Bevezetés a számításelméletbe

7. előadás

előadó: Tichler Krisztián
ktichler@inf.elte.hu

Algoritmikus problémák

(Emlékeztető:)

Algoritmikus eldöntési probléma:

Adott egy $\mathcal{P}$ eldöntendő kérdés (azaz „igen”/„nem” a lehetséges válaszok). A kérdés azon bemeneteit, amelyekre „igen” a válasz „igen”-példányoknak nevezzük.

Olyan mindig termináló, „igen”/„nem” kimenetű algoritmust keresünk, amelynek a bemenete az inputok lehetséges (végtelen) $\mathcal{I}$ halmazának egy eleme, kimenete pedig pontosan akkor „igen”, ha a bemenet $\mathcal{P}$-nek egy „igen”-példánya.

$\mathcal{I}$ lehet például

- $\mathbb{N}$,
- T^*, ahol T egy ábécé
- $\{G \mid G \text{ grammatika}\}$, stb.

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

Algoritmikus 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 egy ilyen matematikai géppel fel tudjuk ismerni $L(\mathcal{P})$-t, akkor a $\mathcal{P}$ problémát algoritmikusan eldöntöttük.

Algoritmikus problémák és matematikai gépek

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

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

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

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

Az algoritmikus fogalmának modelljei

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

- 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.

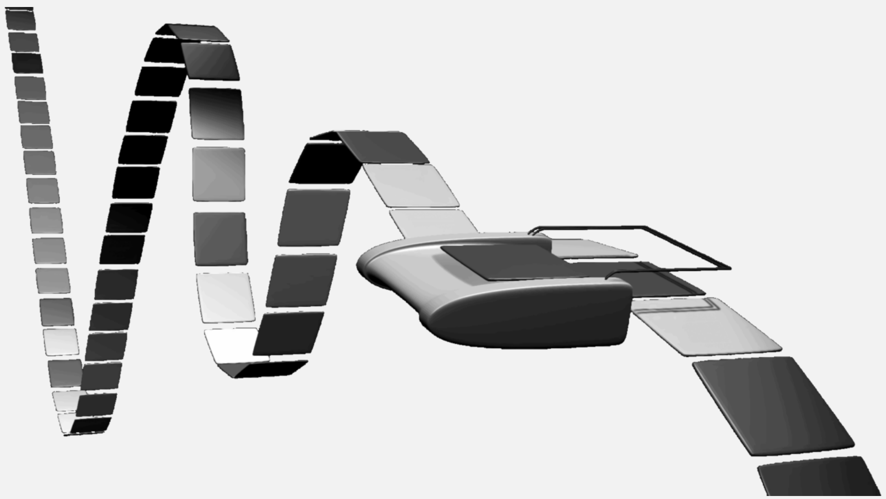
(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

NEM TÉTEL!!!

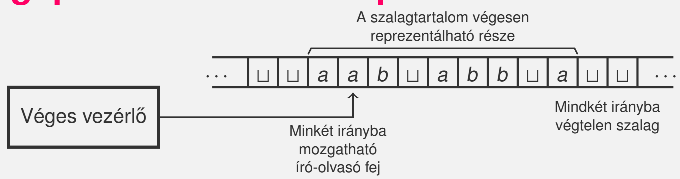
A Church-Turing tézis nem bizonyítható, hiszen nem egy formális matematikai állítás, az algoritmus intuitív fogalmát használja.

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

Turing gépek

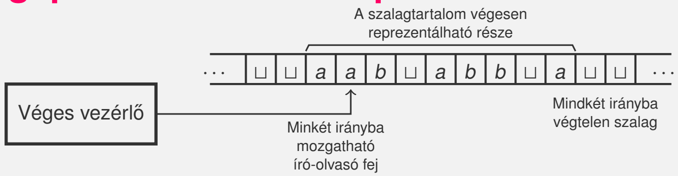


Turing gépek – Informális kép



- 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 $\sqcup \in \Gamma \setminus \Sigma$.
- $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ 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 konfigurácói

A TG működtetését a gép konfigurációival írhatjuk le.

Definíció

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

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

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

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

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

A Turing gépek konfigurácói

A gép egy $u \in \Sigma^*$-beli szóhoz tartozó **kezdőkonfigurációja** a $q_0u\sqcup$ szó. (Vagyis q_0u, 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 $q_0u\sqcup$ és nem q_0u a kezdőkonfiguráció?

Azért, hogy ne legyen két eset. $u = \varepsilon$ esetén ugyanis $q_0u = q_0$ 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 $q_0u\sqcup$ és q_0u ugyanaz a konfiguráció.

Egylépéses konfigurácóátmenet

Jelölje C_M egy M TG-hez tartozó lehetséges konfigurációk halmazát.

Definíció

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

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

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

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

Többlépéses konfigurációátmenet

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

Definíció

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

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

Ekvivalens definíció:
$C \vdash^* C' \Leftrightarrow (C = C') \vee (\exists\, C''$ 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 gépek felismert nyelve

Az M TG által felismert nyelv

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

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

Felismerhetőség és eldönthetőség

Definíció

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

Definíció

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

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

RE és R

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

Definíció

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

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

Nyilván $R \subseteq RE$.

- 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

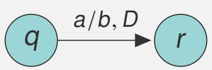
Legyen $f : \mathbb{N} \to \mathbb{N}$ egy függvény. Azt mondjuk, hogy M egy $\boldsymbol{f(n)}$ **időkorlátos** gép (vagy M $f(n)$ időigényű), ha minden $u \in \Sigma^*$ input szóra M futási ideje az u szón legfeljebb $f(|u|)$.

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

Turing gépek – Példa

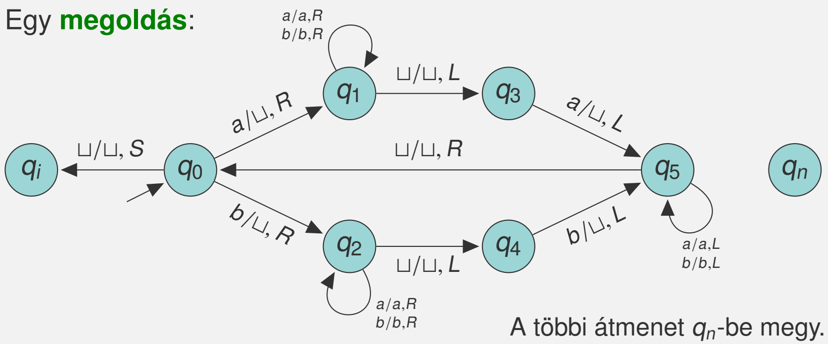
Feladat: Készítsünk egy M Turing gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Az **átmenetdiagram**.



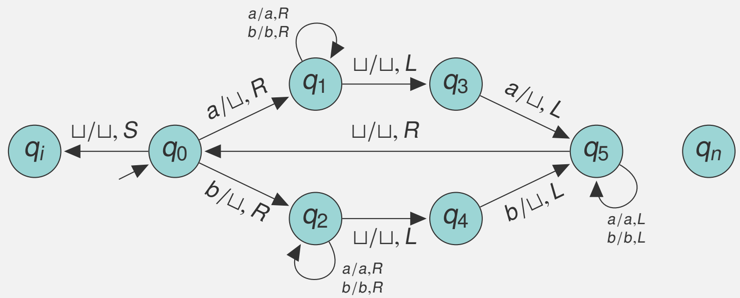
megfelel a $\delta(q, a) = (r, b, D)$ átmenetnek
$(q, r \in Q, a, b \in \Gamma, D \in \{L, S, R\})$

Egy **megoldás**:



A többi átmenet q_n-be megy.

Turing gépek – Példa (folyt.)



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

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.

Függvények aszimptotikus nagyságrendje

Legyenek $f, g : \mathbb{N} \to \mathbb{R}_0^+$ 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: $f(n)$ = nagyordó $g(n)$) 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 $\mathbb{N} \to \mathbb{R}_0^+$ 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 $\mathbb{N} \to \mathbb{R}_0^+$ 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 $f(n)/g(n) \to +\infty \Rightarrow f(n) = \Omega(g(n))$ és $f(n) \neq O(g(n))$

  ha $f(n)/g(n) \to c \quad (c > 0) \Rightarrow f(n) = \Theta(g(n))$

  ha $f(n)/g(n) \to 0 \quad \Rightarrow f(n) = O(g(n))$ és $f(n) \neq \Omega(g(n))$

Függvények aszimptotikus nagyságrendje

- $p(n) = a_k n^k + \cdots + a_1 n + a_0$ ($a_k > 0$), 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:

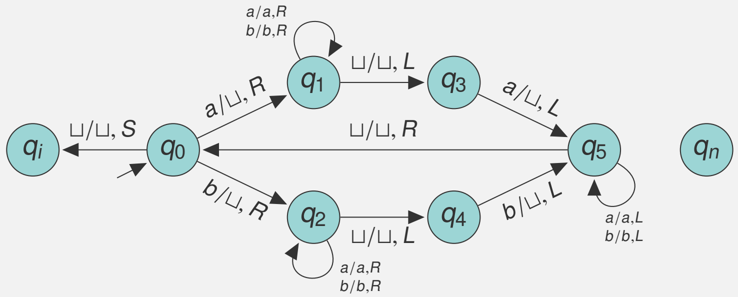
A jelölés Edmund Landau német matematikustól származik.

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.

Turing gépek – Példa (folyt.)



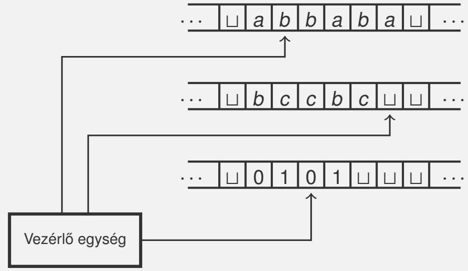
A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, +1 lépés q_i-be vagy q_n-be.

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

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

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

Többszalagos Turing gép – Informális kép



- 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 $\sqcup \in \Gamma \setminus \Sigma$,
- $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^k \to Q \times \Gamma^k \times \{L, S, R\}^k$ 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 $(q, u_1, v_1, \ldots, u_k, v_k)$ 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**: $u_i = \varepsilon$ $(1 \leq i \leq k)$, $v_1 = u\sqcup$, 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 $(q, u_1, v_1, \ldots, u_k, v_k)$ 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-ek egylépéses konfigurációátmenete

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 A $C = (q, u_1, a_1v_1, \ldots, u_k, a_kv_k)$ egy konfiguráció, ahol $a_i \in \Gamma$, $u_i, v_i \in \Gamma^*$ $(1 \leq i \leq k)$. Legyen továbbá $\delta(q, a_1, \ldots, a_k) = (r, b_1, \ldots, b_k, D_1, \ldots, D_k)$, ahol $q, r \in Q$, $b_i \in \Gamma, D_i \in \{L, S, R\}$ $(1 \leq i \leq k)$. Ekkor $C \vdash (r, u'_1, v'_1, \ldots, u'_k, v'_k)$, ahol minden $1 \leq i \leq k$-ra

- ha $D_i = R$, akkor $u'_i = u_ib_i$ és $v'_i = v_i$, ha $v_i \neq \varepsilon$, különben $v'_i = \sqcup$,
- ha $D_i = S$, akkor $u'_i = u_i$ és $v'_i = b_iv_i$,
- ha $D_i = L$, akkor $u_i = u'_ic$ $(c \in \Gamma)$ és $v'_i = cb_iv_i$ ha $u_i \neq \varepsilon$, különben $u'_i = \varepsilon$ és $v'_i = \sqcup b_iv_i$.

k-szalagos TG-ek többlépéses konfigurációátmenete

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

Példa:

Legyen k=2 és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v'_1, u_2, b_2v_2)$, ahol $v'_1 = v_1$, ha $v_1 \neq \varepsilon$, különben $v'_1 = \sqcup$.

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

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

k-szalagos TG által felismert nyelv, a TG időigénye

k-szalagos Turing gép által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \ldots x_k, y_k)$, valamely $x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots, y_k \neq \varepsilon$ -ra$\}$.

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

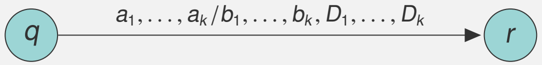
k-szalagos Turing gép futási ideje adott szóra

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

Ezek után az **időigény** definíciója megegyezik az egyszalagos esetnél tárgyalttal.

k-szalagos Turing gép – átmenetdiagram

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

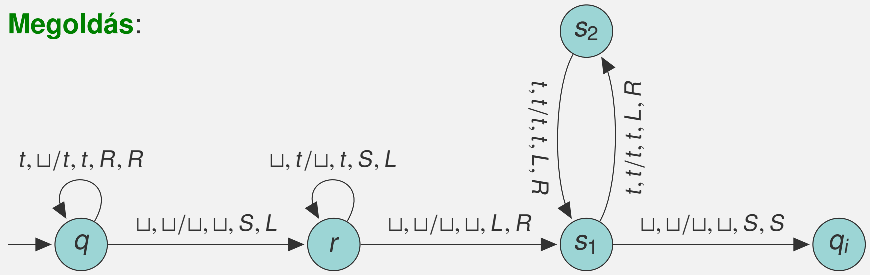


$$\iff \delta(q, a_1, \ldots, a_k) = (r, b_1, \ldots, b_k, D_1, \ldots, D_k)$$
$$(q, r \in Q, a_1, \ldots, a_k, b_1, \ldots, b_k \in \Gamma, D_1, \ldots, D_k \in \{L, S, R\})$$

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

k-szalagos Turing gép – példa

Megoldás:



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

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

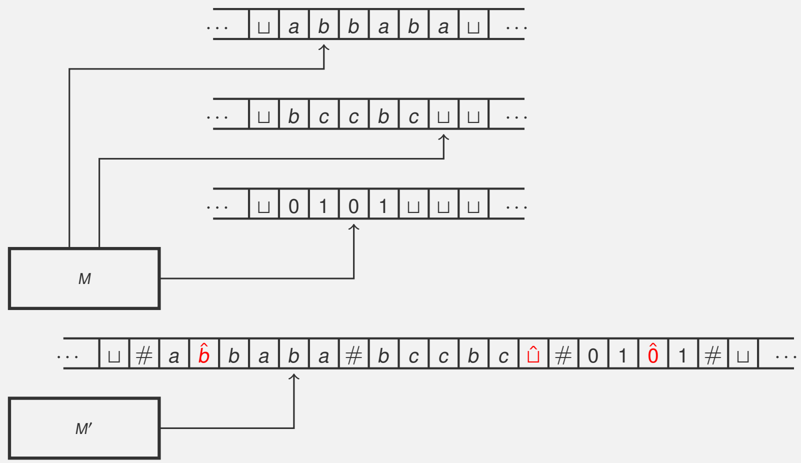
Két TG **ekvivalens**, ha ugyanazt a nyelvet ismerik fel.

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 $\hat{}$ -pal megjelölt szimbólumokat az állapotában (Például az ábrán látható esetben ha M a q állapotában van akkor M' a $(q), (q, b), (q, b, \sqcup)$ állapotokon keresztül a $(q, b, \sqcup, 0)$ állapotába kerül.)
3. M' mégegyszer végigmegy a szalagján és M átmenetfüggvénye alapján aktualizálja azt
4. ha M valamelyik szalagján nő a szalagtartalmat leíró szó hossza, akkor M'-nek az adott ponttól egy cellával jobbra kell mozgatnia a szalagja tartalmát, hogy legyen hely az új betű számára
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(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

- 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.

Tétel

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

Turing gép egy irányban végtelen szalaggal

Tétel

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

Bizonyítás (vázlat):

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

Bevezetés a számításelméletbe

8. előadás

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 $\sqcup \in \Gamma \setminus \Sigma$,
- $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, S, R\})$.

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 egylépéses konfigurácóátmenete

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 $\exists\, n > 0 \wedge C_1, C_2, \ldots C_n \in C_M$, 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**
$L(M) = \left\{u \in \Sigma^* \mid q_0u\sqcup \vdash^* xq_iy \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon \text{ -ra}\right\}$.

Bár a definíció formálisan megegyezik a determinisztikus TG által felismert nyelv definíciójával az egylépéses 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.

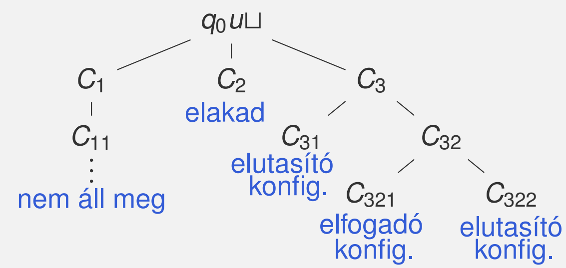
Egy NTG-nek azonban **több számítása is lehet ugyanarra a szóra**. Ezek között lehetnek elfogadó és elutasító (sőt nem termináló!) számítások is. Egy NTG akkor fogad el egy szót, ha az adott szóra **legalább egy számítása q_i-ben ér véget** (hiszen ekkor a kezdőkonfiguráció és ez az elfogadó konfiguráció $\vdash^*$ 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. $q_0u\sqcup$ 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:



Tehát M elfogadja u-t, hiszen a $q_0u\sqcup \vdash C_3 \vdash C_{32} \vdash C_{321}$ számítása elfogadó konfigurációba visz. **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

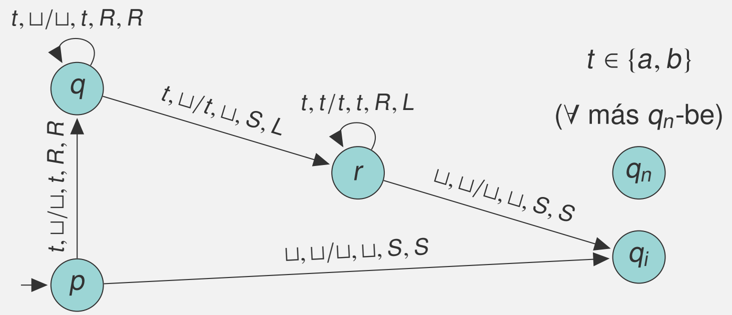
Az M NTG **$f(n)$ időkorlátos** (időigényű), ha minden $u \in \Sigma^*$ n hosszú szóra u számítási fája legfeljebb $f(n)$ magas.

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

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

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.

Nemdeterminisztikus Turing gép

Szimulálás determinisztikus TG-pel

Tétel

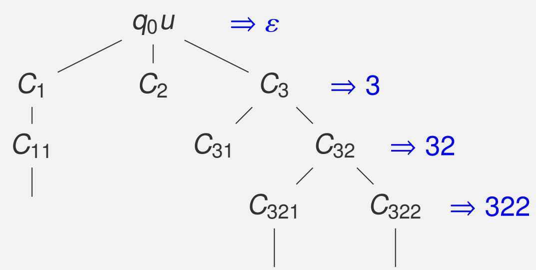
Minden $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ $f(n)$ időkorlátos NTG-hez megadható egy ekvivalens, $2^{O(f(n))}$ időkorlátos M' determinisztikus TG.

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 $T = \{1, 2, \ldots, d\}$ egy (rendezett) ábécé.
- minden $(q, a) \in Q \times \Gamma$ esetén rögzítsük le $\delta(q, a)$ elemeinek egy sorrendjét

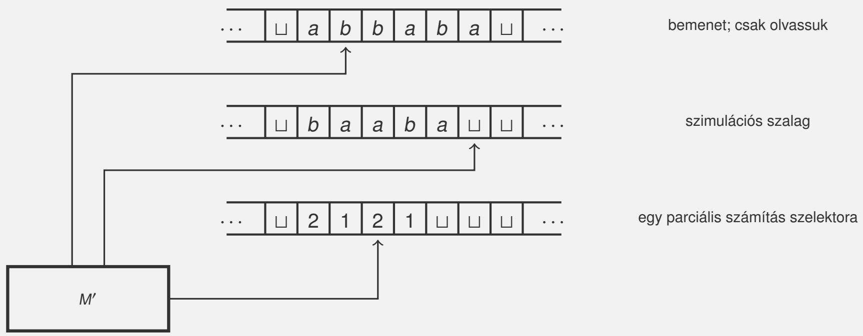
NTG szimulálása determinisztikus TG-pel

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 $wi \in T^*$.

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 van k-ik 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
    - 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 ($\leq$ egy $f(n)$ magasságú teljes d-áris fa belső csúcsainak száma darabot, ami $O(d^{f(n)})$). Mivel minden számítás legfeljebb $f(n)$ lépésből áll, így M' $f(n)O(d^{f(n)}) = 2^{O(f(n))}$ időkorlátos.

Megjegyzés:

- Abból, hogy a bizonyításban alkalmazott szimuláció exponenciális időigényű még nem következik, hogy nincs hatékonyabb szimuláció.
- Az a *sejtés*, hogy nem lehet a nemdeterminisztikus Turing gépet az időigény drasztikus romlása nélkül determinisztikus Turing géppel szimulálni.

Számosság

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

Definíció

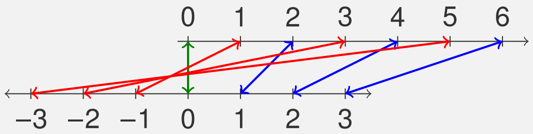
- A és B halmazoknak **megegyezik a számosságuk**, ha $\exists$ bijekció köztük. Jelölése: $|A| = |B|$.
- A-nak **legalább annyi a számossága**, mint B-nek, ha $\exists$ B-ből injekció A-ba. Jelölése: $|A| \geq |B|$.
- A-nak **nagyobb a számossága, mint B-nek**, ha $\exists$ B-ből A-ba injekció, de $\nexists$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

Cantor-Bernstein-Schröder tétel

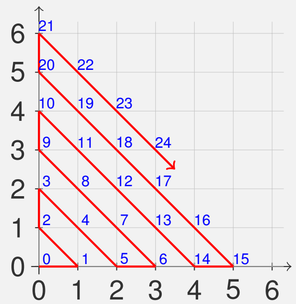
Ha $\exists$ injekció A-ból B-be és B-ből A-ba is, akkor $\exists$ bijekció A és B között, azaz ha $|A| \leq |B|$ és $|A| \geq |B|$, akkor $|A| = |B|$.

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 $|\mathbb{N}| \leq |\mathbb{Q}|$.

$\mathbb{Q}^+ := \{\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+,$ a tört nem egyszerűsíthető$\}$.
$\mathbb{Q}^- := \{-\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+,$ 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 $\mathbb{Q}^+ = \{a_1, a_2 \ldots, \}$, $\mathbb{Q}^- = \{b_1, b_2 \ldots, \}$, ekkor
$\mathbb{Q} = \{0, a_1, b_1, a_2, b_2, \ldots\}$.

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.

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: $\mathrm{tg}(\pi(x - \frac{1}{2}))\big|_{(0,1)} : (0, 1) \to \mathbb{R}$ 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:
ε,0,1,00,01,10,11,000,001,010,011,100,101,110,111,0000,...

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

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

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

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 $\mathbf{b}_L = (b_1, \ldots, b_i, \ldots)$ 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...=0,0011...)

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

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

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

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

$H_0 \subset \{0,1\}^{\mathbb{N}}$, 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 $\{0,1\}^{\mathbb{N}} = \{u_i \mid i \in \mathbb{N}\} = \{u_1, u_2, \ldots\}$ a $\{0,1\}^{\mathbb{N}}$ elemeinek egy felsorolása (a természetes számokkal való megindexelése).

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

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

Tekintsük az $u = (\overline{u_{1,1}}, \overline{u_{2,2}}, \ldots, \overline{u_{i,i}}, \ldots)$ 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 is 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 foglaljuk össze amit tudunk:
$|\mathbb{R}| = |[0,1)| = |\{0,1\}^{\mathbb{N}}| = \left|\left\{L \mid L \subseteq \{0,1\}^*\right\}\right| > |\mathbb{N}| = |\{0,1\}^*|$.

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

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. Definiálunk egy $A \subseteq H$ halmazt: $\forall x \in H : \quad x :\in A \;\Leftrightarrow\; x \notin f^{-1}(x)$

$f(A) \in A$ 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.

$\aleph_0 := |\mathbb{N}|, \aleph_1 := |\mathcal{P}(\mathbb{N})| = |\mathbb{R}|$. Ha $|H| = \aleph_i$ akkor $\aleph_{i+1} := |\mathcal{P}(H)|$.

A Turing gépek egy elkódolása

Tegyük fel, hogy $\Sigma = \{0, 1\}$. Ez feltehető, mivel minden input hatékonyan kódolható Σ felett.

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

- $Q = \{p_1, \ldots, p_k\},\ \Gamma = \{X_1, \ldots, X_m\},\ D_1 = R,\ D_2 = S, D_3 = L$
- $k \geq 3,\ p_1 = q_0,\ p_{k-1} = q_i,\ p_k = q_n,$
- $m \geq 3,\ X_1 = 0,\ X_2 = 1,\ X_3 = \sqcup.$
- Egy $\delta(p_i, X_j) = (p_r, X_s, D_t)$ átmenet kódja $0^i10^j10^r10^s10^t$.
- $\langle M\rangle$ az átmenetek kódjainak felsorolása 11-el elválasztva.

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

Definíció

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

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

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

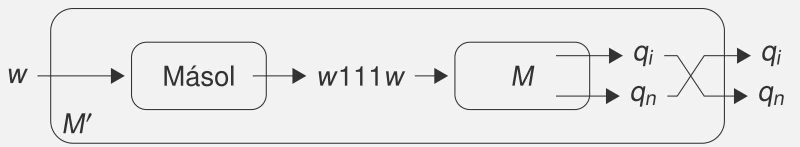
Az univerzális TG
Eldönthetetlenség

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.

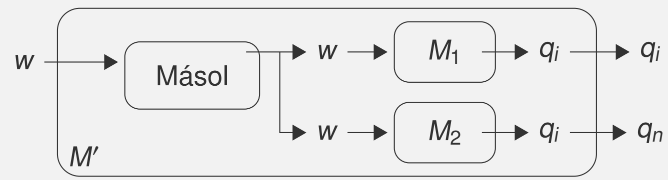
RE és R tulajdonságai

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

Tétel

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

Bizonyítás: Legyen M_1 és M_2 rendre az L-t és $\overline{L}$-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

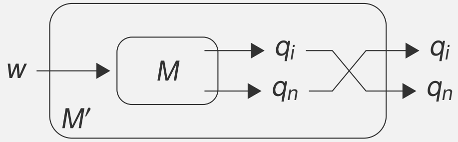
RE nem zárt a komplementer-képzésre.

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

Tétel

Ha $L \in R$, akkor $\overline{L} \in R$. (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' $\overline{L}$-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 $f : \Sigma^* \to \Delta^*$ szófüggvényt, 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:
$f(u) = ub$ $(u \in \{a, b\}^*)$.

Visszavezetés

Definíció

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

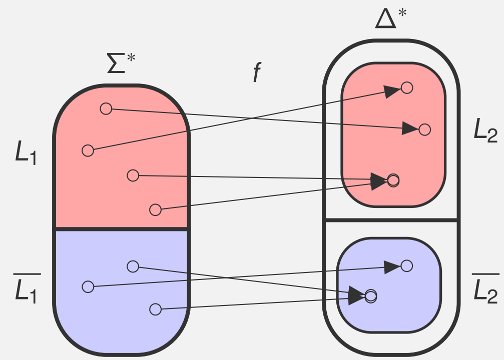
Definíció

$L_1 \subseteq \Sigma^*$ **visszavezethető** $L_2 \subseteq \Delta^*$-ra, ha van olyan $f : \Sigma^* \to \Delta^*$ 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

$L_1 \leq L_2$



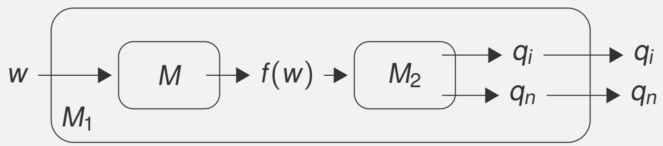
f kiszámítható, az egész Σ^*-on értelmezett, $f(L_1) \subseteq L_2$ valamint $f(\overline{L_1}) \subseteq \overline{L_2}$. f nem kell hogy injektív legyen és az se kell, hogy szürjektív.

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 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

- $f : \langle M, w\rangle \to \langle M', w\rangle$ 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:

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

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 következménye 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

- $f : \langle M, w\rangle \to \langle M', w\rangle$ 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.

Bevezetés a számításelméletbe

9. előadás

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

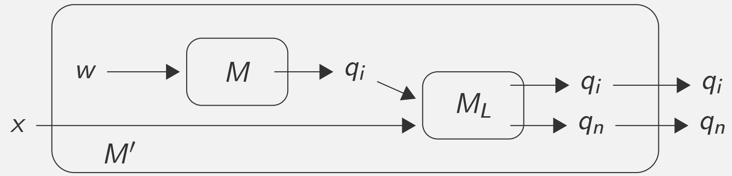
Bizonyítás:

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 $L_{\overline{\mathcal{P}}} \notin R$.
- $\overline{L_{\mathcal{P}}} = L_{\overline{\mathcal{P}}}$, mivel megállapodásunk szerint minden szó TG kód, a nem kellő alakú szavak egy rögzített, egyetlen szót sem elfogadó TG-et kódolnak.
- $\overline{L_{\mathcal{P}}} \notin R \Rightarrow L_{\mathcal{P}} \notin R$ (tétel volt).

Rice tétel

Alkalmazások

Következmények:
Eldönthetetlen, hogy egy M TG

- az üres nyelvet ismeri-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 $u_1, \ldots, u_n, v_1 \ldots, v_n \in \Sigma^+$ $(n \geq 1)$. A $D = \left\{\frac{u_1}{v_1}, \ldots, \frac{u_n}{v_n}\right\}$ 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 $\frac{u_{i_1}}{v_{i_1}} \cdots \frac{u_{i_m}}{v_{i_m}}$ dominósorozat $(m \geq 1, 1 \leq i_1, \ldots, i_m \leq n)$ a $D = \{\frac{u_1}{v_1}, \ldots, \frac{u_n}{v_n}\}$ dominókészlet egy **megoldása**, ha $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$.

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 keresése esetén egy végtelen keresési térrel állunk szemben.

Post Megfelelkezési Probléma (PMP):

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

Post Megfelelkezési Probléma

Tétel

$L_{\mathrm{PMP}} \in RE$.

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

$L_{\mathrm{PMP}} \notin R$.

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.

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

Post Megfelelkezési Probléma

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

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

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

balcsillag$(u) := * a_1 * a_2 \cdots * a_n$
jobbcsillag$(u) := a_1 * a_2 * \cdots a_n *$.
baljobbcsillag$(u) := * a_1 * a_2 * \cdots a_n *$.

Legyen $D = \{d_1, \ldots, d_n\}$ egy tetszőleges dominókészlet, ahol $d_i = \frac{u_i}{v_i}$ $(1 \leq i \leq n)$.
D' legyen a következő $|D| + 2$ méretű készlet:

$$d_i' = \frac{\text{balcsillag}(u_i)}{\text{jobbcsillag}(v_i)} \qquad (1 \leq i \leq n)$$
$$d_0' = \frac{\text{balcsillag}(u_1)}{\text{baljobbcsillag}(v_1)}, \qquad d_{n+1}' = \frac{*\#}{\#}$$
.

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

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 $L_{\mathrm{MPMP}} \leq L_{\mathrm{PMP}}$.

Post Megfelelkezési Probléma

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

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

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

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 $q_0 bab \vdash aq_2 ab \vdash aq_i bb$ egy bab-ot elfogadó konfigurációátmenet.

Az $\langle M, bab\rangle$-hoz tartozó dominókészlet tartalmazza többek között a $\frac{\#}{\#q_0 bab\#}$ kezdő-, $\frac{q_0 b}{aq_2}$ és $\frac{q_2 a}{q_i b}$ 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):

$$\frac{\#}{\#q_0 bab\#} \;\Big|\; \frac{q_0 b}{aq_2}\frac{a}{a}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{a}{a}\frac{q_2 a}{q_i b}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{aq_i}{q_i}\frac{b}{b}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{q_i b}{q_i}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{q_i b}{q_i}\frac{\#}{\#} \;\Big|\; \frac{q_i\#\#}{\#}$$

Post Megfelelkezési Probléma

$$\frac{\#}{\#q_0 bab\#} \;\Big|\; \frac{q_0 b}{aq_2}\frac{a}{a}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{a}{a}\frac{q_2 a}{q_i b}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{aq_i}{q_i}\frac{b}{b}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{q_i b}{q_i}\frac{b}{b}\frac{\#}{\#} \;\Big|\; \frac{q_i b}{q_i}\frac{\#}{\#} \;\Big|\; \frac{q_i\#\#}{\#}$$

A fenti példán szemléltetjük, hogy $w \in L(M) \iff D$-nek van d-vel kezdődő megoldása.

Az első blokk csak a $d = \frac{\#}{\#q_0 bab\#}$ 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.

Post Megfelelkezési Probléma

A fenti példa alapján meg lehet általános esetben is konstruálni egy megoldást, így $w \in L(M) \Rightarrow$ van $\langle D, d\rangle$-nak megoldása.

Másrészt ha van d-vel kezdődő megoldás, akkor ez a dominósorozat két szavának hosszára vonatkozó megfontolások alapján csak q_i-t tartalmazó dominók használatával lehetséges. Meggondolható, hogy minden kirakás alsó szava az első q_i-t követő $\#$-ig egy $\#$-ekkel elválasztott elfogadó konfigurációátmenet sorozata kell legyen. és így a w szóhoz tartozó kezdőkonfigurációból el lehet jutni elfogadó konfigurációba, azaz $w \in L(M)$.

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

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

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.)

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

Tétel

$L_{\mathrm{ECF}} \notin R$

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

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

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

$P_A := \{A \to u_1 A a_1, \ldots, A \to u_n A a_n,\ A \to \varepsilon\}$.
$P_B := \{B \to v_1 B a_1, \ldots, B \to v_n B a_n,\ B \to \varepsilon\}$.

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

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

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

- ha $\frac{u_{i_1}}{v_{i_1}} \cdots \frac{u_{i_m}}{v_{i_m}}$ megoldása D-nek, akkor $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$. De ekkor $u_{i_1} \cdots u_{i_m} a_{i_m} \cdots a_{i_1} = v_{i_1} \cdots v_{i_m} a_{i_m} \cdots a_{i_1}$ kétféleképpen is levezethető, így G_D nem egyértelmű.
- ha G_D nem egyértelmű, akkor van olyan szó, aminek két baloldali levezetése van. De ezek $S \to A$-val illetve $S \to B$-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 $L_{\text{PMP}} \notin R$, következik, hogy $\overline{L_{\text{ECF}}} \notin R$, amiből kapjuk, hogy $L_{\text{ECF}} \notin R$.

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

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

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

és M_δ:

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

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

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

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

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

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

$\#r \to \#s$

Az veremautomata determinisztikus (teljessé tehető egy zsákutcaállapot és a hiányzó átmenetek hozzávételével.) A determisztikus veremautomatával felismerhető nyelvek osztálya zárt a komplementerképzésre, ugyanis $Q \setminus F$-re változtatva az elfogadó állapothalmazt a veremautomata épp a komplementer nyelvet ismeri fel.

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 $D = \left\{\frac{u_1}{v_1}, \ldots, \frac{u_n}{v_n}\right\}$ 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) $L := \overline{L(G_A) \cap L(G_A)} = \overline{L(G_A)} \cup \overline{L(G_B)} \in \mathcal{L}_2$, 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 (1) is az lenne, de az imént láttuk, hogy nem az.

(3) Legyen G_1 ugyanaz,mint (2)-ben és $\Gamma = \Sigma \cup \Delta$. Pont úgy, mint előbb, (3) eldönthetősége (1) 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 matematikai logika formális modelljei

I. **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."
(3) Tehát „Lemegyek a térre."

Ítéletkalkulus

Definíció

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

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

A műveleti jelek elnevezése: **negáció** ($\neg$), **konjukció** ($\wedge$), **diszjunkció** ($\vee$), **implikáció** ($\to$). Ez egyben egy csökkenő precedenciasorrend is zárójelek elhagyásához.

Szemantika:

Definíció

Egy $I : \mathrm{Var} \to \{i, h\}$ függvényt **interpretációnak** (változókiértékelésnek) nevezünk.

Ítéletkalkulus

Egy I interpretációban egy $\varphi \in \mathrm{Form}$ formula $\mathcal{B}_I(\varphi)$ **igazságértékét** (Boole értékét) a következő rekurzóval definiáljuk:

Definíció

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

ahol a műveleteket az alábbi táblázat definiálja.

$\mathcal{B}_I(\varphi)$	$\mathcal{B}_I(\psi)$	$\mathcal{B}_I(\neg\varphi)$	$\mathcal{B}_I(\varphi \wedge \psi)$	$\mathcal{B}_I(\varphi \vee \psi)$	$\mathcal{B}_I(\varphi \to \psi)$
i	i	h	i	i	i
i	h	h	h	i	h
h	i	i	h	i	i
h	h	i	h	h	i

Ítéletkalkulus

Egy formula igazságértéke csak a benne szereplő ítéletváltozók kiértékelésétől függ.

n ítéletváltozó esetén 2^n lehetséges interpretáció van (ha nem törődünk a formulában nem szereplő ítéletváltozók kiértékelésével).

Definíció

Egy φ ítéletlogikai formula **igazságtáblája** egy $2^n \times (n+1)$-es táblázat, ha $x_1, \ldots, x_n$ a φ formulában szereplő ítéletváltozók. A sorok megfelelnek a lehetséges interpretációknak. Az első n oszlop tartalmazza az ítéletváltozók kiértékelését. Egy I interpretációhoz tartozó sor $n+1$. oszlopa pedig $\mathcal{B}_I(\varphi)$-t.

Példa:

x	y	$\neg x \vee y$
i	i	i
i	h	h
h	i	i
h	h	i

Ítéletkalkulus

Definíció

- Egy $\mathcal{I}$ 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.

Példák: $\varphi \to \psi \sim_0 \neg\varphi \vee \psi$, $\quad \neg(\varphi \wedge \psi) \sim_0 \neg\varphi \vee \neg\psi$ (De Morgan)

Ítéletkalkulus

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: $\{x, x \to y\} \models_0 y$

Ítéletkalkulus

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.

Ítéletkalkulus

Definíció

- **Literálnak** nevezünk egy x vagy $\neg x$ alakú formulát, ahol $x \in \mathrm{Var}$. Egy literál **alapja** az az ítéletváltozó, amelyik a literálban szerepel.
- **Klóznak** hívunk egy $\ell_1 \vee \cdots \vee \ell_n$ alakú formulát ($n \in \mathbb{N}$), ahol $\ell_1, \ldots \ell_n$ páronként különböző alapú literálok.
- **Konjunktív normálformának** (röviden KNF-nek) nevezünk egy $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ ($m \geq 1$) alakú formulát, ahol minden $1 \leq i \leq m$-re C_i egy klóz (a KNF egy **tagja**).

Példa:

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

Tétel

Minden ítéletkalkulusbeli formulához megadható egy vele tautologikusan ekvivalens KNF.

Eldönthető problémák a nulladrendű logikában

Állítás: Eldönthetőek 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 ítélettáblákat a szóban forgó formulákra és olvassuk le belőlük.

Megjegyzés: A kérdések eldönthetősége valójában azon múlik, hogy véges sok lehetséges interpretáció van, így megoldhatóak „brute force” módszerrel. Mivel n ítéletváltozó esetén az ítélettáblának 2^n, azaz exponenciális sok sora van, ez nem hatékony. Ugyan ismeretesek az ítélettáblánál jobb módszerek, azonban ezek mindegyike a legrosszabb esetben szintén exponenciális műveletigényű.

A matematikai logika formális modelljei

II. **Elsőrendű logika**

A nulladrendű logika korlátozottan alkalmas a világ leírására, az egyszerű állítások belső szerkezetét nem vizsgálja. Például a „Minden ember halandó.”, „Szókrátész ember.”, „Szókrátész halandó.” állítások nulladrendű formalizálása esetén nincs más lehetőségünk, mint x, y és z-ként formalizálni a fenti állítás-hármast. Ugyanakkor mivel az emberek halmaza részhalmaz a halandók halmazának és Szókrátész az ember-halmaz egy eleme, ezért jó lenne egy olyan modell, ahol a 3. állítás az első 2 következménye.

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

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 **függvényszimbólumok** véges halmaza,
- Cnst, a **konstansszimbólumok** véges halmaza,
- $\mathrm{Ind} = \{x_1, x_2, \ldots\}$, 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 $s \in \mathrm{Pred} \cup \mathrm{Func} \cup \mathrm{Cnst}$-hez hozzá van rendelve egy $\mathrm{ar}(s) \in \mathbb{N}$ szám, a szimbólum **aritása** (a konstansokhoz mindig 0).

Elsőrendű logika

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 $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi) \in \text{Form}$.
- Ha $\varphi \in \text{Form}$, akkor $\forall x\varphi \in \text{Form}$ és $\exists x\varphi \in \text{Form}$.

Elsőrendű logika

Példa

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

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

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

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

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

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

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

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

Elsőrendű logika

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 $f^I : U^{\text{ar}(p)} \to U$ $\text{ar}(p)$-változós műveletet U-n,
- I_{Cnst} minden $c \in \text{Cnst}$-hez hozzárendel egy $c^I \in U$-t.

Definíció

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

Vegyük észre, hogy κ függ az U univerzumtól.

Elsőrendű logika

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

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:

- $|p(t_1, t_2, \ldots, t_{\text{ar}(p)})|^{I,\kappa} = i \Leftrightarrow$ $(|t_1|^{I,\kappa}, |t_2|^{I,\kappa}, \ldots, |t_{\text{ar}(p)}|^{I,\kappa}) \in p^I$,
- $|\neg\varphi|^{I,\kappa} := \neg|\varphi|^{I,\kappa}$
- $|\varphi \circ \psi|^{I,\kappa} := |\varphi|^{I,\kappa} \circ |\psi|^{I,\kappa}$ $\qquad \circ \in \{\wedge, \vee, \rightarrow\}$
- $|\forall x\varphi|^{I,\kappa} = i \Leftrightarrow$ 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

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. $\quad |\varphi_1|^{I,\kappa} = i$,

$\varphi_2 = \forall x \exists y q(f(x, y), a)$,
Minden természetes számhoz hozzá tudjuk adni egy természetes számot úgy, hogy 0-t kapjunk. $\quad |\varphi_2|^{I,\kappa} = h$,

$\varphi_3 = \forall x(\forall y q(f(y, x), y) \rightarrow p(x, a))$,
Ha a természetes számoknak van nulleleme, akkor az egyenlő 0-val, $|\varphi_3|^{I,\kappa} = i$.

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

Elsőrendű logika

Definíció

Legyen φ egy formula, és tekintsük $x \in \text{Ind}$ egy előfordulását φ-ben. 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 kormula **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 φ_1, φ_2, φ_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: $\forall x p(x, x) \rightarrow q(x, x)$, $\forall x p(x, x)$, $p(x, x)$, $q(x, x)$.)

Elsőrendű logika

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.

Eldönthetetlen problémák az elsőrendű logikában

Tétel

Legyen $\textsc{ValidityPred} = \{\langle\varphi\rangle \mid \varphi \text{ logikailag igaz}\}$. Ekkor

(1) $\textsc{ValidityPred} \notin \mathrm{R}$

Nem bizonyítjuk, a bizonyítás megtalálható Gazdag Zsolt: Bevezetés a számításelméletbe elektronikus jegyzetének 61. oldalán.

A bizonyítás egyébként L_{PMP}-t vezeti vissza a fenti problémának megfelelő formális nyelvre. Azaz minden D dominókészlethez megadható egy φ_D elsőrendű formula, amelyre teljesül, hogy D-nek akkor és csak akkor van megoldása, ha $\models \varphi_D$.

(Vagyis egy dominókészlet megoldásának fogalmát formálisan le lehet írni egy elsőrendű logikai formulával.)

Eldönthetetlen problémák az elsőrendű logikában

Következmény

Legyen $\mathcal{F}$ egy elsőrendű formulahalmaz és φ egy elsőrendű formula. Eldönthetetlen, hogy

(2) φ kielégíthetetlen-e

(3) φ kielégíthető-e

(4) $\mathcal{F} \models \varphi$ teljesül-e

Bizonyítás:

(2) $\models \neg\varphi \Leftrightarrow \varphi$ kielégíthetetlen.

(3) Eldönthetetlen nyelv komplementere eldönthetetlen.

(4) $\mathcal{F} \models \varphi \Leftrightarrow \mathcal{F} \cup \{\neg\varphi\}$ kielégíthetetlen

Megjegyzés: Van olyan algoritmus, amely egy tetszőleges φ elsőrendű formulára pontosan akkor áll meg igen válasszal, ha φ kielégíthetetlen (ilyen például az elsőrendű logika rezolúciós algoritmusa). Ezért a kielégíthetetlenség rekurzíve felsorolható. $\Rightarrow$ a kielégíthetőség nem rekurzíve felsorolható.

Bevezetés a számításelméletbe

10. előadás

BONYOLULTSÁGELMÉLET

A továbbiakban eldönthető problémákkal foglalkozunk, ilyenkor a kérdés az, hogy milyen idő illetve tár tekintetében milyen hatékonyan dönthető el az adott probléma.

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

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

Definíció

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

Észrevétel: $\mathrm{P} \subseteq \mathrm{NP}$, mivel minden determinisztikus TG tekinthető nemdeterminisztikusnak.

Sejtés: $\mathrm{P} \neq \mathrm{NP}$ (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 $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ 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 őt polinom időben eldöntő NTG ami gyakran a következőképpen működik: a probléma minden I bemenetére polinom időben „megsejti" (azaz nemdeterminisztikusan generálja) I egy lehetséges m megoldását és polinom időben leellenőrzi (már determinisztikusan), hogy m alapján $I \in L$ teljesül-e. A NTG definíciója szerint elég egyetlen ilyen megoldást, „tanút" megsejteni.

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

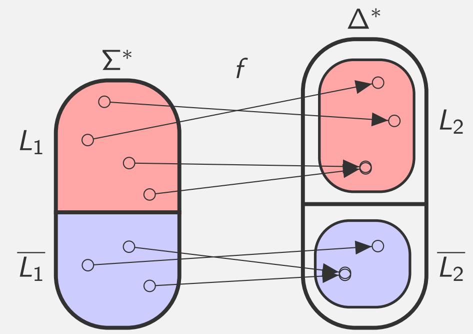
Definíció

$L_1 \subseteq \Sigma^*$ **polinom időben visszavezethető** $L_2 \subseteq \Delta^*$-ra, ha van olyan $f : \Sigma^* \to \Delta^*$ 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 $L_2 \in \mathrm{P}$, akkor $L_1 \in \mathrm{P}$.
- Ha $L_1 \leq_p L_2$ és $L_2 \in \mathrm{NP}$, akkor $L_1 \in \mathrm{NP}$.

Polinom idejű visszavezetés

Bizonyítás:

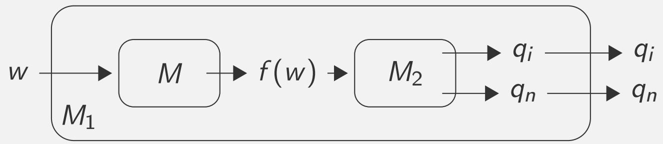
Az elsőt bizonyítjuk, a második analóg.

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

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

Definíció

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

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

NP-teljesség

Ha speciálisan $\mathcal{C}$=NP:

NP-teljes nyelv

Egy L nyelv NP-teljes (a polinom idejű visszavezetésre nézve), ha

- $L \in$ NP
- L NP-nehéz, azaz minden $L' \in$ NP esetén $L' \leq_p L$.

Megjegyzés: Néha úgy fogalmazunk, hogy az L (eldöntési) probléma NP-teljes...

Tétel

Legyen L egy NP-teljes probléma. Ha $L \in$ P, akkor P = NP.

Bizonyítás: Elég megmutatni, hogy NP $\subseteq$ P.
Legyen $L' \in$NP egy tetszőleges probléma.
Ekkor $L' \leq_p L$, hiszen L NP-teljes.
Mivel $L \in$P, ezért az előző tétel alapján $L' \in$P.
Ez minden $L' \in$NP-re elmondható, ezért NP $\subseteq$ P.

Egy NP-teljes nyelv

Ha $L \leq_p L'$, akkor intuitíve L' legalább olyan nehéz, mint L. Így az NP-teljes problémák (ha vannak) az NP-beli problémák legnehezebbjei.

Az előző tétel szerint tehát, ha valaki talál egy NP-teljes problémára polinom idejű determinisztikus algoritmust, azzal bizonyítja, hogy P=NP.

Ezért nyilván egyetlen NP-teljes problémára sem ismeretes jelenleg polinomiális idejű determinisztikus algoritmus és nem túl valószínű, hogy valaha is fogunk ilyet találni. Így a gyakorlatban az NP-teljes problémákra úgy is tekinthetünk, mint bár eldönthető, de hatékonyan nem eldönthető problémákra.

Definíció

$\mathrm{SAT} := \{\langle\varphi\rangle \mid \varphi$ kielégíthető nulladrendű KNF$\}$

Cook-Levin tétel

SAT NP-teljes.

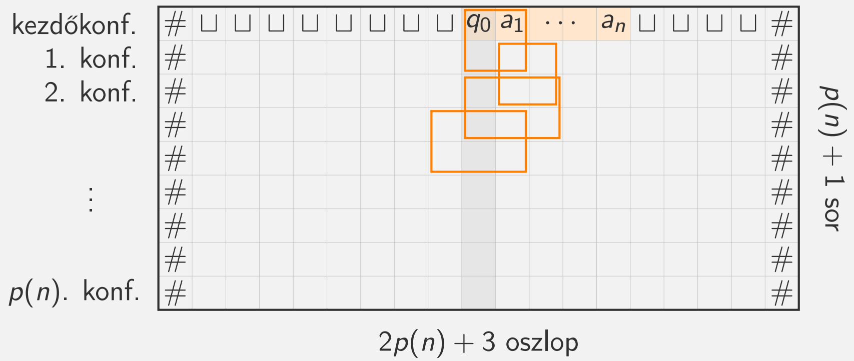
A Cook-Levin tétel bizonyítása

Bizonyítás:

- SAT$\in$ NP: Adott egy φ input. Egy NTG egy számítási ágán polinom időben előállít egy I interpretációt. Majd szintén polinom időben ellenőrzi, hogy ez kielégíti-e φ-t.
- SAT NP-nehéz: ehhez kell, $L \leq_p$ SAT, minden $L \in$ NP-re.
  - Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n\rangle$ egy L-et eldöntő $p(n)$ polinom időkorlátos NTG. (Feltehető, hogy $p(n) \geq n$.)
  - Legyen továbbá $w = a_1 \cdots a_n \in \Sigma^*$ egy szó.
  - M segítségével megadunk egy polinom időben előállítható φ_w nulladrendű KNF formulát, melyre $w \in L \Leftrightarrow \langle\varphi_w\rangle \in$ SAT.
  - M egy számítása w-n leírható egy T táblázattal, melynek
  – első sora $\# \sqcup^{p(n)} C_0 \sqcup^{p(n)-n} \#$, ahol $C_0 = q_0 w$ M kezdőkonfigurációja w-n
  – T egymást követő két sora M egymást követő két konfigurációja (elegendő $\sqcup$-el kiegészítve, elején és a végén egy $\#$-el). Minden sor $2p(n)+3$ hosszú.

A Cook-Levin tétel bizonyítása

– $p(n)+1$ sor van. Ha hamarabb jut elfogadó konfigurációba, akkor onnantól kezve ismételjük meg az elfogadó konfigurációt.



– a konfigurációátmenet definíciója miatt bármely két sor közötti különbség belefér egy 2x3-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:

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

– 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.

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

A Cook-Levin tétel bizonyítása

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

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

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:

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

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 $\Leftrightarrow$ φ_w kielégíthető $\Leftrightarrow \langle\varphi_w\rangle \in$ 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} : $\le p(n)(2p(n)+1)k^6 \cdot 6 = O(p^2(n))$,
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 $L \le_p$ SAT.
• Ez tetszőleges $L \in$ NP nyelvre elmondható. Így SAT NP-nehéz. Mivel NP-beli, ezért NP-teljes is.

Polinom idejű visszavezetés tranzitivitása

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

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.

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

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

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

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 $L' \in$ NP, akkor L' NP-teljes.

Bizonyítás:

Legyen $L'' \in$ 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' NP-nehéz. Ebből és a 3. feltételből következik az állítás.

kSAT

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

KNF: konjuktív normálformájú nulladrendű formula

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

Definíció

k**KNF**-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:

kSAT=$\{\langle\varphi\rangle \mid \varphi$ kielégíthető kKNF$\}$

3SAT NP-teljessége

Tétel

3SAT NP-teljes.

Bizonyítás:

- 3SAT NP-beli. Lásd az érvelést SAT-nál.
- SAT $\leq_p$ 3SAT
  Kell $f : \varphi \mapsto \varphi'$, φ KNF, φ' 3KNF, φ' kielégíthető $\Leftrightarrow \varphi$ kielégíthető, f polinom időben kiszámolható.

$\varphi \mapsto \varphi'$:

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

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

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

3SAT NP-teljessége

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

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-teljessége

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

$n = 1$: $(\Rightarrow)$: 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 $x_1, \ldots, x_{i-2}$ igaz $x_{i-1}, \ldots, x_{n-3}$ hamis. Átgondolható, hogy minden tagban lesz igaz literál.

3SAT NP-teljessége

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

$n \geq 5$: $(\Leftarrow)$: Tegyük fel, hogy jobboldalon minden tag igaz és indirekt tegyük fel, hogy $\ell_1, \ldots, \ell_n$ hamis. Ekkor az új tagokon balról jobbra végighaladva sorra kapjuk, hogy $x_1, \ldots x_{n-3}$ igaz kell legyen, de ekkor az uolsó tag mégiscsak hamis, ellentmondás.

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

2SAT P-beli

Tétel

2SAT $\in$ P.

Bizonyítás Legyen φ egy $x_1, \ldots, x_n$ változókat 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 komplemens 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 komplemens 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 komplemens 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 komplemens 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 komplemens párjába irányított út.

2SAT P-beli

Ez az I interpretáció minden klózt igazra értékel.

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.

HORNSAT P-beli

Definíció

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

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

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

Tétel

HORNSAT $\in$ P.

Nem bizonyítjuk.

Bevezetés a számításelméletbe

11. előadás

3 színezhetőség

Definíció

Legyen $k \geqslant 1$ 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 $\exists\, f : V \to \{1, \ldots, k\}$ 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 $k \geqslant n$-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

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

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

Tétel

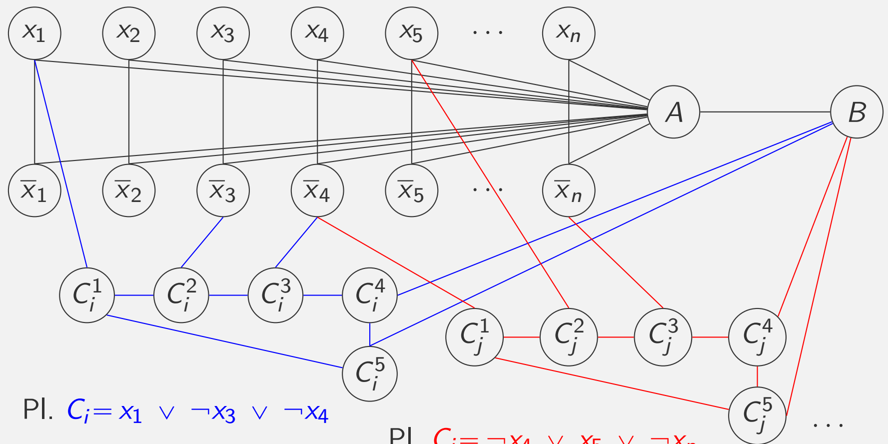
3Színezés NP-teljes.

- kSzínezés NP-beli: egy $\langle G\rangle$ inputra az M NTG egy számítási ága állítson elő egy $f : V(G) \to \{1, \ldots, k\}$ 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.
- 3SAT $\leqslant_p$ 3Színezés: elegendő minden φ 3KNF formulához polinom időben elkészíteni egy G_φ gráfot úgy, hogy φ kielégíthető $\Leftrightarrow$ G_φ 3-színezhető.

3 színezhetőség

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



Pl. $C_i = x_1 \vee \neg x_3 \vee \neg x_4$

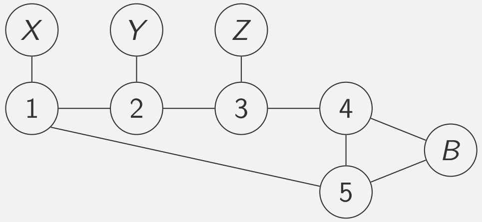
Pl. $C_j = \neg x_4 \vee x_5 \vee \neg x_n$

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.

G_0



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.

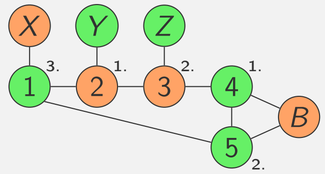
  Ez persze még nem jó, lehetnek azonos színű szomszédok.

  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:

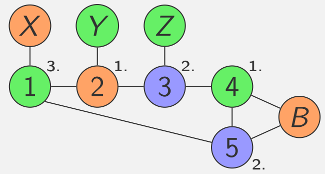
1. lépés utáni színezés

G_0



2. lépés utáni színezés

G_0



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 $\overline{x}_i$ 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 $x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n$ mind A szomszédai, így egyikük se lehet kék. Továbbá az $(x_i, \overline{x}_i)$ párok össze vannak kötve, így minden párban pontosan egy piros és pont 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:=igaz $\Leftrightarrow$ x_i csúcs zöld" interpretáció tehát kielégíti φ-t.

2 színezhetőség

Beláttuk tehát, hogy $\varphi \mapsto G_\varphi$ visszavezetés. Mivel G_φ φ ismeretében az input méretének polinomja időben legyártható, ezért a visszavezetés polinom idejű.

Mivel 3SAT NP-teljes, korábbi tételünk miatt 3Színezés is az. □

A 2-színezhető gráfok éppen a páros gráfok (a páros gráf két csúcsosztálya 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 feltétel 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 elsősége 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 ellenőrzheti. $\square$

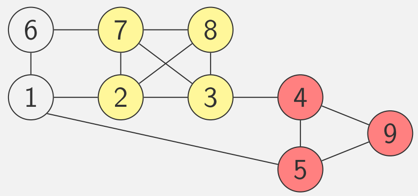
Klikk

Definíció

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

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

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.

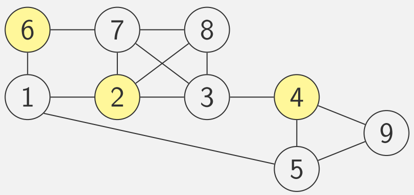
Független ponthalmaz

Definíció

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

$\textsc{Független ponthalmaz} :=$
$\{\langle G, k\rangle \mid G\text{-nek van } k \text{ méretű független ponthalmaza}\}$

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

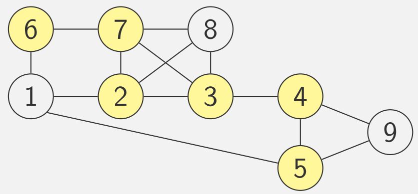
Definíció

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 csúcsfedés néven is ismeretes.

$\textsc{Lefogó ponthalmaz} :=$
$\{\langle G, k\rangle \mid G\text{-nek van } k \text{ méretű lefogó ponthalmaza}\}$

Példa:



$\{2,3,4,5,6,7\}$ lefogó ponthalmaz.

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

Független ponthalmaz

Tétel

Klikk, Független ponthalmaz, Lefogó ponthalmaz NP-teljes.

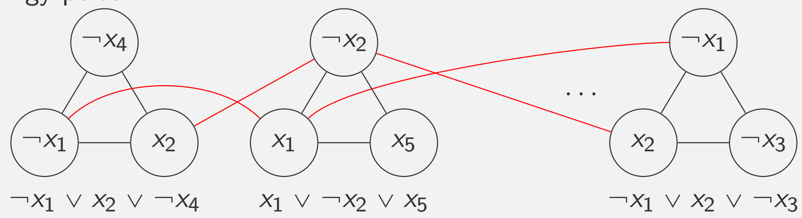
- Egy NTG egy számítási ágán vizsgálja meg a csúcsoknak egy konkrét, k elemű részhalmazát. Egy k elemű ponthalmaz előállítása illetve annak ellenőrzése, hogy ez egy klikk/független ponthalmaz/lefogó ponthalmaz az input méretének polinomiális függvénye. Tehát mindhárom nyelv NP-ben van.
- 3SAT $\leqslant_p$ Független csúcshalmaz

  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 többitől diszjunkt háromszöget, a csúcsokhoz rendeljük hozzá a literálokat. Így m darab klóz esetén $3m$ csúcsot kapunk. Kössük össze éllel ezen felül a komplemens párokat is. $k := m$.

Független ponthalmaz

Egy példa:



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

∗ Ha G_φ-ben van m független csúcs, akkor ez csak úgy lehet, ha háromszögenként 1 van. Vegyünk egy ilyet, ezen csúcsoknak megfelelő literálok között nem lehet komplemens 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 ponthalmaz

- Független ponthalmaz $\leqslant_p$ Klikk

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

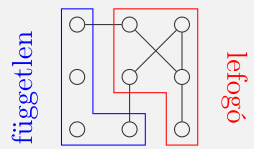
  Ez egy jó visszavezetés, hiszen ami G-ben klikk az $\bar{G}$-ben független ponthalmaz és fordítva, ami G-ben független ponthalmaz az $\bar{G}$-ben klikk.
- Független ponthalmaz $\leqslant_p$ Lefogó ponthalmaz

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

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

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

  Mindkét visszavezetés polinom időben kiszámítható.



Lefogó ponthalmaz hipergráfokban

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

Hipergráf lefogó ponthalmaz:=
$\{\langle \mathcal{S}, k\rangle \,|\, \mathcal{S}$ egy hipergráf és van k elemű $\mathcal{S}$-et lefogó ponthalmaz$\}$.

Tétel

Hipergráf lefogó ponthalmaz NP-teljes.

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

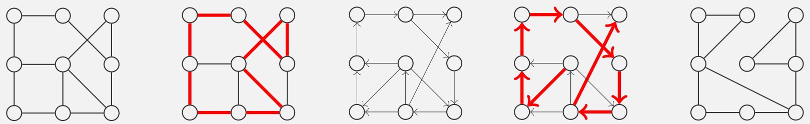
Lefogó ponthalmaz a Hipergráf lefogó ponthalmaz speciális esete. Minden gráf hipergráf is egyben, a megfeleltetés $U = V(G)$, $\mathcal{S} = E(G)$. k ugyanaz, mivel a lefogó ponthalmaz szintén speciális esete a hipergráf lefogó ponthalmaznak.

Irányítatlan/irányított 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.



G H-kör G-ben G' H-út G'-ben   nincs H-kör

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

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

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

Irányított $s \rightsquigarrow t$ Hamilton út NP teljessége

Tétel

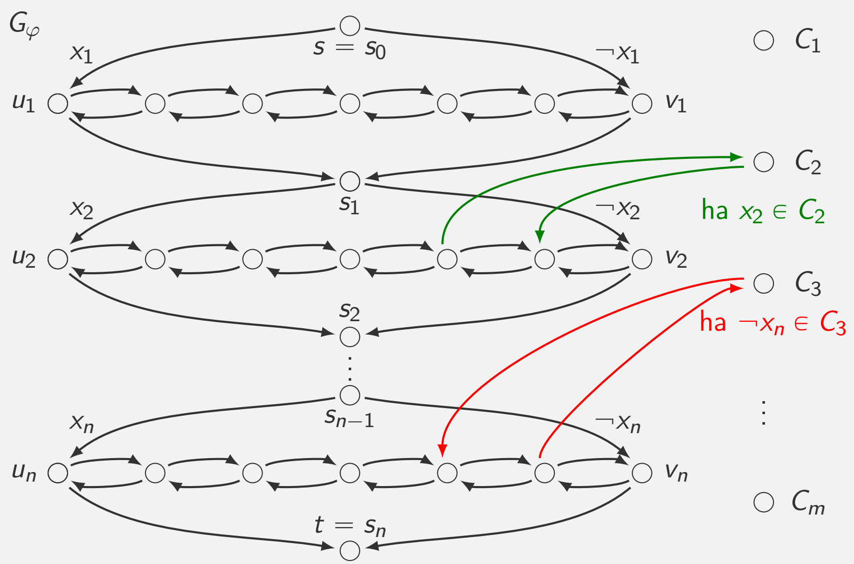
HÚ NP-teljes

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 $\leqslant_p$ HÚ. 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 $x_1, \ldots x_n$ a φ-ben előforduló ítéletváltozók és $C_1, \ldots C_m$ φ klózai.

Irányított $s \rightsquigarrow t$ Hamilton út NP teljessége



Irányított $s \rightsquigarrow t$ Hamilton út NP teljessége

G_φ konstrukciója

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

Az $u_i v_i$ út pozitív bejárása: $u_i \rightsquigarrow v_i$.
Az $u_i v_i$ út negatív bejárása: $u_i \leftsquigarrow v_i$.

Irányított $s \rightsquigarrow t$ Hamilton út NP teljessége

- Egy $s \rightsquigarrow t$ H-út $\forall 1 \leqslant i \leqslant n$-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 ($\forall 1 \leqslant j \leqslant m$). 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 SAT $\leqslant_p$ HÚ, azaz HÚ NP-nehéz, de láttuk, hogy NP-beli, így NP-teljes is.

Irányítatlan $s \leftrightsquigarrow t$ Hamilton út NP teljessége

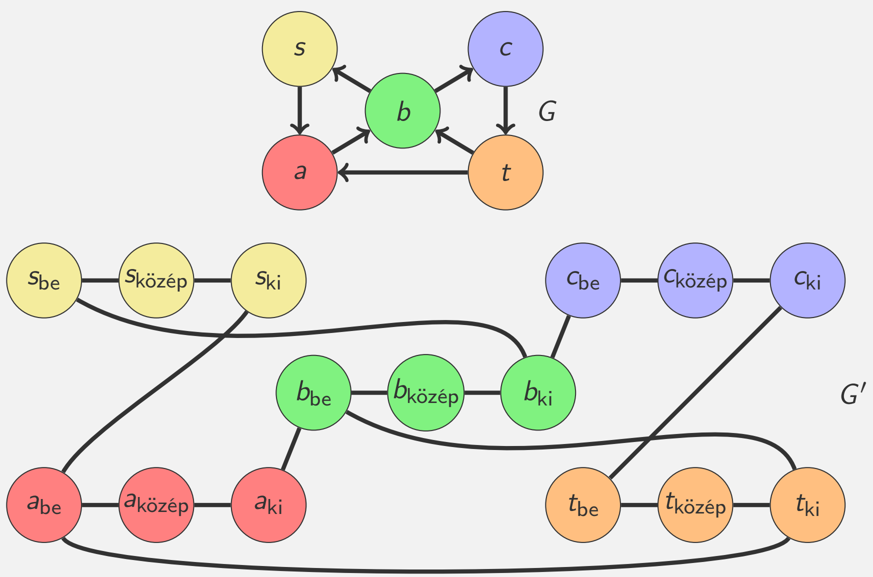
Megjegyzés: IHÚ és IHK NP-belisége az előzőekhez hasonlóan adódik.

Tétel

IHÚ NP-teljes

Bizonyítás: HÚ $\leqslant_p$ 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}, $v_{\text{közép}}$ és v_{ki}. és G' élei közé vegyük be a $\{v_{\text{be}}, v_{\text{közép}}\}$ és $\{v_{\text{közép}}, v_{\text{ki}}\}$ éleket. Továbbá minden $E = (u, v)$ G-beli él estén adjuk hozzá $E(G')$-höz $\{u_{\text{ki}}, v_{\text{be}}\}$-t. $s' := s_{\text{be}}, t' := t_{\text{ki}}$.

Irányítatlan $s \leftrightsquigarrow t$ Hamilton út NP teljessége



Irányítatlan $s \leftrightsquigarrow t$-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}, $v_{\text{közép}}$, v_{ki} csúcsokkal helyettesítsük.
- ha G'-ben van egy $P' : s' \leftrightsquigarrow t'$ H-út, akkor P'-ben minden v-re v_{be}, $v_{\text{közép}}$, v_{ki} egymást követő csúcsok, hiszen $v_{\text{közép}}$ 2-fokú csúcs és máskülönben nem lehetne rajta P'-n. Ezen csúcshármasokat $\{u_{\text{ki}}, v_{\text{be}}\}$ típusú élek kötik össze, melyekhez definíció szerint van $u \to v$ él G-ben.
Tehát ha minden v-re a P'-ben egymást követő v_{be}, $v_{\text{közép}}$, v_{ki} csúcshármast v-vel helyettesítjük egy G-beli irányított H-útat kapunk.

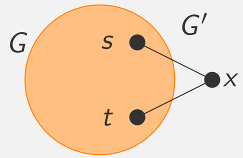
Irányítatlan Hamilton kör NP teljessége

Tétel

IHK NP-teljes

Bizonyítás: IHÚ $\leqslant_p$ 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=$\{\langle G, K\rangle \mid G$-ben van $\leqslant K$ súlyú H-kör$\}$.

Tétel

TSP NP-teljes

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

IHK $\leqslant_p$ 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.

Bevezetés a számításelméletbe

12. előadás

NP lehetséges szerkezete

Definíció

L **NP-köztes**, ha $L \in \mathrm{NP}$, $L \notin \mathrm{P}$ és L nem NP-teljes.

Ladner tétele

Ha $\mathrm{P} \neq \mathrm{NP}$, akkor létezik NP-köztes nyelv.

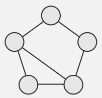
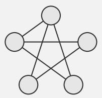
(biz. nélkül)

Mivel nem tudjuk, hogy $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$, ezért nem tudjuk, hogy léteznek-e NP-köztes nyelvek. Valószínűleg igen, hiszen azt gondoljuk, hogy $\mathrm{P} \neq \mathrm{NP}$.

Vannak azonban olyan nyelvek, amelyeknek se a P-beliségét, se az NP-teljességét nem sikerült eddig igazolni, így erős NP-köztes jelölteknek számítanak.

NP-köztes jelöltek

- GRÁFIZOMORFIZMUS $= \{\langle G_1, G_2\rangle \mid G_1$ és G_2 irányítatlan izomorf gráfok$\}$.

  Példa:  és  izomorfak.

  Egy új eredmény: Babai László, magyar matematikus 2017-es eredménye: GRÁFIZOMORFIZMUS $\in$ QP, ahol

  $$\mathrm{QP} = \bigcup_{c \in \mathbb{N}} \mathrm{TIME}(2^{(\log n)^c})$$

  a „kvázipolinom időben” megoldható problémák osztálya.

- Prímfaktorizáció: adjuk meg egy egész szám prímtényezős felbontását! [számítási feladat]

  A probléma eldöntési változata:

  PRÍMFAKTORIZÁCIÓ $=$ $\{\langle n, k\rangle \mid n$-nek van k-nál kisebb prímtényezője$\}$

co$\mathcal{C}$ bonyolutsági osztályok

Definíció

Ha $\mathcal{C}$ egy bonyolultsági osztály $\mathrm{co}\mathcal{C} := \{L \mid \overline{L} \in \mathcal{C}\}$.

Definíció

$\mathcal{C}$ **zárt a polinomidejű visszavezetésre nézve**, ha minden esetben ha $L_2 \in \mathcal{C}$ és $L_1 \leqslant_p L_2$ 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 co$\mathcal{C}$ is.

Bizonyítás: Legyen $L_2 \in \mathrm{co}\mathcal{C}$ és L_1 tetszőleges nyelvek, melyekre $L_1 \leqslant_p L_2$. Utóbbiból következik, hogy $\overline{L_1} \leqslant_p \overline{L_2}$ (ugyananaz a visszavezetés jó!). Mivel $\overline{L_2} \in \mathcal{C}$, ezért a tétel feltétele miatt $\overline{L_1} \in \mathcal{C}$. Azaz $L_1 \in \mathrm{co}\mathcal{C}$.

co$\mathcal{C}$ bonyolutsági osztályok

Következmény

coNP zárt a polinom idejű visszavezetésre nézve.

Igaz-e, hogy P = coP? Igen. (L-et polinom időben eldöntő TG q_i és q_n állapotát megcseréljük: $\overline{L}$-t polinom időben eldöntő TG.)
Igaz-e, hogy NP = coNP? A fenti konstrukció NTG-re nem feltétlen $\overline{L}$-t dönti el. Valójában azt sejtjük, hogy NP$\neq$coNP.

Tétel

L $\mathcal{C}$-teljes $\iff$ $\overline{L}$ co$\mathcal{C}$-teljes.

Bizonyítás:

- Ha $L \in \mathcal{C}$, akkor $\overline{L} \in$ co$\mathcal{C}$.
- Legyen $L' \in \mathcal{C}$, melyre $L' \leqslant_p L$. Ekkor $\overline{L'} \leqslant_p \overline{L}$.
  Ha L' befutja $\mathcal{C}$-t akkor $\overline{L'}$ befutja co$\mathcal{C}$-t. Azaz minden co$\mathcal{C}$-beli nyelv polinom időben visszavezethető $\overline{L}$-re.

Tehát $\overline{L}$ co$\mathcal{C}$-beli és co$\mathcal{C}$-nehéz, így co$\mathcal{C}$-teljes.

Példák coNP teljes nyelvekre

UNSAT := $\{\langle\varphi\rangle \mid \varphi$ kielégíthetetlen nulladrendű formula$\}$.

TAUT := $\{\langle\varphi\rangle \mid$ a φ nulladrendű formula tautológia$\}$.

Tétel

UNSAT és TAUT coNP-teljesek.

Bizonyítás: ÁLTSAT = $\{\langle\varphi\rangle \mid \varphi$ kielégíthető nulladrendű formula$\}$ is NP-teljes (NP-beli és SAT speciális esete neki.)
$\overline{\text{ÁLTSAT}}$:= UNSAT, az előző tétel alapján UNSAT coNP-teljes.
UNSAT $\leqslant_p$ TAUT, hiszen $\varphi \mapsto \neg\varphi$ polinom idejű visszavezetés.

Informálisan: coNP tartalmazza a polinom időben **cáfolható** problémákat.

Megjegyzések: Sejtés, hogy NP $\neq$ coNP. Egy érdekes osztály ekkor az NP $\cap$ coNP. Nyilván P $\subseteq$ NP $\cap$ coNP. Sejtés: P $\neq$ NP $\cap$ coNP. Bizonyított, hogy ha egy coNP-teljes problémáról kiderülne, hogy NP-beli, akkor NP = coNP.

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 problémáknál beszámítjuk.

Számítási problémáknál 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
$\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma^{k+1} \to Q \times \Gamma^k \times \{L, S, R\}^{k+1}$.

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 állaptotátmenetfüggvénye tehát $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma^{k+1} \to Q \times \Gamma^{k+1} \times \{L, S, R\}^{k+2}$.

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

- SPACE $(f(n)) := \{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos determinisztikus offline TG-pel$\}$
- NSPACE $(f(n)) := \{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos nemdeterminisztikus offline TG-pel$\}$
- PSPACE:=$\bigcup_{k\geqslant 1}$SPACE (n^k).
- NPSPACE:=$\bigcup_{k\geqslant 1}$NSPACE (n^k).
- L:=SPACE $(\log n)$.
- NL:=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

Elér $= \{\langle G, s, t\rangle \mid$ A G irányított gráfban van s-ből t-be út$\}$.
Algo 2-ből, tudjuk, hogy Elér P-ben van (szélességi bejárás).

Tétel

Elér $\in$ TIME(n^2).

Tétel

Elér $\in$ SPACE$(\log^2 n)$.

Bizonyítás:

- Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ÚT(x, y, i) :=igaz, ha $\exists$ x-ből y-ba legfeljebb 2^i hosszú út.
- s-ből van t-be út G-ben $\iff$ ÚT$(s, t, \lceil \log_2 n \rceil)$=igaz.
- ÚT(x, y, i)=igaz $\iff$ $\exists z($ ÚT$(x, z, i-1)$=igaz $\wedge$ ÚT$(z, y, i-1)$=igaz $)$.
- Ez alapján egy rekurzív algoritmust készítünk, 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 ÚT(x, y, i) függvény meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i-1)$ hármast a munkaszalagra az (x, y, i) utáni helyre majd kiszámítja 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 ÚT$(z, y, i-1)$ igaz, akkor ÚT(x, y, i) igaz (ezt (x, y, i) és $(z, y, i-1)$ 2. argumentumának egyezéséből látja)
  - Ha ÚT$(z, y, i-1)$ hamis akkor kitörli a $(z, y, i-1)$-t és z értékét eggyel növelve ÚT$(x, z, i-1)$-en dolgozik tovább.
- Ha egyik z se volt jó, akkor ÚT(x, y, i) hamis.

ELÉR determinisztikus tárbonyolultsága

A főprogram, tehát $(s, t, \lceil \log_2 n \rceil)$ feírásából és az 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 logaritmusa $+1$ darab számjegy szükséges.

Így a rendezett hármasokból mindvégig $O(\log n)$ van és egyenként $O(\log n)$ hosszúak, így ELÉR $\in$ 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 ELÉR $\in$ TIME(n^2) vagy ELÉR $\in$ 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 $f(n) \geqslant \log n$, akkor NSPACE$(f(n)) \subseteq$ 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 $\geqslant \log n$ tár kell ennek eltárolásához). Ha $f(n) \geqslant \log n$, 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 $O(\log^2(2^{df(n)})) = O(f^2(n))$ 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

PSPACE = NPSPACE

Bizonyítás: $L \in \mathrm{NSPACE}(n^k) \overset{\text{Savitch}}{\Longrightarrow} L \in \mathrm{SPACE}(n^{2k})$.

Tétel

NL $\subseteq$ P

Bizonyítás

Legyen $L \in$ 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 $L \in$ P.

Elér eldöntése nemdeterminisztikus log. tárral

Elér fontos szerepet tölt be az L$\overset{?}{=}$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.

Logaritmikus táras visszavezetés, NL-teljesség

Definíció

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre, ha $L_1 \leqslant L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmikus többlet tárkorlátos determinisztikus offline Turing géppel. Jelölése: $L_1 \leqslant_\ell L_2$.

Definíció

Egy L nyelv **NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in$ NL nyelvre, $L' \leqslant_\ell L$. Ha ezen felül $L \in$ NL is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Tétel

Az L osztály zárt a logaritmikus tárral való visszavezetésre nézve.

Bizonyítás: Tegyük fel, hogy $L_1 \leqslant_\ell L_2$ és $L_2 \in$ L. Legyen M_2 az L_2-t eldöntő, M pedig a visszavezetésben használt f függvényt kiszámoló logaritmikus táras determinisztikus OTG.

L logaritmikus 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 logaritmikus méretű tárat használ, azaz $L_1 \in$ L.

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 $L' \leqslant_\ell L$. $L \in$ L, így L logaritmikus tárral való visszavezetésre való zártsága miatt $L' \in$ L. Tehát NL $\subseteq$ L. A másik irány a definíciókból következik.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

Bizonyítás:

- Korábban láttuk, hogy ELÉR $\in$ NL
- Legyen $L \in$ NL, megmutatjuk, hogy $L \leqslant_\ell$ ELÉR
- Legyen M egy L-et eldöntő $O(\log n)$ táras NOTG és $|u| = n$
- Az $O(\log n)$ tárat használó konfigurációk $\leqslant c \cdot \log n$ 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 $L \leqslant$ 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 = coNL

(biz. nélkül)

Hierarchia tétel

EXPTIME:=$\bigcup_{k \in \mathbb{N}}$TIME$(k^n)$.

Hierarchia tétel

(I) NL $\subset$ PSPACE és P $\subset$ EXPTIME.

(II) L $\subseteq$ NL=coNL $\subseteq$ P $\subseteq$ NP $\subseteq$ NPSPACE=PSPACE $\subseteq$ EXPTIME

Sejtés: A fenti tartalmazási lánc minden tartalmazása valódi.

Hierarchia tétel

(I)-et nem bizonyítjuk.

(II) bizonyítása:

$$\mathrm{L} \overset{(1)}{\subseteq} \mathrm{NL} \overset{(2)}{=} \mathrm{coNL} \overset{(3)}{\subseteq} \mathrm{P} \overset{(4)}{\subseteq} \mathrm{NP} \overset{(5)}{\subseteq} \mathrm{NPSPACE} \overset{(6)}{=} \mathrm{PSPACE} \overset{(7)}{\subseteq} \mathrm{EXPTIME}$$

(1) és (4): a nemdeterminisztikusság definíciójából következik

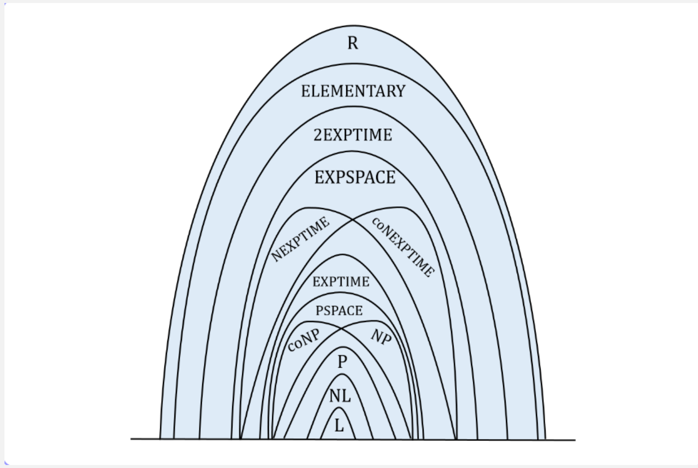
(2): Immerman- Szelepcsényi

(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]