezhető minden $k \geq n$ -re, de nem $(n-1)$ -színezhető (semelyik 2 csúcs sem lehet azonos színű).

2. Példa: Egy 5 csúcsú kör 3-színezhető, de nem 2-színezhető.



3 színezhetőség

$k\text{SZÍNEZÉS} := \{ \langle G \rangle \mid G \text{ } k\text{-színezhető} \}$

Itt $\langle G \rangle$ a G gráf kódját jelöli $\{0, 1\}$ felett, mondjuk a szomszédsági mátrixa sorfolytonosan.

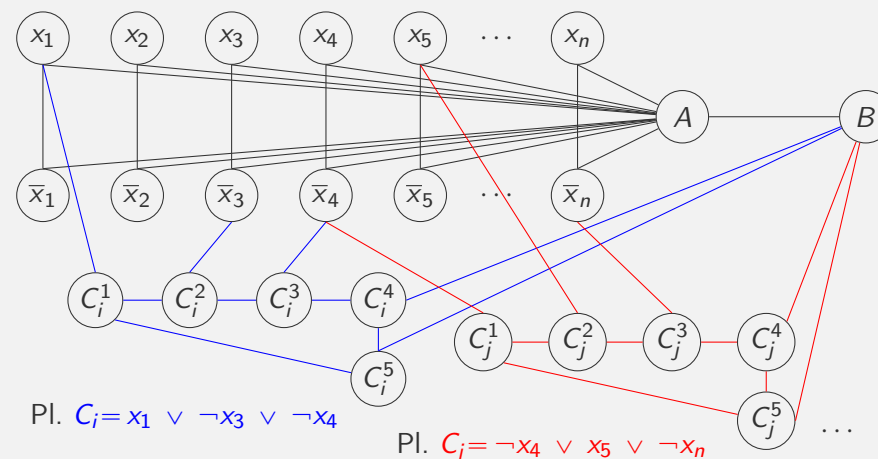
Tétel

3SZÍNEZÉS NP-teljes.

- $k\text{SZÍNEZÉS}$ NP-beli: egy $\langle G \rangle$ inputra az M NTG egy számítási ága állítson elő egy $f : V(G) \rightarrow \{1, \dots, k\}$ k -színezést. Mind egy konkrét k -színezés előállítása, mind pedig a konkrét színezés helyességének ellenőrzése polinom időben megtehető. M egy számítása végződjön q_i -ben, ha az egy jó k -színezés. G k -színezhető $\Leftrightarrow \exists$ jó k -színezése $\Leftrightarrow M$ elfogadja $\langle G \rangle$ -t.
- $3\text{SAT} \leq_p 3\text{SZÍNEZÉS}$: elegendő minden φ 3KNF formulához polinom időben elkészíteni egy G_φ gráfot úgy, hogy φ kielégíthető $\Leftrightarrow G_\varphi$ 3-színezhető.

3 színezhetőség

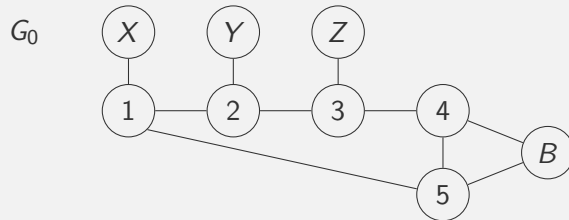
Legyenek x_1, \dots, x_n a φ -ben előforduló ítéletváltozók. Továbbá $\varphi = C_1 \wedge \dots \wedge C_m$, azaz C_1, \dots, C_m φ pontosan 3 literálból álló klózzai. G_φ konstrukciója:



Minden klózhoz tartozik egy ötszög a fenti módon.

3 színezhetőség

Lemma: Legyen G_0 az alábbi gráf és tegyük fel, hogy az X, Y, Z, B csúcsokat 2 színnel kiszíneztük. Akkor és csak akkor létezik ehhez a parciális színezéshez az egész G_0 -ra kiterjeszthető 3-színezés, ha X, Y, Z, B nem mind egyszínű.



A lemma bizonyítása:

- Ha X, Y, Z, B egyszínű, akkor a maradék 2 színnel kéne az ötszöget kiszínezni, amit nem lehet.

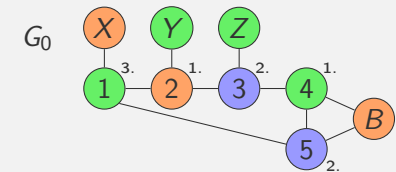
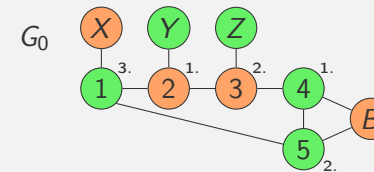
3 színezhetőség

- Ha X, Y, Z, B nem egyszínű, akkor megadható egy színezés.
 - 1. lépés:** első körben 2 színt használunk, 1,2,3,4,5-öt színezzük az $\{X, Y, Z, B\}$ -beli szomszédjával ellentétes színűre. Ez persze még nem jó, lehetnek azonos színű szomszédok.
 - 2. lépés:** bevetjük a 3. színt: ha 1,2,3,4,5 között van valahány egymás utáni azonos színű csúcs (az óramutató járása szerint és ciklikusan), akkor ezen egymás utáni azonos színű csúcsok közül minden párosadikat színezzük át a 3. színre.

Példa:

1. lépés utáni színezés

2. lépés utáni színezés



3 színezhetőség

A visszavezetés bizonyítása:

- Tegyük fel hogy φ kielégíthető, ekkor meg kell adnunk G_φ egy 3-színezését. Legyenek a színek piros, zöld és kék. Ha x_i igaz, akkor legyen az x_i csúcs zöld, az \bar{x}_i csúcs piros. Ha hamis, akkor épp fordítva. A legyen kék és B legyen piros. Mivel minden klóz ki van elégítve, így minden ötszöghöz van zöld (az igaz literál) és piros szomszéd (B) is, így a lemma miatt a színezés minden ötszögre kiterjeszthető.
- Tegyük fel most, hogy G_φ jól ki van színezve 3 színnel. Feltehető, hogy A kék. Mivel $x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n$ mind A szomszédai, így egyikük se lehet kék. Továbbá az (x_i, \bar{x}_i) párok össze vannak kötve, így minden párban pontosan egy piros és egy zöld csúcs van. Ámnfth. B piros (a zöld eset analóg). Mivel az ötszögek ki vannak színezve, ezért a lemma miatt minden ötszögnek van zöld szomszédja. Az " $x_i := \text{igaz} \Leftrightarrow x_i$ csúcs zöld" interpretáció tehát kielégíti φ -t.

2 színezhetőség

Beláttuk tehát, hogy $\varphi \mapsto G_\varphi$ visszavezetés. Mivel φ ismeretében G_φ mérete és elkészítésének ideje az input (φ) méretének polinomja, ezért a visszavezetés polinom idejű.

Mivel 3SAT NP-teljes, korábbi tételünk miatt 3SZÍNEZÉS is az. \square

A 2-színezhető gráfok éppen a páros gráfok (a páros gráf két csúccsoztálya a két színosztály). Lineáris időben eldönthető, hogy egy gráf páros-e:

Tétel

2SZÍNEZÉS $\in P$

Bizonyítás:

Indítsunk egy szélességi bejárást G egy tetszőleges x csúcsából. Ez az x -szel egy komponensben lévő csúcsokat szintekre particionálja az x -től való távolságuk (legrövidebb út hossza) szerint. Ha a gráf nem összefüggő, akkor minden komponensre végezzük ezt el. Az algoritmus $O(|V| + |E|)$ idejű.

2 színezhetőség

Állítás: G 2-színezhető \Leftrightarrow a szélességi bejárás G semelyik komponensében se talál élt két azonos szintű csúcs között.

Az állítás bizonyítása:

(\Leftarrow) Legyenek a páros szinteken lévő csúcsok kékek, a páratlan szinteken lévők pirosak. Ekkor nincs két azonos színű csúcs között él. A felétel és a színezés miatt ilyen csak úgy lehetne ha azonos komponensben legalább 2 szintkülönbségű csúcsokról lenne szó. Azonban irányítatlan gráfok szélességi bejárása során ilyen nincs.

(\Rightarrow) Ha van két azonos szintű x és y csúcs között él, akkor legyen a k szinttel feljebb levő z az a csúcs, ami x és y legnagyobb szintszámú (azaz legelső) közös őse a szélességi feszítőfában. Ekkor kaptunk egy x, y, z -t tartalmazó $2k + 1$ hosszú kört, hiszen z elsőse miatt a szélességi feszítőfában a $z \rightsquigarrow x$ és $z \rightsquigarrow y$ utaknak z -n kívül nem lehet közös pontja. Így G nem 2-színezhető, mivel páratlan hosszú körök nyilván nem 2-színezhetők.

Az állítás feltételét a szélességi bejárás menet közben ellenőrizheti. \square

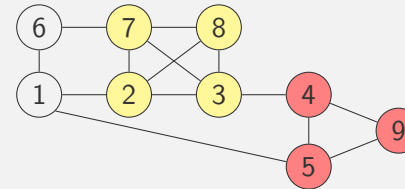
Klikk

Definíció

Egy G egyszerű, irányítatlan gráf egy teljes részgráfját **klikknek** nevezzük.

$\text{KLIKK} := \{ \langle G, k \rangle \mid G\text{-nek van } k \text{ méretű klikkje} \}$

Példa:



$\{2, 3, 7, 8\}$ és $\{4, 5, 9\}$ klikk. $\{1, 2, 6, 7\}$ nem klikk.

Észrevétel: Ha G -nek van k méretű klikkje, akkor bármely kisebb k -ra is van.

Független ponthalmaz

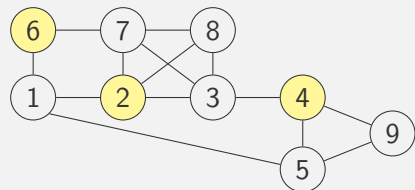
Definíció

Egy G egyszerű, irányítatlan gráf egy üres részgráfját **független ponthalmaznak** mondjuk.

$\text{FÜGGETLEN PONTALMAZ} :=$

$\{ \langle G, k \rangle \mid G\text{-nek van } k \text{ méretű független ponthalmaza} \}$

Példa:



$\{2, 6, 4\}$ független. $\{1, 7, 3, 9\}$ nem független a $\{3, 7\}$ él miatt.

Észrevétel: Ha G -nek van k méretű független ponthalmaza, akkor bármely kisebb k -ra is van.

Lefogó ponthalmaz

Definíció

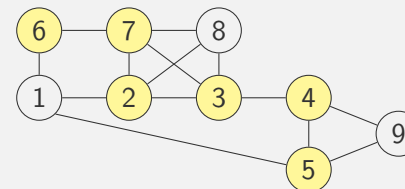
Legyen $S \subseteq V(G)$ és $E \in E(G)$. Ha $S \cap E \neq \emptyset$, akkor a csúcshalmaz **lefogja** E -t. Ha S minden $E \in E(G)$ élt lefog, akkor S egy **lefogó ponthalmaz**.

Megjegyzés: A fenti fogalom **csúcshalmaz** néven is ismeretes.

$\text{LEFOGÓ PONTALMAZ} :=$

$\{ \langle G, k \rangle \mid G\text{-nek van } k \text{ méretű lefogó ponthalmaza} \}$

Példa:



$\{2, 3, 4, 5, 6, 7\}$
lefogó ponthalmaz.

Észrevétel: Ha G -nek van k méretű lefogó ponthalmaza, akkor bármely $k \leq k' \leq |V(G)|$ -re is van.

FÜGGETLEN PONTALMAZ

Tétel

KLIKK, FÜGGETLEN PONTALMAZ, LEFOGÓ PONTALMAZ NP-teljes.

- ▶ Egy NTG egy számítási ágán vizsgálja meg a csúcsoknak egy konkrét, k elemű részalmazát. Egy k elemű pontalmaz előállítása illetve annak ellenőrzése, hogy ez egy klikk/független pontalmaz/lefogó pontalmaz az input méretének polinomiális függvénye. Tehát mindhárom nyelv NP-ben van.

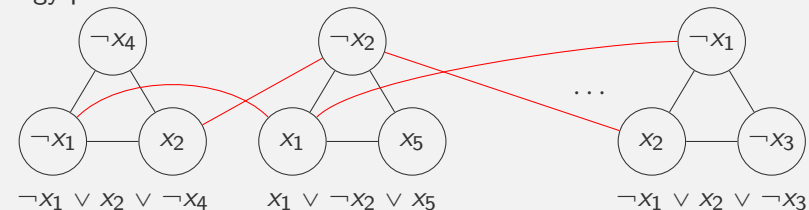
- ▶ $3SAT \leq_p FÜGGETLEN CSÚCSALMAZ$

Kell: $f : \varphi \mapsto (G_\varphi, k)$, φ 3KNF, G_φ -ben van k független csúcs akkor és csak akkor ha φ kielégíthető.

(G_φ, k) konstrukciója: minden egyes $\ell_1 \vee \ell_2 \vee \ell_3$ klózhoz vegyünk fel egy a többletől diszjunkt háromszöget, a csúcsokhoz rendeljük hozzá a literálokat. Így m darab klóz esetén $3m$ csúcsot kapunk. Kössük össze éllel ezen felül a komplementens párokat is. $k := m$.

FÜGGETLEN PONTALMAZ

Egy példa:



* Ha φ kielégíthető, akkor minden klózban van kielégített literál, válasszunk klózonként egyet, ezeknek megfelelő csúcsok m elemű független csúcsalmazt alkotnak.

* Ha G_φ -ben van m független csúcs, akkor ez csak úgy lehet, ha háromszögenként 1 van. Vegyünk egy ilyen, ezen csúcsoknak megfelelő literálok között nem lehet komplementens pár, hiszen azok össze vannak kötve. Így a független halmaznak megfelelő, (esetleg csak parciális) interpretáció kielégít minden klózt. Ha nincs minden változó kiértékelve, egészítsük ki tetszőlegesen egy teljes interpretációvá.

KLIKK, LEFOGÓ PONTALMAZ

- ▶ $FÜGGETLEN PONTALMAZ \leq_p KLIKK$

$$f : (G, k) \mapsto (\bar{G}, k)$$

Ez egy jó visszavezetés, hiszen ami G -ben klikk az \bar{G} -ben független pontalmaz és fordítva, ami G -ben független pontalmaz az \bar{G} -ben klikk.

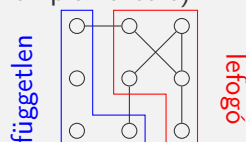
- ▶ $FÜGGETLEN PONTALMAZ \leq_p LEFOGÓ PONTALMAZ$

$$f : (G, k) \mapsto (G, |V(G)| - k)$$

Ha G -ben van k méretű F független pontalmaz, akkor van $|V(G)| - k$ méretű lefogó pontalmaz (F komplementere).

Ha G -ben van $|V(G)| - k$ méretű L lefogó pontalmaz, akkor van k méretű független pontalmaz (L komplementere).

Mindkét visszavezetés polinom időben kiszámítható.



Lefogó pontalmaz hipergráfokban

Definíció

\mathcal{S} egy **hipergráf** (vagy halmazrendszer), ha $\mathcal{S} = \{A_1, \dots, A_n\}$, ahol $A_i \subseteq U$, $(1 \leq i \leq n)$ valamely U alaphalmazra. $H \subseteq U$ egy **hipergráf lefogó pontalmaz**, ha $\forall 1 \leq i \leq n : H \cap A_i \neq \emptyset$.

HIPERGRÁF LEFOGÓ PONTALMAZ:=

$\{\langle \mathcal{S}, k \rangle \mid \mathcal{S} \text{ egy hipergráf és van } k \text{ elemű } \mathcal{S}\text{-et lefogó pontalmaz}\}$.

Tétel

HIPERGRÁF LEFOGÓ PONTALMAZ NP-teljes.

Bizonyítás: A nyelv NP-beli, hiszen polinom időben előállítható U egy tetszőleges H részalmazára és szintén polinom időben ellenőrizhető, hogy H minden \mathcal{S} -beli halmazt metsz-e.

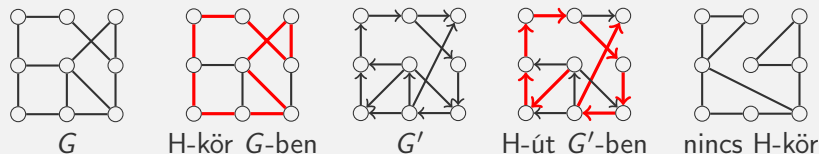
LEFOGÓ PONTALMAZ a HIPERGRÁF LEFOGÓ PONTALMAZ speciális esete. Minden gráf hipergráf is egyben, a megfeleltetés $U = V(G)$, $\mathcal{S} = E(G)$. k ugyanaz, mivel a lefogó pontalmaz szintén speciális esete a hipergráf lefogó pontalmaznak. \square

Írányítatlan/írányított Hamilton út/kör

Definíció

Adott egy G gráf. Egy a G összes csúcsát pontosan egyszer tartalmazó utat **Hamilton útnak**, egy a G összes csúcsát pontosan egyszer tartalmazó kört **Hamilton körnek** nevezünk. Ha a gráf irányított, a Hamilton útnak/körnek irányítottnak kell lennie.

Rövidítés: H-út/ H-kör: Hamilton út/ Hamilton kör.



$H\dot{U} = \{ \langle G, s, t \rangle \mid \text{van a } G \text{ irányított gráfban } s\text{-ből } t\text{-be H-út} \}$.

$IH\dot{U} = \{ \langle G, s, t \rangle \mid \text{van a } G \text{ irányítatlan gráfban } s \text{ és } t \text{ végpontokkal H-út} \}$.

$IHK = \{ \langle G \rangle \mid \text{van a } G \text{ irányítatlan gráfban H-kör} \}$.

Írányított $s \rightsquigarrow t$ Hamilton út NP teljessége

Tétel

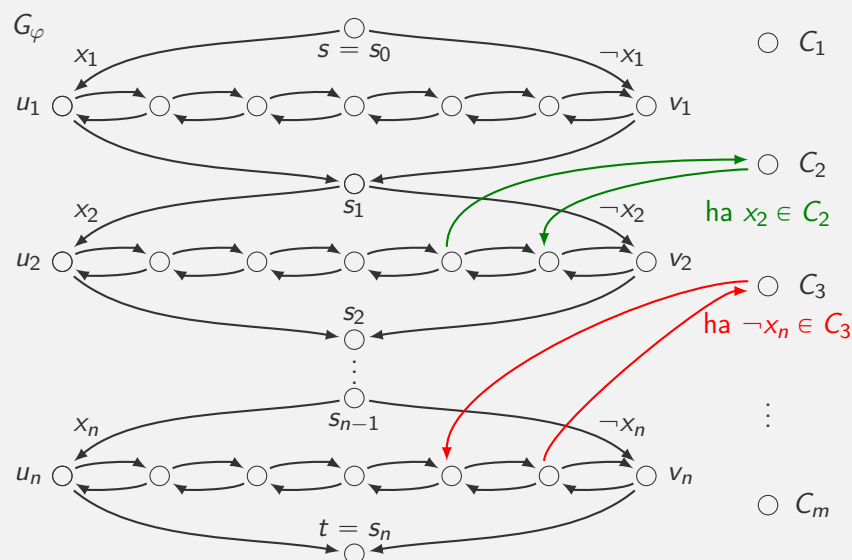
HÚ NP-teljes

Bizonyítás: NP-beli, hiszen polinom időben előállítható n darab csúcs egy P felsorolása. P -ről polinom időben ellenőrizhető, hogy a csúcsok egy permutációja-e és hogy tényleg H-út-e.

$SAT \leq_p H\dot{U}$. Elég bármely φ KNF-hez konstruálni (G_φ, s, t) -t azzal a tulajdonsággal, hogy φ kielégíthető \Leftrightarrow a G_φ -ben van s -ből t -be H-út.

Legyenek x_1, \dots, x_n a φ -ben előforduló ítéletváltozók és C_1, \dots, C_m φ klózai.

Írányított $s \rightsquigarrow t$ Hamilton út NP teljessége



Írányított $s \rightsquigarrow t$ Hamilton út NP teljessége

G_φ konstrukciója

- ▶ $\forall 1 \leq i \leq n : (s_{i-1}, u_i), (s_{i-1}, v_i), (u_i, s_i), (v_i, s_i) \in E(G_\varphi)$
- ▶ $s := s_0, t := s_n$
- ▶ $\forall 1 \leq i \leq n$ -re u_i és v_i között $3m - 1$ belső pontú kétirányú út $w_{i,1}, \dots, w_{i,3m-1}$.
- ▶ Minden $w_{i,k}$ legfeljebb egy C_j -vel lehet összekötve, ha $3 \mid k$, akkor eggyel se.
- ▶ Ha $x_i \in C_j$, akkor $(w_{i,3j-2}, C_j)$ és $(C_j, w_{i,3j-1}) \in E(G_\varphi)$. (pozitív bekötés)
- ▶ Ha $\neg x_i \in C_j$, akkor $(w_{i,3j-1}, C_j)$ és $(C_j, w_{i,3j-2}) \in E(G_\varphi)$. (negatív bekötés)

Az $u_i v_i$ út **pozitív bejárása**: $u_i \rightsquigarrow v_i$.

Az $u_i v_i$ út **negatív bejárása**: $u_i \rightsquigarrow v_i$.

Irányított $s \rightsquigarrow t$ Hamilton út NP teljessége

- ▶ Egy $s \rightsquigarrow t$ H-út $\forall 1 \leq i \leq n$ -re az (s_{i-1}, u_i) és (s_{i-1}, v_i) közül pontosan egyiket tartalmazza, előbbi esetben az $u_i v_i$ utat pozitív, utóbbi esetben negatív irányban járja be.
- ▶ Egy $s \rightsquigarrow t$ H-út minden C_j -t pontosan egyszer köt be. Az $u_i v_i$ út pozitív bejárása esetén csak pozitív, negatív bejárása esetén csak negatív bekötés lehetséges.
- ▶ Ha van H-út, akkor az $u_i v_i$ utak pozitív/negatív bejárása meghatároz egy I változókiértékelést. A C_j klóz bekötése mutat C_j -ben egy igaz literált ($\forall 1 \leq j \leq m$). Tehát I kielégíti φ -t.
- ▶ Fordítva, ha φ kielégíthető, válasszunk egy φ -t igazra kiértékelő I interpretációt és φ minden klózához egy I -ben igaz literált. Az $u_i v_i$ utat $I(x_i) = i$ esetén pozitívan, $I(x_i) = h$ esetén negatívan járjuk be. Ha a kiválasztott literálokhoz rendre bekötjük a C_j csúcsokat H-utat kapunk.

G_φ polinom időben megkonstruálható így $\text{SAT} \leq_p \text{HÚ}$, azaz HÚ NP-nehéz, de láttuk, hogy NP-beli, így NP-teljes is. \square

Irányítatlan $s \rightsquigarrow t$ Hamilton út NP teljessége

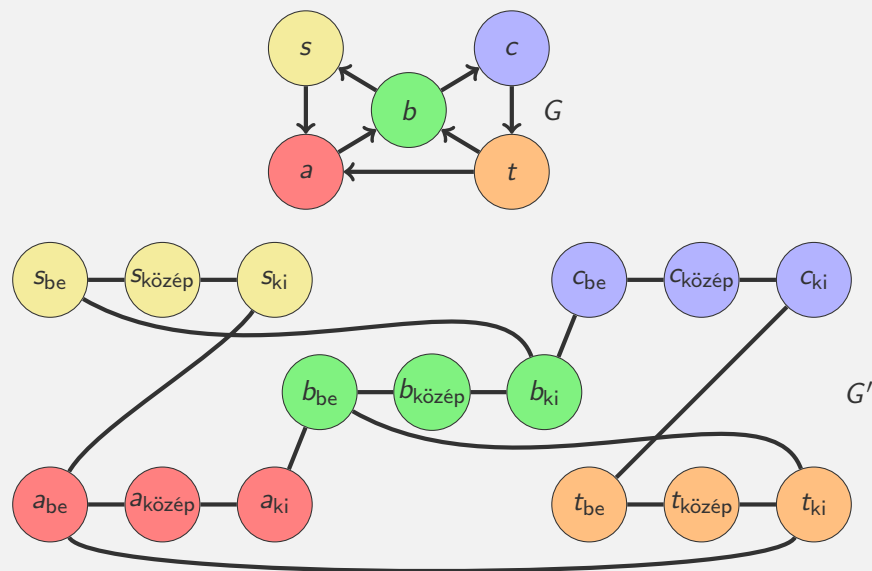
Megjegyzés: IHÚ és IHK NP-belisége az előzőekhez hasonlóan adódik.

Tétel

IHÚ NP-teljes

Bizonyítás: $\text{HÚ} \leq_p \text{IHÚ}$. Adott G, s, t , ahol G irányított. Kell G', s', t' , ahol G' irányítatlan és akkor és csak akkor van G -ben s -ből t -be H-út, ha G' -ben van s' -ből t' -be.
 G minden v csúcsának feleljen meg G' -ben 3 csúcs v_{be} , $v_{közép}$ és v_{ki} . és G' élei közé vegyük be a $\{v_{be}, v_{közép}\}$ és $\{v_{közép}, v_{ki}\}$ éleket. Továbbá minden $E = (u, v)$ G -beli él estén adjuk hozzá $E(G')$ -höz $\{u_{ki}, v_{be}\}$ -t. $s' := s_{be}$, $t' := t_{ki}$.

Irányítatlan $s \rightsquigarrow t$ Hamilton út NP teljessége



Irányítatlan $s \rightsquigarrow t$ -Hamilton út NP teljessége

Könnyen meggondolható, hogy ez egy polinomiális visszavezetés:

- ▶ G' mérete G méretének polinomja és G' G -ből nyilván polinom időben előállítható.
- ▶ ha G -ben van egy $P : s \rightsquigarrow t$ irányított H-út, akkor G' konstrukciója miatt a következő G' -beli csúcssorozat H-út G' -ben: P szerint haladva minden v csúcsot sorra a v_{be} , $v_{közép}$, v_{ki} csúcsokkal helyettesítsük.
- ▶ ha G' -ben van egy $P' : s' \rightsquigarrow t'$ H-út, akkor P' -ben minden v -re v_{be} , $v_{közép}$, v_{ki} egymást követő csúcsok, hiszen $v_{közép}$ 2-fokú csúcs és máskülönben nem lehetne rajta P' -n. Ezen csúcshármasokat $\{u_{ki}, v_{be}\}$ típusú élek kötik össze, melyekhez definíció szerint van $u \rightarrow v$ él G -ben. Tehát ha minden v -re a P' -ben egymást követő v_{be} , $v_{közép}$, v_{ki} csúcshármas v -vel helyettesítjük egy G -beli irányított H-utat kapunk. \square

Irányítatlan Hamilton kör NP teljessége

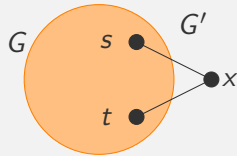
Tétel

IHK NP-teljes

Bizonyítás: $IH\dot{U} \leq_p IHK$. Adott G, s, t . G' konstrukciója: adjunk hozzá G csúcshalmazához egy új x csúcsot és élhalmazához két új élt $\{s, x\}$ -et és $\{t, x\}$ -t.

Könnyen meggondolható, hogy ez egy polinomiális visszavezetés:

- ▶ G' G -ből nyilván polinom időben előállítható.
- ▶ ha G -ben van $P : s \rightsquigarrow t$ H-út, akkor G' -ben van H-kör: egészítsük ki P -t az $\{s, x\}$ és $\{t, x\}$ élekkel.
- ▶ ha G' -ben van C H-kör, akkor G -ben van $s \rightsquigarrow t$ H-út: C -nek tartalmaznia kell az $\{s, x\}$ és $\{t, x\}$ éleket, mivel x 2-fokú. C -ből $\{s, x\}$ -et, $\{t, x\}$ -et és x -et elhagyva egy G -beli $s \rightsquigarrow t$ H-út marad.



□

Az utazóügynök probléma

Számítási (optimalizálási) verzió: Adott egy G élsúlyozott irányítatlan gráf nemnegatív élsúlyokkal. Határozzuk meg a legkisebb összsúlyú H-kört (ha van).

Eldöntési verzió:

$TSP = \{ \langle G, K \rangle \mid G\text{-ben van } \leq K \text{ súlyú H-kör} \}$.

Tétel

TSP NP-teljes

Bizonyítás: $TSP \in NP$, hasonló érvek miatt, mint HÚ, az összköltség feltétel is polinom időben ellenőrizhető.

$IHK \leq_p TSP$. Adott egy G gráf. G függvényében konstruálunk egy G' élsúlyozott gráfot és megadunk egy K számot. $G' := G$, minden élsúly legyen 1 és $K := |V|$. Könnyen látható, hogy G -ben van H-kör $\Leftrightarrow G'$ -ben van legfeljebb K összsúlyú H-kör.

A visszavezetés nyilvánvalóan polinom idejű.

□

A számításelmélet alapjai II.

10. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Lineáris diophantoszi egyenlőtlenségrendszer

DIOPHANTOSZI EGYENLŐTLÉNSÉGRENDSZER =
 $\{\langle \mathbf{A}, \mathbf{b} \rangle \mid \mathbf{Ax} \leq \mathbf{b} \text{ egészgyűthetős egyenlőtlenségrendszernek van egész megoldása} \}.$

Tétel

DIOPHANTOSZI EGYENLŐTLÉNSÉGRENDSZER NP-nehéz.

Bizonyítás: A 3SAT problémát vezetjük rá vissza polinom időben. Legyen φ egy 3KNF, változói x_1, \dots, x_n . Vegyük fel a $0 \leq x_i \leq 1$ egyenlőtlenségeket. Továbbá, ha $L_1 \vee L_2 \vee L_3$ φ egy klóza, akkor vegyük fel a $t_1 + t_2 + t_3 \geq 1$ egyenlőtlenséget, ahol $t_i = x_j$, ha $L_i = x_j$ és $t_i = 1 - x_j$, ha $L_i = \neg x_j$ ($i = 1, 2, 3$).

Könnyen látható, hogy az így kapott lineáris diophantoszi egyenlőtlenségrendszernek akkor és csak akkor van egész megoldása, ha φ kielégíthető. (Az 1 az igaznak, a 0 a hamisnak felel meg.) \square

Lineáris diophantoszi egyenlőtlenségrendszer

1. Megjegyzés: Az is igaz, hogy DIOPHANTOSZI EGYENLŐTLÉNSÉGRENDSZER NP-teljes. Az NP-beliség bizonyításához szükségünk lenne egy felső korlátra egy megoldás méretére vonatkozóan. Adható ilyen polinomiális korlát (de ez egyáltalán nem nyilvánvaló állítás, hiszen negatívák is lehetnek az egyűthetők).

2. Megjegyzés: Tetszőleges (nem feltétlen lineáris) diophantoszi egyenletek megoldhatósága (Hilbert 10. problémája) eldönthetetlen (Jurij Matijaszevics, 1970). Ez nem meglepő, hiszen a problémaosztály tartalmazza például a Nagy Fermat sejtést/ Wiles tételt is ($a^n + b^n = c^n$ -nek nincs pozitív egész megoldása, ha $n > 2$ egész).

Részletösszeg probléma

RÉSZLETÖSSZEG = $\{\langle S, K \rangle \mid S \text{ egész számok egy halmaza, } K \in \mathbb{Z}, \text{ van } S\text{-nek egy olyan } S' \text{ részhalmaza, hogy az } S'\text{-beli számok összege } K \}.$

Példa: $S = \{5, 8, 9, 13, 17\}, K = 27$
Ekkor $\langle S, K \rangle \in \text{RÉSZLETÖSSZEG}$, mivel $5+9+13=27$.

Tétel

RÉSZLETÖSSZEG NP-teljes.

Bizonyítás: Egy S' részhalmaz polinom időben előállítható, majd polinom időben ellenőrizhető, hogy az S' -beli számok összege K -e. Így RÉSZLETÖSSZEG \in NP.

Megmutatjuk, hogy $3\text{SAT} \leq_p \text{RÉSZLETÖSSZEG}$.
Legyen φ 3KNF n változóval és m klózzal.

S $3m + 2n$ darab $n + m$ számjegyű számból fog állni. j . számjegy alatt a legkisebb helyiértékűtől (azaz hátulról) számított j -edik számjegyet értjük (az 1-es helyiértékű az 1. számegy).

Részletösszeg probléma

Az $1 - m$. számjegyeket megfeleltetjük a klózoknak, az $(m + 1) - (m + n)$. számjegyeket pedig a változóknak.

- ▶ az i . változóhoz két számot rendelünk hozzá: mindkettőben az $m + i$. bit 1-es. Ezen felül az első számban azon $1 - m$. számjegyek 1-esek, ahol x_i , míg a másik számban azon $1 - m$. számjegyek 1-esek, ahol $\neg x_i$ szerepel a helyiértéknek megfelelő klózban. A többi számjegy 0.
- ▶ A j . klózhoz 3 számot rendelünk. Mindegyiknek 1 számjegy kivételével minden számjegye 0, az egyetlen kivétel a j . számjegy, ez a 3 számban legyen rendre 5, 6 és 7.

Példa: $\varphi = (x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4)$.

A számok:

000101 001000 010000 100001
000100 001010 010011 100010
000005 000006 000007
000050 000060 000070

Részletösszeg probléma

Vegyük észre, hogyha minden számot összeadunk akkor az $1 - m$. számjegyek mindegyike 21, az $(m + 1) - (m + n)$. számjegyek mindegyike 2.

Nem választottuk még meg a K számot, ehhez válasszunk egy 21-nél nagyobb számot, pl. 32-t. K -t 32-es számrendszerben adjuk meg. Legyen K minden $1 - m$. számjegye 8, minden $(m + 1) - (m + n)$. számjegye pedig 1-es. Mivel $21 < 32$, ezért K csak úgy érhető el, ha minden helyiértéken 8 illetve 1 az összeg.

Ha φ kielégíthető, akkor van egy I interpretáció, ami igazra értékeli. Válasszuk az első $2n$ szám közül azt az n -et, ami I igaz literáljainak felel meg. Ekkor az $(m + 1) - (m + n)$. számjegyeknél 1 az összeg, míg az $1 - m$. számjegyeknél 1, 2, vagy 3 aszerint, hogy klózonként hány literál igaz. A további $3m$ számmal minden 1 és m közötti számjegy 8-cá egészíthető ki.

Részletösszeg probléma

Fordítva, tegyük fel, hogy az S' részhalmazbeli számok összege K . Ez csak úgy lehetséges, ha az első $2n$ számból pontosan n -et választottunk, minden változóra 1-et. Ha az i . változóhoz rendelt 2 szám közül az első S' -beli, akkor legyen x_i igaz, különben hamis.

S' minden $1 \leq j \leq m$ -re pontosan 1-et tartalmaz a j . klózhoz rendelt 3 számból, hiszen ha egyet se tartalmazna, akkor legfeljebb 3, ha legalább 2-t akkor legalább 11 lenne a j . számjegy. Ez azt jelenti, hogy 8-at csak úgy kaphatunk, hogy 1, 2 vagy 3 darab 1-es szerepel az összegben. Ez viszont azt jelenti, hogy minden klózban 1, 2 vagy 3 literál igaz, tehát φ kielégíthető.

Mivel S $O(n + m)$ darab $O(n + m)$ jegyű számból áll és K is $O(n + m)$ jegyű ezért a $\varphi \mapsto (S, K)$ függvény polinom időben kiszámítható, tehát 3SAT polinom időben visszavezethető

RÉSZLETÖSSZEG-re, így RÉSZLETÖSSZEG NP-teljes. \square

Hátizsák probléma

A HÁTIZSÁK nyelv olyan $a_1, \dots, a_n, b, p_1, \dots, p_n, k$ rendezett $(2n + 2)$ -esekből áll, ahol ezen számok mindegyike nemnegatív és van egy olyan $I \subseteq \{1, \dots, n\}$ halmaz, amelyre $\sum_{i \in I} a_i \leq b$ és $\sum_{i \in I} p_i \geq k$.

Story: Adott egy kincsesbarlangban n kincs, az i . kincs térfogata a_i , eladása p_i profitot hoz. Ki tudunk-e hozni a b kapacitású hátizsákunkban legalább k profitot hozó kincset? (Feltesszük, hogy ha a kincsek össztérfogata legfeljebb b , akkor azt valahogy be tudjuk zsúfolni a hátizsákba.)

Tétel

HÁTIZSÁK NP-teljes.

Bizonyítás: HÁTIZSÁK NP-beli, mivel a tárgyak egy I részhalmazát előállítani és arra a 2 egyenlőtlenség teljesülését ellenőrizni az input méretében polinomiális.

Hátizsák probléma

RÉSZLETÖSSZEG \leq_p HÁTIZSÁK:

Legyen (S, K) RÉSZLETÖSSZEG egy bemenete, ahol

$$S = \{s_1, \dots, s_n\}.$$

$$a_i := s_i, p_i := s_i, b := K, k := K \text{ minden } 1 \leq i \leq n\text{-re.}$$

Ha valamely $I \subseteq \{1, \dots, n\}$ -re $\sum_{i \in I} s_i = K$, akkor

$$\sum_{i \in I} a_i = K = b \leq b \text{ és } \sum_{i \in I} p_i = K = k \geq k.$$

Ha valamely $I \subseteq \{1, \dots, n\}$ -re $\sum_{i \in I} a_i \leq b = K$ és $\sum_{i \in I} p_i \geq k = K$, akkor $\sum_{i \in I} s_i = K$, mivel $a_i = p_i = s_i$ minden $1 \leq i \leq n$ -re.

(S, K) -ból $a_1, \dots, a_n, b, p_1, \dots, p_n, k$ polinom időben kiszámítható, tehát RÉSZLETÖSSZEG \leq_p HÁTIZSÁK, így HÁTIZSÁK NP-nehéz, és NP-belisége miatt NP-teljes is. \square

Partíció probléma

PARTÍCIÓ := $\{\langle B \rangle \mid B \text{ olyan pozitív számok multihalmaza, amely két egyenlő összegű részre particionálható}\}$.

Példa: A 2,2,2,3,3,4 multihalmaz ilyen, hiszen pl. $2+2+4=2+3+3$.

Tétel

PARTÍCIÓ NP-teljes.

Bizonyítás: PARTÍCIÓ NP-beli, egy részhalmazt előállítani és a két részhalmaz elemeit összeadni polinom időben megy.

RÉSZLETÖSSZEG \leq_p PARTÍCIÓ. Legyen $S = \{s_1, \dots, s_m\}$ és b a RÉSZLETÖSSZEG egy bemenete, feltehető, hogy $b \leq s = \sum_{i=1}^m s_i$.

$B := \{s_1, \dots, s_m, s+1-b, b+1\}$. Ekkor

$$\langle B \rangle \in \text{PARTÍCIÓ} \iff \langle S, b \rangle \in \text{RÉSZLETÖSSZEG}.$$

Ehhez elég annyit észrevenni, hogy B -ben $2s+2$ a számok összege, az utolsó kettő pedig $s+2$, ami több, mint az összeg fele, így ez a két szám másik félben kell legyen.

A visszavezetés nyilván polinomiális. \square

Ládapakolás

Számítási feladat: Hány egységnyi súlykapacitású ládába lehet bepakolni az s_1, \dots, s_n (≤ 1) súlyú tárgyakat?

Eldöntési probléma: Bele lehet-e pakolni az s_1, \dots, s_n súlyú tárgyakat k darab egységnyi súlykapacitású ládába?

LÁDAPAKOLÁS := $\{\langle s_1, \dots, s_n, k \rangle \mid s_i \in \mathbb{Q}^+ (1 \leq i \leq n) \text{ súlyok}$
particionálhatók $k \in \mathbb{N}^+$ részre úgy, hogy minden
particióban a súlyok összege $\leq 1\}$.

Példa: 0,34; 0,44; 0,54; 0,64 súlyú tárgyak esetén nem járunk jól, ha a 2 legkebbebb súlyút egy ládába rakjuk, ekkor ugyanis 3 láda kell. Könnyű találni csak 2 ládát használó ládapakolást.

Ládapakolás

Tétel

LÁDAPAKOLÁS NP-teljes.

Bizonyítás: NP-beli, hiszen egy partíció előállítása, majd a partíciókra a súlyhatár betartásának ellenőrzése polinomiális időben megy.

PARTÍCIÓ \leq_p LÁDAPAKOLÁS:

Legyenek b_1, \dots, b_n a B multihalmaz elemei és $b = \sum_{i=1}^n b_i$.

Legyenek az L multihalmaz elemei $2b_1/b, \dots, 2b_n/b$. Ekkor könnyen láthatóan $\langle B \rangle \in \text{PARTÍCIÓ} \iff \langle L, 2 \rangle \in \text{LÁDAPAKOLÁS}$.

A visszavezetés nyilván polinomiális. \square

NP lehetséges szerkezete

Definíció

L NP-köztes, ha $L \in \text{NP}$, $L \notin \text{P}$ és L nem NP-teljes.

Ladner tétele

Ha $\text{P} \neq \text{NP}$, akkor létezik NP-köztes nyelv.

(biz. nélkül)

Mivel nem tudjuk, hogy $\text{P} \stackrel{?}{=} \text{NP}$, ezért nem tudjuk, hogy léteznek-e NP-köztes nyelvek. Valószínűleg igen, hiszen azt gondoljuk, hogy $\text{P} \neq \text{NP}$.

Vannak azonban olyan nyelvek, amelyeknek se a P-beliségét, se az NP-teljességét nem sikerült eddig igazolni az intenzív próbálkozások ellenére sem, így erős NP-köztes jelölteknek számítanak.

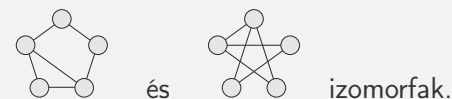
NP-köztes jelöltek – Gráfizomorfizmus

Definíció

A $G_i = (V_i, E_i)$ ($i = 1, 2$) irányítatlan gráfok **izomorfak**, ha van olyan $f : V_1 \rightarrow V_2$ bijekció, hogy $\forall u, v \in V_1$ esetén $\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$.

GRÁFIZOMORFIZMUS = $\{\langle G_1, G_2 \rangle \mid G_1 \text{ és } G_2 \text{ irányítatlan izomorf gráfok}\}$.

Példa:



Megjegyzés: A gráfizomorfizmus probléma számos, gyakorlatban előforduló speciális esete P-beli. Például:

- ▶ fák
- ▶ síkba rajzolható gráfok
- ▶ korlátos fokszámú gráfok

NP-köztes jelöltek – Gráfizomorfizmus

Nehéz eset például, ha nagyméretű gráfokban minden fokszám $\sqrt{|V|}$ körüli.

Egy új eredmény: Babai László, magyar matematikus 2017-es eredménye:

Tétel: $\text{GRÁFIZOMORFIZMUS} \in \text{QP}$, ahol

$$\text{QP} = \bigcup_{c \in \mathbb{N}} \text{TIME}(2^{(\log n)^c})$$

a „kvázipolinom időben” megoldható problémák osztálya.

A gráfizomorfizmus probléma alábbi általánosítása viszont már NP-teljes.

$\text{RÉSZGRÁFIZOMORFIZMUS} = \{\langle G_1, G_2 \rangle \mid G_1 \text{ és } G_2 \text{ irányítatlan gráfok és } G_1 \text{ izomorf } G_2 \text{ egy részgráfiával}\}$.

Részgráfizomorfizmus NP-teljes

Megjegyzés: A részgráf nem feszítetten értendő, tehát például $G_1 = (\{a, b, c\}, \{\{a, b\}, \{b, c\}\})$ 2 élből álló út részgráfja $G_2 = (\{A, B, C, D\}, \{\{A, B\}, \{A, C\}, \{A, D\}, \{B, C\}, \{B, D\}, \{C, D\}\})$ teljes 4 csúcsú gráfnak, hiszen G_1 izomorf $G_3 = (\{A, B, C\}, \{\{A, B\}, \{B, C\}\})$ -mal, ami részgráfja G_2 -nek annak ellenére, hogy $\{A, C\} \in E(G_2)$.

Tétel

$\text{RÉSZGRÁFIZOMORFIZMUS}$ NP-teljes.

Bizonyítás: $\text{RÉSZGRÁFIZOMORFIZMUS}$ NP-beli, hiszen egy NTG polinom időben előlíthat egy f bijekciót $V(G_1)$ és $V(G_2)$ egy $|V(G_1)|$ méretű részhalmaza között, majd polinom időben ellenőrizhető, hogy f izomorfizmus-e.

$\text{IHK} \leq_p \text{RÉSZGRÁFIZOMORFIZMUS}$. Legyen G egy irányítatlan gráf. G_1 legyen egy $|V(G)|$ csúcsú kör, $G_2 := G$. Ekkor G -ben van Hamilton kör, akkor és csak akkor, ha G_2 -nek van G_1 -gyel izomorf részgráfja. A visszavezetés nyilván polinomiális. \square

NP-köztes jelöltek – Prímfaktorizáció

Számítási feladat:

Prímfaktorizáció: adjuk meg egy egész szám prímtényezőző felbontását!

A probléma eldöntési változata:

PRÍMFAKTORIZÁCIÓ =

$$\{\langle n, k \rangle \mid n\text{-nek van } k\text{-nál kisebb prímtényezője}\}$$

Alkalmazás:

RSA eljárás: 1976-ban Ron Rivest, Adi Shamir és Len Adleman kifejlesztett egy nyílt kulcsú titkosító algoritmust, amely két nagy prím összeszorozásával azt használja ki, hogy nem ismeretes polinomiális algoritmus egy összetett szám prímtényezőinek meghatározására.

Bár bizonyítani nem tudjuk, hogy PRÍMFAKTORIZÁCIÓ nem P-beli, mégis az RSA algoritmus adatok biztonságos továbbítására a mai napig az egyik leggyakrabban használt algoritmus.

coC bonyolultsági osztályok

Definíció

Ha \mathcal{C} egy bonyolultsági osztály $\text{co}\mathcal{C} := \{L \mid \bar{L} \in \mathcal{C}\}$.

Definíció

\mathcal{C} **zárt a polinomidejű visszavezetésre nézve**, ha minden esetben ha $L_2 \in \mathcal{C}$ és $L_1 \leq_p L_2$ teljesül következik, hogy $L_1 \in \mathcal{C}$.

Volt: P és NP zártak a polinomidejű visszavezetésre nézve.

Tétel

Ha \mathcal{C} zárt a polinomidejű visszavezetésre nézve, akkor $\text{co}\mathcal{C}$ is.

Bizonyítás: Legyen $L_2 \in \text{co}\mathcal{C}$ és L_1 tetszőleges nyelvek, melyekre $L_1 \leq_p L_2$. Utóbbiból következik, hogy $\bar{L}_1 \leq_p \bar{L}_2$ (ugyananaz a visszavezetés jó!). Mivel $\bar{L}_2 \in \mathcal{C}$, ezért a tétel feltétele miatt $\bar{L}_1 \in \mathcal{C}$. Azaz $L_1 \in \text{co}\mathcal{C}$. \square

coC bonyolultsági osztályok

Következmény

coNP zárt a polinom idejű visszavezetésre nézve.

Igaz-e, hogy $P = \text{coP}$? **Igen.** (L -et polinom időben eldöntő TG q_i és q_n állapotát megcseréljük: \bar{L} -t polinom időben eldöntő TG.)

Igaz-e, hogy $NP = \text{coNP}$? A fenti konstrukció NTG-re **nem feltétlen** \bar{L} -t dönti el. Valójában azt sejtjük, hogy $NP \neq \text{coNP}$.

Tétel

$L \in \mathcal{C}$ -teljes $\iff \bar{L} \in \text{co}\mathcal{C}$ -teljes.

Bizonyítás:

- Ha $L \in \mathcal{C}$, akkor $\bar{L} \in \text{co}\mathcal{C}$.
- Legyen $L' \in \mathcal{C}$, melyre $L' \leq_p L$. Ekkor $\bar{L}' \leq_p \bar{L}$. Ha L' befutja \mathcal{C} -t akkor \bar{L}' befutja $\text{co}\mathcal{C}$ -t. Azaz minden $\text{co}\mathcal{C}$ -beli nyelv polinom időben visszavezethető \bar{L} -re.

Tehát \bar{L} $\text{co}\mathcal{C}$ -beli és $\text{co}\mathcal{C}$ -nehéz, így $\text{co}\mathcal{C}$ -teljes. \square

Példák coNP teljes nyelvekre

$\text{UNSAT} := \{\langle \varphi \rangle \mid \varphi \text{ kielégíthetetlen nulladrendű formula}\}.$

$\text{TAUT} := \{\langle \varphi \rangle \mid \varphi \text{ nulladrendű formula tautológia}\}.$

Tétel

UNSAT és TAUT coNP-teljesek.

Bizonyítás: $\bar{\text{ALTSAT}} = \{\langle \varphi \rangle \mid \varphi \text{ kielégíthető nulladrendű formula}\}$ is NP-teljes (NP-beli és SAT speciális esete neki.)

$\bar{\text{ALTSAT}} := \text{UNSAT}$, az előző tétel alapján UNSAT coNP-teljes.

$\text{UNSAT} \leq_p \text{TAUT}$, hiszen $\varphi \mapsto \neg\varphi$ polinom idejű visszavezetés. \square

Informálisan: coNP olyan nyelveket tartalmaz, amelyekbe való tartozás **polinom időben cáfolható**.

Például egy φ -t kielégítő interpretáció cáfolja, hogy φ kielégíthetetlen lenne. Egy φ -t hamisra értékelő interpretáció cáfolja, hogy φ tautológia lenne. Egy interpretáció polinom időben előállítható és adott interpretációban a formula igazságértéke polinom időben kiszámítható.

coNP-teljes nyelvek egy tulajdonsága

Tétel

Ha L coNP-teljes és $L \in \text{NP}$, akkor $\text{NP} = \text{coNP}$.

Bizonyítás: Legyen $L' \in \text{coNP}$ tetszőleges. Mivel L coNP-teljes, ezért $L' \leq_p L$, de mivel L NP-beli, ezért korábbi tételünk szerint $L' \in \text{NP}$. Tehát $\text{coNP} \subseteq \text{NP}$.

Legyen most $L \in \text{NP}$ tetszőleges, ekkor coNP definíciója miatt $\bar{L} \in \text{coNP}$. $\text{coNP} \subseteq \text{NP}$ miatt $\bar{L} \in \text{NP}$, majd ismét coNP definíciója miatt $L \in \text{coNP}$. Tehát $\text{NP} \subseteq \text{coNP}$ is teljesül. \square

Sejtés: $\text{NP} \neq \text{coNP}$.

Amennyiben a sejtés igaz, akkor UNSAT és TAUT még csak nem is NP-beliek. Mindenesetre nem ismeretes ezen nyelvekhez való tartozásra polinom időben kiszámítható és ellenőrizhető bizonyíték (tanú).

NP \cap coNP

Ha $\text{NP} \neq \text{coNP}$ egy érdekes osztály lehet $\text{NP} \cap \text{coNP}$. Nyilván $P \subseteq \text{NP} \cap \text{coNP}$. Először nézzünk néhány példát $\text{NP} \cap \text{coNP}$ -beli nyelvre.

1. példa

ÖSSZEFÜGGŐ := $\{\langle G \rangle \mid G = (V, E) \text{ összefüggő, irányítatlan gráf}\}$

ÖSSZEFÜGGŐ NP-beli, hiszen egy n csúcsú gráf összefüggőségére bizonyíték $\binom{n}{2}$ út a pontpárok között. Egy út hossza legfeljebb n . Így egy NTG előállíthat $\binom{n}{2}$ darab legfeljebb n csúcsból álló sorozatot polinom idő alatt, majd polinom időben ellenőrizheti, hogy ezek tényleg utak-e a pontpárok között.

ÖSSZEFÜGGŐ coNP-beli is, hiszen a csúcsok egy olyan X részhalmaza cáfolat, hogy X és $V \setminus X$ között nem megyél. V egy X részhalmazának előállítása és annak ellenőrése hogy X és $V \setminus X$ között nem megyél polinom időben megvalósítható.

ÖSSZEFÜGGŐ P-ben van, például szélességi kereséssel polinom időben ellenőrizhető egy gráf összefüggősége.

NP \cap coNP

2. példa

Definíció

Legyen $G = (A, B, E)$ (irányítatlan) páros gráf. Egy $M \subseteq E$ élhalmaz **teljes párosítás**, ha az $(A \cup B, M)$ gráfban minden csúcs foka pontosan 1.

TELJES PÁROSÍTÁS := $\{\langle G \rangle \mid G = (A, B, E) \text{ páros gráfban van teljes párosítás}\}$

TELJES PÁROSÍTÁS NP-beli, hiszen $(a, b) \in A \times B$ párok egy $|A|$ méretű listája polinom időben előállítható, és polinom időben ellenőrizhető, hogy ez teljes párosítást ad.

NP \cap coNP

TELJES PÁROSÍTÁS coNP-belisége Frobenius tételéből adódik.

Tétel: A $G = (A, B, E)$ páros gráfban akkor és csak akkor létezik teljes párosítás, ha $|A| = |B|$ és minden $X \subseteq A$ halmazra legalább $|X|$ olyan B-beli csúcs van, amelyik valamelyek X-beli csúccsal szomszédos.

A egy X részhalmaza polinom időben előállítható és szintén polinom időben ellenőrizhető, hogy nem teljesül-e a rá a tétel feltétele.

TELJES PÁROSÍTÁS P-ben van, amit a **magyar módszer** nevű König Dénes és Egerváry Jenő munkássága nyomán Harold Kuhn által adott polinomiális algoritmus mutat, mely egy páros gráfban keres maximális méretű (részleges) párosítást.

Ötlete: vegyünk független éleket, amíg tudunk, majd keressünk javító alternáló utat, azaz olyan utat ami egy A-beli és egy B-beli párosításon kívüli csúcs között fut és az élei váltakozva párosításon kívüliek illetve belüliek.

NP \cap coNP

3. példa Prímtesztelés

PRÍMEK := $\{p \mid p \text{ prím}\}$.

Fontos észrevétel, hogy egy $p \in \text{PRÍMEK}$ hossza p számjegyeinek száma, azaz $\Theta(\log p)$.

PRÍMEK coNP-belisége könnyen látható, hiszen egy n szám legfeljebb \sqrt{n} méretű osztója cáfolja, hogy n prím lenne. A cáfolat mérete $O(\log n)$, a maradékos osztás $O(\log^3 n)$ időben végrehajtható.

Megjegyzés: \sqrt{n} -ig determinisztikusan kipróbálni minden számot túl lassú, $\log n$ -ben exponenciális.

PRÍMEK NP-belisége már nem ilyen egyszerű, szükség van egy gyorsan ellenőrizhető prímtesztre.

NP \cap coNP

Tétel (Lucas prímtesztje) n prím \Leftrightarrow létezik $1 \leq x \leq n-1$, melyre $x^{n-1} \equiv 1 \pmod{n}$, de $x^{(n-1)/p} \not\equiv 1 \pmod{n}$ ($n-1$)-nek minden p prímosztójára.

Ez alapján a következő $(\log n)$ -ben polinom idejű rekurzív nemdeterminisztikus algoritmus készíthető:

Nemdeterminisztikus prímfelismerés(n)

if $n = 2$ then return 'igen';

if $n = 1$ vagy $n > 2$ páros then return 'nem';

if $n > 2$ páratlan then

legyen $1 < x < n$;

ellenőrizzük, hogy $x^{n-1} \equiv 1 \pmod{n}$ igaz-e

tippelünk $n-1$ prímfelbontására: p_1, \dots, p_k

ellenőrizzük, hogy $n-1 = p_1 \cdots p_k$

ellenőrizzük, minden $1 \leq i \leq k$ -ra, hogy p_i prím és

hogy $x^{(n-1)/p_i} \not\equiv 1 \pmod{n}$

return 'igen', ha minden ellenőrzés rendben

NP \cap coNP

Egy potenciális prímfelbontás $O(\log n)$ prímből áll, melyek mindegyike $O(\log n)$ hosszú. Egy $\text{mod } n$ hatványozás végrehajtható $O(\log^3 n)$, az összes $O(\log^4 n)$, időben.

Összességében az algoritmus lépésszáma $m = \lceil \log_2 n \rceil$ jegyű bemenet esetén

$T(m) \leq cm^4 + \sum_{i=1}^k T(m_i)$ alakú rekurzió adódik, ahol m_i a p_i számjegyeinek száma ($1 \leq i \leq k$).

Nyilván $\sum_{i=1}^k m_i \leq m$ és $m_i \leq m-1$ minden $1 \leq i \leq k$ -ra.

Innen teljes indukcióval $T(m) \leq cm^5$ bizonyítható:

$$T(m) \leq cm^4 + \sum_{i=1}^k cm_i^5 \leq cm^4 + c(m-1)^4 \sum_{i=1}^k m_i = cm(m^3 + (m-1)^4) \leq cm^5$$

Az algoritmus $O(\log^5 n)$ időkorlátos, így PRÍMEK NP-beli.

NP \cap coNP

Sokáig nyitott kérdés volt a számjegyek számában polinomiális determinisztikus prímteszt létezése, azaz, hogy PRÍMEK P-beli-e.

AKS-prímteszt: 2002-ben M. Agrawal, N. Kayal, N. Saxena indiai tudósok készítettek egy $O(\log^{12} n \log^k \log n)$ idejű determinisztikus prímtesztet (k egy konstans), amivel bebizonyították, hogy PRÍMEK P-beli. Munkájukkal elnyerték a Gödel- és Fulkerson-díjakat 2006-ban. Később a hatékonyságot $O(\log^6 n \log^k \log n)$ -re sikerült javítani.

A fenti példák azt sugallják, hogy egy NP \cap coNP-beli problémáról végül mindig kiderül, hogy P-beli, ám ez valószínűleg nem igaz.

Sejtés: $P \neq \text{NP} \cap \text{coNP}$.

A számításelmélet alapjai II.

11. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

Egy megoldási javaslat: Bevezethetjük az többlet tárigény fogalmát, ami az **input tárolására használt cellákon felül** igénybevett cellák száma.

Vannak olyan TG-ek, melyek csak az input területét használják, ám azt akár többször is átírják. Ezt beszámítsuk?

Eldöntési problémánál beszámítjuk.

Számítási problémánál viszont ne számítsanak bele a tárigénybe a csak a kimenet előállításához felhasznált cellák.

Az offline Turing gép

Definíció

Az **offline Turing gép** (OTG) egy olyan TG, melynek az első szalagja csak olvasható, a többi írható is. Első szalagját bemeneti szalagnak, további szalagjait munkaszalagoknak nevezzük.

Megjegyzés: Egy k munkaszalaggal rendelkező OTG állapotátmenetfüggvénye tehát
 $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^k \times \{L, S, R\}^{k+1}$.

Tétel

Minden TG-hez megadható vele ekvivalens offline TG.

Bizonyítás: Legyen M tetszőleges k szalagos TG. Az M' OTG-nak legyen $k + 1$ szalagja. M' másolja át az inputját a $k + 1$. szalagra és utána működjön úgy a $2 - (k + 1)$. szalagján, mint M . A $k + 1$. szalag felel meg M 1. szalagjának. Ekkor nyilván $L(M') = L(M)$.

Megjegyzés: Fordítva is igaz, az offline TG-ek speciális TG-ek.

Offline Turing gép verziók

Definíció

A **nemdeterminisztikus offline Turing gép** (NOTG) egy nemdeterminisztikusan működő offline Turing gép.

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító Turing gép, amelynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Az első szalagot bemeneti szalagnak, utolsó szalagot kimeneti szalagnak, a többi szalagot munkaszalagnak nevezzük.

Megjegyzés: Egy $k + 2$ szalagos, azaz k munkaszalaggal rendelkező OTG állapottátmenetfüggvénye tehát
 $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^{k+1} \rightarrow Q \times \Gamma^{k+1} \times \{L, S, R\}^{k+2}$.

A bal oldalon a Γ^{k+1} az $1 - (k + 1)$. szalagoknak, a jobboldalon $2 - (k + 2)$. szalagoknak felel meg.

Az offline Turing gépek tárigénye

Definíció

Egy offline TG **többlet tárigénye** egy adott inputra azon celláknak a száma, amelyeken a működés során valamelyik munkaszalag feje járt.

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ a többlet tárigénye.

Számító OTG-re hasonlóan.

Definíció

Egy nemdeterminisztikus offline TG **többlet tárigénye** egy adott inputra a legnagyobb többlet tárigényű számításának az többlet tárigénye.

Egy nemdeterminisztikus offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ az többlet tárigénye.

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos determinisztikus offline TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ többlet tárkorlátos nemdeterminisztikus offline TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k)$.
- ▶ $\text{NPSPACE} := \bigcup_{k \geq 1} \text{NSPACE}(n^k)$.
- ▶ $\text{L} := \text{SPACE}(\log n)$.
- ▶ $\text{NL} := \text{NSPACE}(\log n)$.

Megjegyzés: Így tehát az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk. Legalább lineáris tárigények esetén nem lenne szükség az offline TG fogalmára, használhattuk volna az eredeti TG fogalmat is.

ELÉR determinisztikus tárbonyolultsága

$\text{ELÉR} = \{\langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út}\}.$
Algo 2-ből, tudjuk, hogy ELÉR P-ben van (szélességi bejárás).

Tétel

$\text{ELÉR} \in \text{TIME}(n^2).$

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n).$

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ▶ $\text{ÚT}(x, y, i) := \text{igaz}$, ha \exists x -ből y -ba legfeljebb 2^i hosszú út.
- ▶ s -ből van t -be út G -ben $\iff \text{ÚT}(s, t, \lceil \log_2 n \rceil) = \text{igaz}$.
- ▶ $\text{ÚT}(x, y, i) = \text{igaz} \iff \exists z (\text{ÚT}(x, z, i-1) = \text{igaz} \wedge \text{ÚT}(z, y, i-1) = \text{igaz})$.
- ▶ Ez alapján egy rekurzív algoritmust készíthetünk, melynek persze munkaszalagján tárolnia kell, hogy a felsőbb szinteken milyen (x, y, i) -kre létezik folyamatban lévő hívás.

ELÉR determinisztikus tárbonyolultsága

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út kéne: ez az input alapján megválaszolható
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log_2 n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log_2 n \rceil$ -től.
- ▶ Az $\text{ÚT}(x, y, i)$ függvény meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i-1)$ hármaszt a munkaszalagra az (x, y, i) utáni helyre majd kiszámítja $\text{ÚT}(x, z, i-1)$ értékét.
- ▶ Ha hamis, akkor kitörli $(x, z, i-1)$ -et és z értékét növeli.
- ▶ Ha igaz, akkor is kitörli $(x, z, i-1)$ -et és $(z, y, i-1)$ -et írja a helyére (y -t tudja az előző (x, y, i) hármasból).
 - Ha $\text{ÚT}(z, y, i-1)$ igaz, akkor $\text{ÚT}(x, y, i)$ igaz (ezt (x, y, i) és $(z, y, i-1)$ 2. argumentumának egyezéséből látja)
 - Ha $\text{ÚT}(z, y, i-1)$ hamis akkor kitörli a $(z, y, i-1)$ -t és z értékét eggyel növelve $\text{ÚT}(x, z, i-1)$ -en dolgozik tovább.
- ▶ Ha egyik z se volt jó, akkor $\text{ÚT}(x, y, i)$ hamis.

ELÉR determinisztikus tárbonyolultsága

A főprogram, tehát $(s, t, \lceil \log_2 n \rceil)$ feírásából és az $\text{ÚT}(s, t, \lceil \log n \rceil)$ függvény meghívásából áll. Pontosan akkor lesz igaz a kimenet, ha t elérhető s -ből.

Az algoritmus a munkaszalagján végig legfeljebb $\lceil \log_2 n \rceil$ darab rendezett hármast tárol.

Egy szám tárolásához legfeljebb a szám adott számrendszer alapú logaritmus +1 darab számjegy szükséges.

Így a rendezett hármásokból mindvégig $O(\log n)$ van és egyenként $O(\log n)$ hosszúak, így $\text{ELÉR} \in \text{SPACE}(\log^2 n)$.

Konfigurációs gráf, elérhetőségi módszer

Definíció

Egy M NTG G_M konfigurációs gráfjának csúcsai M konfigurációi és $(C, C') \in E(G_M) \Leftrightarrow C \vdash_M C'$.

Elérhetőségi módszer: az $\text{ELÉR} \in \text{TIME}(n^2)$ vagy $\text{ELÉR} \in \text{SPACE}(\log^2 n)$ tételek valamelyikét alkalmazva a konfigurációs gráfra (vagy annak egy részgráfjára) bonyolultsági osztályok közötti összefüggéseket lehet bizonyítani.

Lássunk erre egy példát!

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NOTG és w az M egy n hosszú bemenete. Kell egy vele ekvivalens, $O(f^2(n))$ táras OTG. M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor $O(f(n) + \log n) = O(f(n))$. Feltehető, hogy M -nek csak egyetlen C_{elf} elfogadó konfigurációja van. (Törölje le a TG a munkaszalagjait, mielőtt q_i -be lép!) A legfeljebb $O(f(n))$ méretű konfigurációkat tartalmazó konfigurációs gráf mérete $2^{d \cdot f(n)}$ valamely $d > 0$ konstansra. Így az előző tétel szerint van olyan M' determinisztikus OTG, ami $O(\log^2(2^{d \cdot f(n)})) = O(f^2(n))$ tárral el tudja dönteni, hogy a kezdőkonfigurációból elérhető-e C_{elf} . M' lépjen pontosan ekkor az elfogadó állapotába, így $L(M') = L(M)$.

Determinisztikus/nemdeterminisztikus polinom tár

Következmény

$\text{PSPACE} = \text{NPSPACE}$

Bizonyítás: $L \in \text{NSPACE}(n^k) \xrightarrow{\text{Savitch}} L \in \text{SPACE}(n^{2k})$.

Tétel

$\text{NL} \subseteq \text{P}$

Bizonyítás

Legyen $L \in \text{NL}$ és M L -et $f(n) = O(\log n)$ tárral eldöntő NOTG. Meggondolható, hogy egy n méretű inputra M legfeljebb $f(n)$ méretű szalagtartalmakat tartalmazó konfigurációinak a száma legfeljebb $cnd^{\log n}$ alkalmas c, d konstansokkal, ami egy $p(n)$ polinommal felülről becsülhető. Így a G konfigurációs gráfnak legfeljebb $p(n)$ csúcsa van. G polinom időben megkonstruálható. Feltehető, hogy G -ben egyetlen elfogadó konfiguráció van. G -ben a kezdőkonfigurációból az elfogadó konfiguráció elérhetősége $O(p^2(n))$ idejű determinisztikus TG-pel eldönthető, azaz $L \in \text{P}$.

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra $(n = |V(G)|)$ a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0 -t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúcs
 - Nemdeterminisztikusan kiválasztja v egy ki-szomszédját és felírja u helyére a második szalagra
 - Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot (binárisan) eggyel
- ▶ Ha n -nél nagyobb szám áll a 3. szalagon, akkor elutasítja a bemenetet.

Mindkét szalag tartalmát $O(\log n)$ bittel kódolhatjuk.

L logaritmusos táras visszavezetésre való zártsága

Az M_1 OTG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)
- ▶ Ezután M_1 szimulálja M_2 aktuális lépését a harmadik szalagon lévő betű felhasználásával és aktualizálja a második szalagon M_2 fejének újabb pozícióját
- ▶ Ha M_2 elfogadó vagy elutasító állapotba lép, akkor M_1 lépjen a saját elfogadó vagy elutasító állapotába, egyébként folytassa a szimulációt a következő lépéssel

Belátható, hogy M_1 L_1 -et dönti el és a működése során csak logaritmusos méretű tárat használ, azaz $L_1 \in L$.

Logaritmusos táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmusos tárral visszavehető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmusos többlet tárkorlátozott determinisztikus offline Turing géppel. Jelölése: $L_1 \leq_L L_2$.

Definíció

Egy L nyelv **NL-nehez** (a log. táras visszavezetésre nézve), ha minden $L' \in NL$ nyelvre, $L' \leq_L L$. Ha ezen felül $L \in NL$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Tétel

Az L osztály zárt a logaritmusos tárral való visszavezetésre nézve.

Bizonyítás: Tegyük fel, hogy $L_1 \leq_L L_2$ és $L_2 \in L$.

Legyen M_2 az L_2 -t eldöntő, M pedig a visszavezetésben használt f függvényt kiszámoló logaritmusos táras determinisztikus OTG.

ELÉR NL-teljessége

Következmény

Ha egy L nyelv NL-teljes és $L \in L$, akkor $L = NL$.

Bizonyítás: Legyen $L' \in NL$ tetszőleges, ekkor L NL-teljessége miatt $L' \leq_L L$. $L \in L$, így L logaritmusos tárral való visszavezetésre való zártsága miatt $L' \in L$. Tehát $NL \subseteq L$. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmusos tárral történő visszavezetésre nézve.

Bizonyítás:

- ▶ Korábban láttuk, hogy $ELÉR \in NL$
- ▶ Legyen $L \in NL$, megmutatjuk, hogy $L \leq_L ELÉR$
- ▶ Legyen M egy L -et eldöntő $O(\log n)$ táras NOTG és $|u| = n$
- ▶ Az $O(\log n)$ tárat használó konfigurációk $\leq c \cdot \log n$ hosszúak (alkalmas c -re)

ELÉR NL-teljessége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak egy ilyen van), ha $u \in L(M)$. Így $L \leq \text{ELÉR}$.

Kell még, hogy a visszavezetés log. tárat használ, azaz G_M megkonstruálható egy log. táras N determinisztikus OTG-pel:

- ▶ N sorolja fel a hossz-lexikografikus rendezés szerint az összes legfeljebb $c \cdot \log n$ hosszú szót az egyik szalagján, majd tesztelje, hogy az legális konfigurációja-e M -nek, ha igen, akkor a szót írja ki a kimenetre
- ▶ Az élek (konfiguráció párok) hasonlóképpen felsorolhatók, tesztelhetők és a kimenetre írhatók

Immerman-Szelepcsényi tétel

$NL = \text{coNL}$

(bizonyítás nélkül)

Hierarchia tétel

$$\text{EXPTIME} := \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k}).$$

Hierarchia tétel

(I) $NL \subset \text{PSPACE}$ és $P \subset \text{EXPTIME}$.

(II) $L \subseteq NL = \text{coNL} \subseteq P \subseteq NP \subseteq \text{NPSpace} = \text{PSPACE} \subseteq \text{EXPTIME}$

Sejtés: A fenti tartalmazási lánc minden tartalmazása valódi.

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$L \stackrel{(1)}{\subseteq} NL \stackrel{(2)}{=} \text{coNL} \stackrel{(3)}{\subseteq} P \stackrel{(4)}{\subseteq} NP \stackrel{(5)}{\subseteq} \text{NPSpace} \stackrel{(6)}{=} \text{PSPACE} \stackrel{(7)}{\subseteq} \text{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

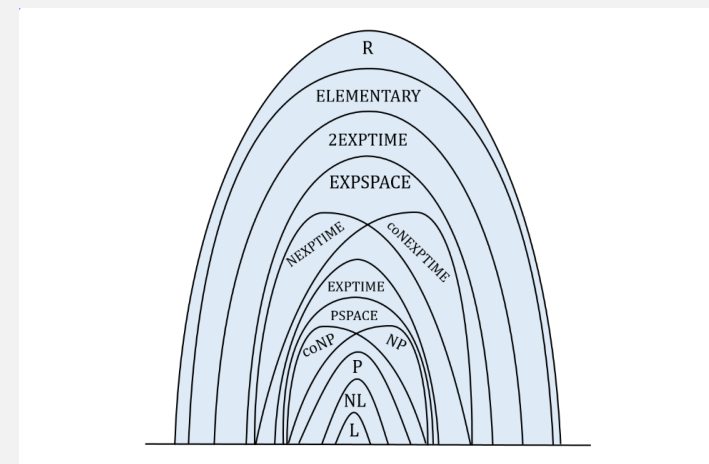
(2): Immerman- Szelepcsényi

(3),(6): előbb bizonyítottuk

(5): Ha egy NTG egy számítására adott egy időkorlát, akkor ennél a korlátnál több új cellát nincs ideje egyik fejnek sem felfedezni. Így ez az időkorlát egyben tárkorlát is.

(7): Elérhetőségi módszerrel: a használt tár méretének exponenciális függvénye a konfigurációs gráf mérete. A konfigurációs gráf méretében négyzetes (azaz összességében a tár méretében exponenciális) időben tudja egy determinisztikus TG az elérhetőséget tesztelni a kezdőkonfigurációból az elfogadó konfigurációba.

R szerkezete



R szerkezete (a tartalmazások valódisága nem mindenütt bizonyított)

[ábra: Gazdag Zs. e-jegyzet]

A számításelmélet alapjai II.

12. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

PSPACE-teljes problémák

A hierarchia tétel alapján tudjuk, hogy $NP \subseteq PSPACE \subseteq EXPTIME$.

Az a sejtésünk, hogy a fenti tartalmazások valódiak.

Számos érdekes probléma tartozik ebbe az osztályba, köztük számos olyan akármekkora táblán játszható kétszemélyes játék nyerő állásainak meghatározása, ahol a lépésszám polinomiálisan korlátozott, a játékosok felváltva lépnek és aktuális lépésük függ az ellenfél előző lépésétől.

A PSPACE-teljes problémák a problémaosztály legnehezebb problémái.

Teljesen kvantifikált Boole formulák

Definíció

Teljesen kvantifikált Boole formulának (TKBF-nek) nevezünk egy olyan formulát, melyben minden ítéletváltozó univerzális vagy egzisztenciális kvantorral kötött, ráadásul úgy, hogy a formula két részre osztható: elől vannak a kvantorok és utána egy kvantortmentes ítéletlogikai formula.

A kvantorok hatásköre a saját pozíciójuktól a formula végéig tart.

Példák:

$$\varphi_1 = \exists x_1 \forall x_2 (x_1 \rightarrow x_2)$$

$$\varphi_2 = \exists x_1 \forall x_2 ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))$$

A TKBF-ek zártak, igazságértékük változókiértékeléstől független.

φ_1 igaz, hiszen bármi is x_2 igazságértéke, ha x_1 -et hamisra állítjuk az ítéletlogikai rész igaz lesz.

φ_2 hamis, hiszen x_1 akár igaz akár hamis, van x_2 -nek olyan igazságértéke, hogy az egyik tag hamis lesz.

Q_{SAT} PSPACE-teljes

Definíció

$Q_{SAT} = \{ \langle \varphi \rangle \mid \varphi \text{ igaz teljesen kvantifikált Boole formula} \}$.

Az alábbi tétel szerint Q_{SAT} a legnehezebb problémák közé tartozik a PSPACE osztályon belül.

Tétel

Q_{SAT} PSPACE-teljes.

A PSPACE-teljességet a polinom idejű visszavezetésre nézve értjük.

Q_{SAT} PSPACE-teljes

Bizonyítás: Q_{SAT} PSPACE-beliségéhez tekintsük az ÉRTÉK(φ) függvényt, mely visszaadja a φ TKBF igazságértékét. ÉRTÉK(φ)-t kiszámíthatjuk a következő rekurzív algoritmussal.

function ÉRTÉK(φ)
if $\varphi = \forall x\psi$ **then return** ÉRTÉK(ψ^i) \wedge ÉRTÉK(ψ^h)
else if $\varphi = \exists x\psi$ **then return** ÉRTÉK(ψ^i) \vee ÉRTÉK(ψ^h)
else return $B(\varphi)$

ahol ha ψ egyetlen szabad változóval rendelkező formula

ψ^i : ψ egyetlen szabad változóját igazgal helyettesítjük.

ψ^h : ψ egyetlen szabad változóját hamissal helyettesítjük.

továbbá ha φ változómentes formula

$B(\varphi)$: a φ kiértékelésével kapott igazságérték.

Példa: Ha $\psi = \forall x(x \wedge \neg y)$, akkor $\psi^h = \forall x(x \wedge \neg h)$.

Ha $\varphi = i \wedge \neg h \rightarrow h$, akkor $B(\varphi) = h$.

Q_{SAT} PSPACE-teljes

Például $\varphi = \exists x_1 \forall x_2 ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))$ -re:

$\varphi_{ii} = ((\neg i \vee i) \wedge (i \vee \neg i))$. ÉRTÉK(φ_{ii}) = $B(\varphi_{ii}) = i$.

$\varphi_{ih} = ((\neg i \vee h) \wedge (i \vee \neg h))$. ÉRTÉK(φ_{ih}) = $B(\varphi_{ih}) = h$.

$\varphi_i = \forall x_2 ((\neg i \vee x_2) \wedge (i \vee \neg x_2))$

$((\neg i \vee x_2) \wedge (i \vee \neg x_2))^i = \varphi_{ii}$. $((\neg i \vee x_2) \wedge (i \vee \neg x_2))^h = \varphi_{ih}$.

ÉRTÉK(φ_i) = ÉRTÉK(φ_{ii}) \wedge ÉRTÉK(φ_{ih}) = $i \wedge h = h$.

$\varphi_{hi} = ((\neg h \vee i) \wedge (h \vee \neg i))$ ÉRTÉK(φ_{hi}) = $B(\varphi_{hi}) = h$.

$\varphi_{hh} = ((\neg h \vee h) \wedge (h \vee \neg h))$ ÉRTÉK(φ_{hh}) = $B(\varphi_{hh}) = i$.

$\varphi_h = \forall x_2 ((\neg h \vee x_2) \wedge (h \vee \neg x_2))$.

$((\neg h \vee x_2) \wedge (h \vee \neg x_2))^i = \varphi_{hi}$. $((\neg h \vee x_2) \wedge (h \vee \neg x_2))^h = \varphi_{hh}$.

ÉRTÉK(φ_h) = ÉRTÉK(φ_{hi}) \wedge ÉRTÉK(φ_{hh}) = $h \wedge i = h$.

$\varphi = \exists x_1 \forall x_2 ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))$.

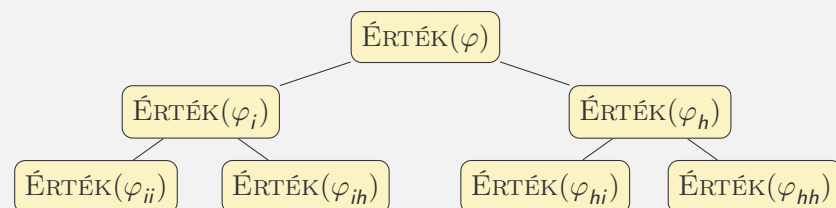
$\forall x_2 ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))^i = \varphi_i$.

$\forall x_2 ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))^h = \varphi_h$.

ÉRTÉK(φ) = ÉRTÉK(φ_i) \vee ÉRTÉK(φ_h) = $h \vee h = h$.

Q_{SAT} PSPACE-teljes

A rekurzív függvényhívások fája:



Az algoritmus tárigénye a formula méretében lineáris, hiszen a rekurzív hívási fa mélysége $O(n)$ és elég a rekurzió minden szintjén egy változó igazságértékét tárolni (a második gyerekre csak akkor hívjuk meg az eljárást, ha a szülő értéke nem következik az első gyerek értékéből, ha pedig meghívtuk a második gyerekre, abból következik az első gyerek visszaadott értéke). Tehát $Q_{SAT} \in PSPACE$.

Q_{SAT} PSPACE-teljes

A PSPACE-nehézség bizonyításához legyen L tetszőleges PSPACE-beli nyelv. Megmutatjuk, hogy $L \leq_p Q_{SAT}$. Legyen $M = \langle Q, \{0, 1\}, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $p(n)$ többlet tárigényű L -et eldöntő offline Turing gép és w az M egy tetszőleges $n(n \in \mathbb{N})$ hosszú bemenete. Feltehető, hogy $p(n)$ polinom és $p(n) \geq n$. Megadunk egy φ TKBF-et úgy, hogy $w \in L(M)$ akkor és csak akkor, ha $\langle \varphi \rangle \in Q_{SAT}$.

A tárigényre vonatkozó korlátból adódik, hogy M -nek w -re vonatkozó számítása legfeljebb $h = 2^{cp(n)}$ különböző konfigurációt tartalmaz valamely $c > 0$ -ra. Így feltehető, hogy M legfeljebb h lépésben elfogadja vagy elutasítja w -t, és az előző előadáson látottak alapján az is, hogy egyetlen elfogadó állapot van.

Q_{SAT} PSPACE-teljes

A bizonyítás során azzal az egyszerűsítéssel élünk, hogy a konfigurációkat az egyetlen munkaszalag tartalma, ezen szalag fejének helyzete és az aktuális állapot együttesen meghatározzák.

A lehetséges konfigurációk legyenek c_1, \dots, c_h , köztük $c_1 = c_{\text{kezdő}}$, $c_h = c_{\text{elfogadó}}$.

Ítéleváltozók: $x_{i,j,s}$ ($1 \leq i \leq h, 1 \leq j \leq p(n), s \in Q \cup \Gamma$).
 $x_{i,j,s}$ annak az állításnak felel meg, hogy c_i j -edik betűje s .

Egy I interpretáció akkor és csak akkor írja le azt, hogy a c_i konfiguráció legális, ha

- minden $1 \leq j \leq p(n)$ -re pontosan egy $s \in Q \cup \Gamma$ van, amelyre $x_{i,j,s}$ igaz
- pontosan egy $1 \leq j \leq p(n)$ -hez van olyan $s \in Q$, hogy $x_{i,j,s}$ igaz

Q_{SAT} PSPACE-teljes

Jelölés: ha X ítéleváltozók egy halmaza, akkor $\exists X$ -szel rövidítjük az X -beli változók egzisztenciális kvantifikálását, azaz a $\exists x_1 \dots \exists x_m$ -et, ha $X = \{x_1, \dots, x_m\}$. $\forall X$ -re analóg módon.

Mi csak konfigurációkat azonosító változóhalmazokat fogunk tekinteni, azaz az $X_k = \{x_{k,j,s} \mid 1 \leq j \leq p(n), s \in Q \cup \Gamma\}$ változóhalmaz megfelel a c_k konfigurációnak minden $1 \leq k \leq h$ -ra.

Példa: Legyen $c_k = aqbb$ egy konfiguráció.

A következő formula azt írja le, hogy az X_k -beli változók értékei úgy vannak beállítva, hogy azok a c_k konfigurációt határozzák meg.

$$\exists X_k (x_{k,1,a} \wedge x_{k,2,q} \wedge x_{k,3,b} \wedge x_{k,4,b} \wedge \bigwedge_{x \in X_k \setminus H} \neg x),$$

ahol $H = \{x_{k,1,a}, x_{k,2,q}, x_{k,3,b}, x_{k,4,b}\}$.

A formula persze igaz és mérete $O(p(n))$.

Q_{SAT} PSPACE-teljes

$\text{EQ}(X_k, X_\ell)$ egy olyan formulát jelöl, ami akkor és csak akkor igaz, ha minden $1 \leq j \leq p(n), s \in Q \cup \Gamma$ -ra $x_{k,j,s}$ és $x_{\ell,j,s}$ igazságértéke ugyanaz.

$\varphi_{k,\ell,t}$ jelölje azt a TKBF-et, ami akkor és csak akkor igaz, ha c_k -ből legfeljebb 2^t konfigurációátmenettel el lehet jutni c_ℓ -be.

Ekkor az $\text{ELÉR} \in \text{SPACE}(\log^2 n)$ bizonyításánál látott módon $\varphi_{k,\ell,t}$ igaz, akkor és csak akkor, ha van olyan $1 \leq m \leq h$, hogy $\varphi_{k,m,t-1}$ és $\varphi_{m,\ell,t-1}$ is igaz.

$\varphi_{k,\ell,0}$ leírható egy $O(p(n))$ hosszú $\text{EQ}(X_k, X_\ell) \vee \varphi'_{k,\ell,0}$ formulával, ahol $\varphi'_{k,\ell,0}$ azt írja le, hogy $c_k \vdash c_\ell$. Ugyanis a konfigurációk $O(p(n))$ hosszúak, a változás 2×3 -as „ablaka” $O(p(n))$ pozícióban lehet és konstans féleképpen tölthető ki, a többi változó egyenlősége szintén egy $O(p(n))$ hosszú részformulával leírható.

Azonban a $\varphi_{k,\ell,t} = \exists X_m (\varphi_{k,m,t-1} \wedge \varphi_{m,\ell,t-1})$ rekurzió nem jó, mivel ez exponenciális sok rekurzív hívást és így exponenciális méretű formulát és időigényt eredményezne.

Q_{SAT} PSPACE-teljes

Ehelyett:

$$\varphi_{k,\ell,t} = \exists X_m \forall X_i \forall X_j ((\text{EQ}(X_i, X_k) \wedge \text{EQ}(X_j, X_m)) \vee (\text{EQ}(X_i, X_m) \wedge \text{EQ}(X_j, X_\ell)) \rightarrow \varphi_{i,j,t-1}),$$

ami összesen csak $O(\log h) = O(p(n))$ rekurzív hívást eredményez (hiszen $w \in L$ akkor és csak akkor, ha $\varphi_{1,h,\lceil \log_2 h \rceil}$ igaz) és $\varphi_{k,\ell,t}$ $\varphi_{i,j,t-1}$ -n kívüli része $O(p(n))$ méretű ($\forall 1 \leq t \leq \lceil \log_2 h \rceil$ -re).

Tehát $\varphi' := \exists X_1 \exists X_h (\varphi_K \wedge \varphi_{1,h,\lceil \log_2 h \rceil} \wedge \varphi_V)$ polinom időben kiszámítható, ahol a φ_K és φ_V részformulák literálok 1 -1 olyan konjunkciója, melyek M w -hez tartozó c_1 kezdő- és c_h elfogadó konfigurációját írják le.

A kapott formulában nem feltétlen van minden kvantor a formula elején, de ez a kvantorok kiemelésével polinom időben orvosolható, így $L \leq_p \text{Q}_{\text{SAT}}$.

Így Q_{SAT} PSPACE-beli és PSPACE-nehéz, tehát PSPACE-teljes. \square

QSAT mint kétszemélyes játék

Megjegyzés: QSAT-nak az a változata is PSPACE teljes, amikor a kvantormentes résznek KNF-nek kell lennie. Ez nem teljesen nyilvánvaló, hiszen egy ítéletlogikai formulával tautológikusan ekvivalens KNF megadása akár exponenciális méretű formulát is eredményezhet.

A QSAT azon megszorítása amikor csak olyan bemeneteket tekintünk ahol a kvantorok alternálnak, az első és az utolsó kvantor a \exists , és a kvantormentes rész egy KNF, felfogható úgy, mint az alábbi **kétszemélyes játék**.

Tegyük fel, hogy adott egy ilyen alakú φ TKBF. Az első játékos választja meg a páratlan sorszámú változók értékét, és a célja a formula igazsá tétele. A második választja meg a páros sorszámú változók értékét, és a célja a formula hamissá tétele. Nyilvánvaló, hogy az első játékosnak akkor és csak akkor van nyerő stratégiája ebben a játékban, ha $\langle \varphi \rangle \in \text{QSAT}$.

Ez a változat is PSPACE-teljes.

FÖLDRAJZI JÁTÉK

Definíció

A **Földrajzi játék** a következő kétszemélyes játék. Adott egy G irányított gráf és annak egy p csúcsa. A két játékos p -ből kiindulva felváltva jelöli meg G még meg nem jelölt csúcsait, mindig az utoljára megjelölt csúcsból elérhető csúcsok közül választva. Az veszít, aki nem tud további csúcsot megjelölni.

1. példa: Alíz és Borcsa a következő játékot játssza. Megyeszékhelyeket mondanak felváltva, a következő játékosnak az előző játékos által mondott város utolsó betűjével kell kezdenie a sajátját.

Alíz: Szolnok

Borcsa: Kecskemét

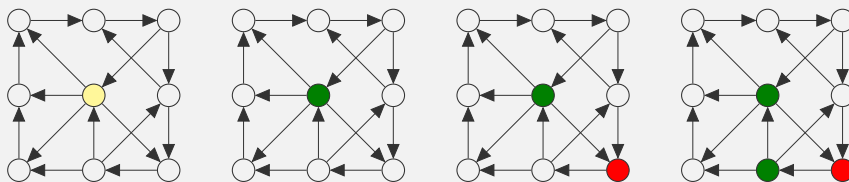
Alíz: Tatabánya

Alíz nyert, mert Borcsa nem tud mondani A-val új megyeszékhelyt.

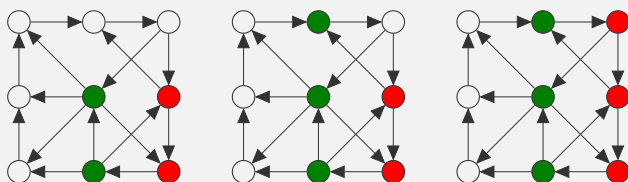
Ez egy földrajzi játék a megyeszékhelyek a gráfján, ahol az u városból akkor vezet irányított él a v városba, ha u utolsó betűje megegyezik v első betűjével (amennyiben a kezdő város rögzített).

FÖLDRAJZI JÁTÉK

2. példa: Az előző játék általánosítása, ha **Alíz** és **Borcsa** egy tetszőleges gráfon játszanak. Alíz kezd.



● : startcsúcs



Borcsa nyert, Alíz nem tud lépni.

Erre a gráfra és kezdőcsúcsra Borcsának van nyerő stratégiája. Az előző játék bizonyítja, Alíznek mindig csak 1 lépése volt.

FÖLDRAJZI JÁTÉK PSPACE-teljes

Definíció

$\text{FÖLDRAJZI JÁTÉK} = \{ \langle G, p \rangle \mid \text{a } G \text{ irányított gráfban } p \text{ kezdőcsúccsal az 1. játékosnak van nyerő stratégiája} \}$

Tétel

FÖLDRAJZI JÁTÉK PSPACE-teljes.

Bizonyítás: A játék **állása** alatt G csúcsainak egy olyan 2 színnel színezését értjük, ahol az egyik színosztály a jelölt, a másik a jelöletlen csúcsok, továbbá azt az információt, hogy ki következik soron és melyik csúcsból.

FÖLDRAJZI JÁTÉK PSPACE-beliségének belátásához készítsük el a játék fáját, ahol a csúcsok az állások és egy állás gyerekei a lehetséges következő állások. Irányítsuk a fát a levelektől a gyökér felé.

FÖLDRAJZI JÁTÉK PSPACE-teljes

A leveleket értékeljük ki igazra vagy hamisra aszerint, hogy az az 1. játékosnak a győzelmét vagy vereségét jelenti-e.

Bármely más, nem levél állás tekinthető egy VAGY kapunak ha az 1. játékos van soron, illetve egy ÉS kapunak, ha a 2. játékos következik. A kettőnél több bemenettel rendelkező kapuk helyettesíthetők bináris kapukkal.

A kiértékelés polinom tárral elvégezhető, mert

- ▶ minden játék polinomiális hosszúságú (legfeljebb a gráf csúcsszáma)
- ▶ adott állásból polinomiális tárral megkonstruálható minden lehetséges következő állás, illetve ha nincs következő állás akkor eldönthető ki nyert
- ▶ a játékfa polinomiális magasságú, és ez akkor is igaz, ha csak bináris kapukat használunk (az eredeti játékfaban a foksám nem nagyobb, mint G -ben a maximális ki-fok, ami konstans).

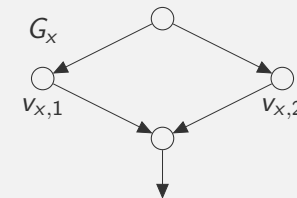
Ezekből a QSAT-nál már látott érvelés alapján következik, hogy a játékfa polinom tárral kiértékelhető.

FÖLDRAJZI JÁTÉK PSPACE-teljes

FÖLDRAJZI JÁTÉK PSPACE-nehézségének bizonyításához QSAT alternáló kvantoros, KNF magú változatát vezetjük vissza rá polinom időben.

Legyen tehát $\varphi = \exists x_1 \forall x_2 \dots \exists x_k \psi$ TKBF, ahol $\psi = c_1 \wedge \dots \wedge c_m$, ahol $m \geq 1$ és a c_i -k klózek, azaz ψ KNF.

Megkonstruáljuk a G_φ gráfot a következőképpen. Először minden φ -beli x változóhoz elkészítjük az alábbi G_x részgráfot.



A G_x részgráfok „egymás alatt” vannak és össze vannak kötve.

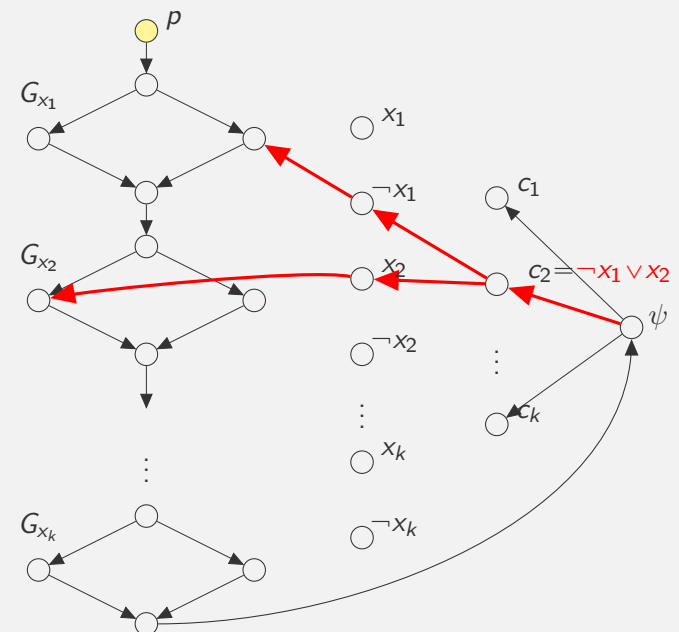
FÖLDRAJZI JÁTÉK PSPACE-teljes

G_φ -nek van egy csúcsa ami ψ -vel van címkézve. Az ebből kiinduló élek olyan csúcsokba vezetnek, melyek rendre ψ klózaival vannak címkézve. Tartozik egy csúcs minden literálhoz is. Minden klóznak megfelelő csúcsból vezessen el azoknak a literáloknak megfelelő csúcsokhoz, amelyekből az adott klóz áll.

Minden literállal címkézett csúcsból egyetlen él induljon ki a következőképpen. Ha a csúcs x -el van címkézve valamely x változóra, akkor a belőle kiinduló él $G_x v_{x,1}$ csúcsába, ha $\neg x$ -el van címkézve, akkor $G_x v_{x,2}$ csúcsába vezessen.

A p csúcsból induljon egyetlen él G_{x_1} felső csúcsába, valamint vegyünk hozzá egy élt G_{x_k} alsó csúcsából a ψ címkéjű csúcsba.

FÖLDRAJZI JÁTÉK PSPACE-teljes



FÖLDRAJZI JÁTÉK PSPACE-teljes

Nézzük meg, hogyan zajlik egy játék G_φ -ben.

A páratlan sorszámú változóknak megfelelő G_x részgráfok legfelső csúcsaiban az első játékos választja az irányt (azaz kezdetben ő választ egy csúcsot a p -ből kiindulva. A páros sorszámúaknak megfelelőekben viszont a második játékos. G_{x_k} -ban az első játékos választja az irányt és így a 2. játékos a legalsó csúcsot.

Az első játékosnak kell választania a ψ címkéjű csúcsot. A második játékos választ egy c klózt, majd az első játékos kiválasztja c egy ℓ literálját.

Itt két lehetőség van: vagy tud a második játékos még korábban nem választott csúcsot választani vagy nem.

G_φ konstrukciója miatt a második játékos pontosan akkor tud csúcsot választani, ha a következők egyike teljesül:

- ▶ $\ell = x$ valamely x változóra és $v_{x,1}$ -et még nem jelölték meg,
- ▶ $\ell = \neg x$ valamely x változóra és $v_{x,2}$ -et még nem jelölték meg.

FÖLDRAJZI JÁTÉK PSPACE-teljes

Ennek megfelelően az első játékos úgy próbálja megválasztani a páratlan indexű ítéletváltozóknak megfelelő gráfokban az irányt, hogy később tudjon bármely klóznak megfelelő csúcshoz egy olyan literálnak megfelelő csúcsot választani, ami olyan csúccsal van összekötve, ami már korábban választásra került.

Ezzel ellentétben a második játékos úgy próbálja megválasztani a páros indexű változóknak megfelelő részgráfokban az irányt, hogy később tudjon olyan c klózt választani, hogy a c -ből elérhető csúcsok mindegyike még korábban ki nem választott csúcsokkal legyen összekötve.

FÖLDRAJZI JÁTÉK PSPACE-teljes

Belátható, hogy az 1. játékosnak pontosan akkor van nyerő stratégiája G_φ -ben, ha a φ -beli kétszemélyes játék során is van, azaz pontosan akkor ha az 1. játékos meg tudja adni a páratlan indexű ítéletváltozók értékét úgy, hogy később akárhogyan is választ a második játékos egy ψ -beli klózt, az 1. játékos ki tud választani abban a klózban egy olyan literált, ami igaz lesz a kettejük játéka által meghatározott interpretációban.

Ebből következik, hogy a $\varphi \mapsto G_\varphi$ megfeleltetés Q_{SAT} visszavezetése a FÖLDRAJZI JÁTÉK-ra.

Az is könnyen látható, hogy G_φ a φ méretében polinom időben megkonstruálható, és ezzel a tételt bebizonyítottuk. \square

További PSPACE-teljes problémák

1-típusú szóprobléma

Reguláris grammatikák esetében lineáris, környezetfüggetlen grammatikák esetében pedig köbös időigényű algoritmust (CYK) tanultunk is ismert a szóprobléma eldöntésére az előző félévben.

Mivel $\mathcal{L}_0 = \text{RE}$, ezért a 0. típusú szóprobléma eldönthetetlen (de RE-beli).

Előző félévben tanultunk egy, az 1-típusú szóproblémát eldöntő exponenciális időigényű algoritmust. Bizonyítás nélkül megjegyezzük, hogy az 1-típusú szóprobléma PSPACE-teljes.

További PSPACE-teljes problémák

Általánosított, korlátos idejű go játék

Tétel: Annak eldöntése, hogy egy $n \times n$ -es táblán legfeljebb n^2 lépésig játszott go játékban egy adott állás fehér számára nyerő állás-e PSPACE-teljes.

Megjegyzés: A konkrét méretben (például 8x8-as vagy 19x19-es) táblán játszott kétszemélyes játékok (go, dáma, sakk) véges játékok, a PSPACE-teljesség azonban aszimptotikus fogalom. Ezekben a játékokban a keresési tér véges, bár hatalmas.

Megjegyzés: Ha elhagyjuk a lépésszámkorlátot, akkor a probléma EXPTIME-teljessé válik.

További PSPACE-teljes problémák

Helyben elfogadás problémája

HELYBEN ELFOGADÁS = $\{\langle M, w \rangle \mid \text{az } M \text{ determinisztikus TG úgy fogadja el } w\text{-t, hogy a fej nem hagyja el } w \text{ területét}\}$

Tétel

HELYBEN ELFOGADÁS PSPACE-teljes.

Bizonyítás: HELYBEN ELFOGADÁS PSPACE-beli:

M lehetséges konfigurációinak száma egy w bemenetre legfeljebb $m(w) = |Q| \cdot |w| \cdot |\Gamma|^{|w|}$, ahol Q az M állapotthalmaza és Γ a szalagábécéje.

HELYBEN ELFOGADÁS a következő S Turing géppel polinom tárral eldönthető.

- ▶ S szimulálja M -et w -n
- ▶ ha M elutasítja w -t akkor S lépjen q_n -be
- ▶ ha M feje elhagyja w területét, akkor S lépjen q_n -be
- ▶ ha M elfogadja w -t, akkor S lépjen q_i -be
- ▶ ha M $m(w)$ -nél többet lép, akkor S lépjen q_n -be

További PSPACE-teljes problémák

Számolni kell a megtett lépéseket. A lépések száma $O(|w|)$ bittel leírható.

Az S determinisztikus TG működése során tehát $O(|w|) = O(|\langle M, w \rangle|)$ tárat használ és $L(S) = \text{HELYBEN ELFOGADÁS}$.

Tehát $\text{HELYBEN ELFOGADÁS} \in \text{PSPACE}$

Legyen $L \in \text{PSPACE}$ tetszőleges. Ekkor van olyan M determinisztikus TG, mely L -et $p(n)$ tárral eldönti. Tetszőleges w szóra legyen $f(w) := \langle M', \#^{p(n)} w \#^{p(n)} \rangle$, ahol $\#$ egy olyan szimbólum, ami nincs benne M szalagábécéjében, M' pedig w első betűjére teker, majd ugyanúgy működik, mint M $\#$ -t \sqcup -ként kezelve.

Ekkor f polinom idejű visszavezetés, hiszen $w \in L(M)$ akkor és csak akkor ha $\langle M', \#^{p(n)} w \#^{p(n)} \rangle \in \text{HELYBEN ELFOGADÁS}$.

Így $L \leq_p \text{HELYBEN ELFOGADÁS}$ bizonyítva, hogy HELYBEN ELFOGADÁS PSPACE-nehéz. Mivel láttuk, hogy PSPACE-beli, ezért PSPACE-teljes is. \square