Számításelmélet

1. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Alapfogalmak, jelölések – Ismétlés

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

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

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

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

Alapfogalmak, jelölések – Ismétlés

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

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

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

Nyelvcsalád (vagy nyelvosztály) alatt nyelveknek egy halmazát értjük.

Tehát ha V egy ábécé:

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

Grammatikák – Ismétlés

Definíció

Legyen $G = \langle N, T, P, S \rangle$ egy grammatika és legyen $u, v \in (N \cup T)^*$. A v szó közvetlenül vagy **egy lépésben levezethető** az u szóból G -ben, jelölése $u \Rightarrow_G v$, ha $u = u_1 x u_2$ és $v = u_1 y u_2$,ahol $u_1, u_2 \in (N \cup T)^*$ és $x \to y \in P$.

u-ból (több lépésben vagy közvetetten) **levezethető** v, ha $u = v$ vagy van olyan $n \geqslant 1$ és $w_0, \ldots w_n \in (N \cup T)^*$, hogy $w_{i-1} \Rightarrow_G w_i$ $(1 \leqslant i \leqslant n)$ és $w_0 = u$ és $w_n = v$. Jelölés: $u \Rightarrow_G^* v$.

Mondatforma: A kezdőszimbólumból levezethető szó.

$\Rightarrow_G$ illetve $\Rightarrow_G^*$ helyett gyakran röviden $\Rightarrow$-t illetve $\Rightarrow^*$-t írunk.

Definíció

Legyen $G = \langle N, T, P, S \rangle$ egy tetszőleges grammatika. A G által **generált nyelv** alatt az $L(G) := \{w \mid S \Rightarrow_G^* w, w \in T^*\}$ szavakból álló halmazt értjük.

Grammatikák osztályozása – Ismétlés

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

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

Nyelvek Chomsky féle osztályozása – Ismétlés

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

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

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

Chomsky nyelvhierarchia tétele

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

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

Formális nyelvek megadása

- (Generatív) grammatikákkal

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

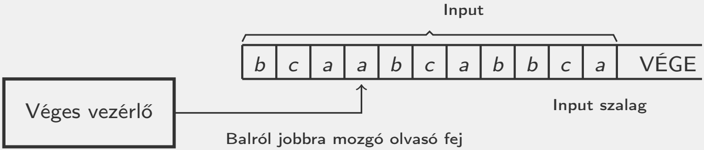
- Matematikai gépek, automaták segítségével

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

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

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

Véges automata, intuitív kép – Ismétlés



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

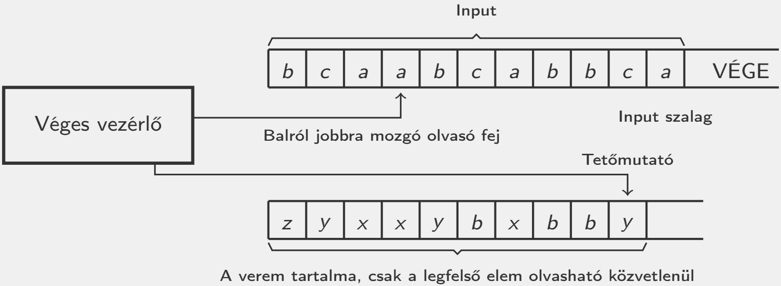
3-as típusú nyelvek – Ismétlés

Tétel

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

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

Veremautomata – Ismétlés



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

Veremautomata – Ismétlés

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

Definíció

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

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

Veremautomata – Ismétlés

- A verem tetején lévő szimbólum, az aktuális állapot és az inputszimbólum együttesen határozzák meg a következő átmenetet.

- Minden lépésben mindenképpen kiveszünk egyetlen egy elemet a verem tetejéről és beteszünk helyette néhányat. ($0, 1, 2, \ldots$ darabot)

- Ha $\delta(z, q, \varepsilon)$ nem üres, akkor ún. **ε-átmenet** (ε-lépés, ε-mozgás) hajtható végre, ami lehetővé teszi, hogy a veremautomata anélkül változtassa meg az állapotát, hogy valamilyen szimbólumot olvasson az inputszalagról.

- ε-mozgásra lehetőség van már az első inputszimbólum elolvasása előtt is illetve még az utolsó inputszimbólum elolvasása után is.

Veremautomata – Ismétlés

Veremautomata konfigurációi

Definíció

A veremautomata **konfigurációja** alatt egy zqw alakú szót értünk, ahol $z \in Z^*$ a verem aktuális tartalma és $q \in Q$ az aktuális állapot és $w \in T^*$ az input még feldolgozatlan része.

z első betűje van a verem alján, míg utolsó betűje a verem tetején.

Az input olvasófeje w első betűjén áll.

Így a q baloldalán lévő szimbólum van a verem tetején, míg a jobboldalán lévő szimbólum az input következő feldolgozandó betűje.

Definíció

Az $A = \langle Z, Q, T, \delta, z_0, q_0, F \rangle$ veremautomata $w \in T^*$ bemenethez tartozó **kezdőkonfigurációja** $z_0 q_0 w$.

Veremautomata – Ismétlés

Alapvető veremműveletek megvalósítása

Legyen $t \in T \cup \{\varepsilon\}$, $q, r \in Q$ és $z \in Z$

- $(\varepsilon, r) \in \delta(z, q, t)$: a z elemet kivehetjük a veremből (POP művelet)
- $(z, r) \in \delta(z, q, t)$: a verem tartalma változatlan maradhat
- $(z', r) \in \delta(z, q, t)$: z-t lecserélhetjük z'-re a verem tetején ($z' \in Z$)
- $(zz', r) \in \delta(z, q, t)$: z'-t a verem tetejére (z-re rá) tehetjük ($z' \in Z$) (PUSH művelet)
- Egyéb lehetőségek, például $(zz'z'', r) \in \delta(z, q, t)$: $z'z''$-t a verem tetejére tehetjük, z'' lesz a tetején ($z', z'' \in Z$) .
- Általánosan $(w, r) \in \delta(z, q, t)$, ahol $w \in Z^*$ tetszőleges Z feletti szó. A w szó kerül z helyére és w utolsó betűje lesz a verem tetején.

Veremautomata – Ismétlés

Egylépéses redukció

Definíció

Az A veremautomata az $\alpha \in Z^*QT^*$ konfigurációt a $\beta \in Z^*QT^*$ konfigurációra **redukálja egy lépésben**, amelyet $\alpha \Rightarrow_A \beta$-val jelölünk, ha létezik olyan $z \in Z, q, p \in Q, a \in T \cup \{\varepsilon\}$, $r, u \in Z^*$ és $w \in T^*$, hogy $(u, p) \in \delta(z, q, a)$ és $\alpha = rzqaw$ és $\beta = rupw$ teljesül.

Példák:

- ha A-ban $\delta(c, q_1, a) = \{(dd, q_2), (\varepsilon, q_4)\}$ és $z_0cddcq_1ababba$ egy konfiguráció, akkor $z_0cddcq_1ababba \Rightarrow_A z_0cddddq_2babba$ és $z_0cddcq_1ababba \Rightarrow_A z_0cddq_4babba$ is teljesül,
- ha A-ban $\delta(c, q_3, \varepsilon) = \{(dd, q_2)\}$ és $z_0cddcq_3ababba$ egy konfiguráció, akkor $z_0cddcq_3ababba \Rightarrow_A z_0cddddq_2ababba$
- ha A-ban $\delta(c, q_5, \varepsilon) = \emptyset$ és $\delta(c, q_5, a) = \emptyset$, akkor nem létezik olyan C konfiguráció, melyre $z_0ccq_5aab \Rightarrow_A C$

Veremautomata – Ismétlés

Többlépéses redukció és a felismert nyelv

Definíció

Az A veremautomata az $\alpha \in Z^*QT^*$ konfigurációt a $\beta \in Z^*QT^*$ konfigurációra **redukálja**, amelyet $\alpha \Rightarrow_A^* \beta$-val jelölünk, ha vagy $\alpha = \beta$, vagy létezik olyan $\alpha_1, \ldots, \alpha_n$ szavakból álló véges sorozat, ahol $\alpha = \alpha_1, \beta = \alpha_n$ és $\alpha_i \Rightarrow_A \alpha_{i+1}$, $1 \leqslant i \leqslant n-1$.

Tehát a $\Rightarrow_A^*$ reláció a $\Rightarrow_A$ reláció reflexív, tranzitív lezártja.

Példa: Ha $\delta(d, q_6, b) = \{(\varepsilon, q_5)\}$ és $\delta(d, q_5, \varepsilon) = \{(dd, q_2), (\varepsilon, q_4)\}$ akkor $\#cddq_6bab \Rightarrow_A \#cdq_5ab \Rightarrow_A \#cddq_2ab$ és
$\#cddq_6bab \Rightarrow_A \#cdq_5ab \Rightarrow_A \#cq_4ab$.
Tehát $\#cddq_6bab \Rightarrow_A^* \#cddq_2ab$ és $\#cddq_6bab \Rightarrow_A^* \#cq_4ab$.

Definíció

Az A veremautomata által **elfogadó állapottal (végállapottal) elfogadott nyelv**
$L(A) = \{w \in T^* \mid z_0q_0w \Rightarrow_A^* up$, ahol $u \in Z^*, p \in F\}$.

Veremautomata – Ismétlés

Determinisztikus veremautomata

Definíció

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

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

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

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

Veremautomata – Ismétlés

Alternatív reprezentációk

- **Átírási szabályokkal:**
  A δ leképezést szabályok formájában is megadhatjuk. Az így nyert szabályhalmazt M_δ -val jelöljük. Tehát ezzel az alternatív jelöléssel:

  $$zqa \to up :\in M_\delta \iff (u,p) \in \delta(z,q,a),$$
  $$zq \to up :\in M_\delta \iff (u,p) \in \delta(z,q,\varepsilon).$$
  $$(p, q \in Q, a \in T, z \in Z, u \in Z^*)$$

- **Átmenetdiagrammal:**

  $p, q \in Q, a \in T \cup \{\varepsilon\}, z \in Z, u \in Z^*$ esetén:

  $$\textcircled{q} \xrightarrow{a;\, z \,\to\, u} \textcircled{p} \qquad \iff \quad (u,p) \in \delta(z,q,a)$$

  A végállapotokat duplán karikázzuk. A kezdőállapotot $\to$ jelöli.

Veremautomata – Ismétlés

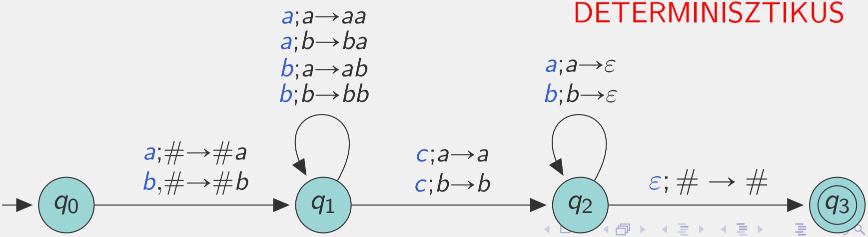
1. Példa: Legyen $L_1 = \{wcw^{-1} \mid w \in \{a,b\}^+\}$. Készítsünk egy A veremautomatát, melyre $L(A) = L_1$.

Megoldás:

$A = \langle \{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \{\#, a, b\}, \delta, q_0, \#, \{q_3\} \rangle$, ahol:

$$
\begin{aligned}
(\#t, q_1) &\in \delta(\#, q_0, t) && \forall t \in \{a,b\} \\
(zt, q_1) &\in \delta(z, q_1, t) && \forall z, t \in \{a,b\} \\
(z, q_2) &\in \delta(z, q_1, c) && \forall z \in \{a,b\} \\
(\varepsilon, q_2) &\in \delta(t, q_2, t) && \forall t \in \{a,b\} \\
(\#, q_3) &\in \delta(\#, q_2, \varepsilon)
\end{aligned}
$$

DETERMINISZTIKUS



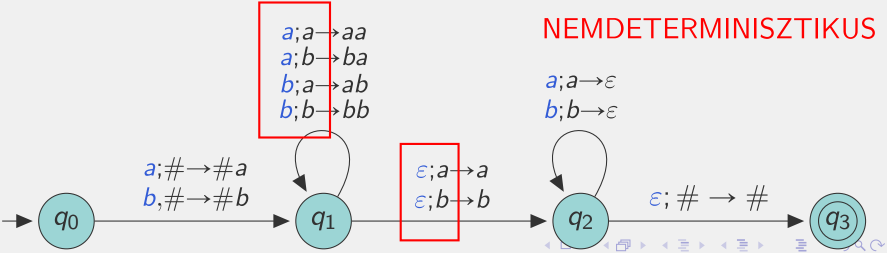
Veremautomata – Ismétlés

2. Példa: Legyen $L_2 = \{ww^{-1} \mid w \in \{a,b\}^+\}$. Készítsünk egy A veremautomatát, melyre $L(A) = L_2$.

Megoldás:

$A = \langle \{q_0, q_1, q_2, q_3\}, \{a, b\}, \{\#, a, b\}, \delta, q_0, \#, \{q_3\}\rangle$, ahol:

$$\begin{aligned}
(\#t, q_1) &\in \delta(\#, q_0, t) && \forall t \in \{a, b\}\\
(zt, q_1) &\in \delta(z, q_1, t) && \forall z, t \in \{a, b\}\\
(z, q_2) &\in \delta(z, q_1, \varepsilon) && \forall z \in \{a, b\}\\
(\varepsilon, q_2) &\in \delta(t, q_2, t) && \forall t \in \{a, b\}\\
(\#, q_3) &\in \delta(\#, q_2, \varepsilon) &&
\end{aligned}$$



Veremautomata – Ismétlés

Üres veremmel elfogadott nyelv

Definíció

Az A veremautomata által **üres veremmel elfogadott nyelv** $N(A) = \{w \in T^* \mid z_0 q_0 w \Rightarrow_A^* p, \text{ ahol } p \in Q\}$.

Megjegyzés:
Vegyük észre, hogy ha a verem üres, akkor az automata működése blokkolódik, mivel nincs átmenet definiálva üres verem esetére. (lásd δ definíciója).

Így a verem az input teljes feldolgozása után, az utolsó átmenettel kell üressé váljon.

Szintén a blokkolás elkerülése végett definiáltuk úgy a kezdőkonfigurációt, hogy a veremábécé egy eleme (z_0) már eleve a veremben van.

Megjegyzés: Vegyük észre, hogy az elfogadó állapotok halmaza irreleváns $N(A)$ szempontjából.

Veremautomata – Ismétlés

Üres veremmel elfogadott nyelv

Példa: Az alábbi $A = \langle \{\$, a\}\{q_0, q_1\}, \{a, b\}, \delta, \$, q_0, \{\ \}\rangle$ veremautomata esetén $N(A) = \{a^n b^n \mid n \geqslant 1\}$, azaz ezt nyelvet ismeri fel üres veremmel.

M_δ :

$$
\begin{aligned}
\$q_0 a &\to \$aq_0 \\
aq_0 a &\to aaq_0 \\
aq_0 b &\to q_1 \\
aq_1 b &\to q_1 \\
\$q_1 \quad &\to q_1.
\end{aligned}
$$

A determinisztikus, a^2b^3-re:

$\$q_0aabbb \Rightarrow \$aq_0abbb \Rightarrow \$aaq_0bbb \Rightarrow \$aq_1bb \Rightarrow \$q_1b \Rightarrow q_1b.$

A elutasítja $aabbb$-t, mivel hiába lett üres a verem, még volt hátra az inputból.

Veremautomata – Ismétlés

Veremautomaták és a 2-es típusú nyelvek

Tétel

Bármely L nyelvre ekivalensek a következő állítások

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

Tétel

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

Chomsky normálforma

Definíció

Egy $G = \langle N, T, P, S \rangle$ grammatika **Chomsky normálformájú**, ha minden P-beli szabály alakja a következők valamelyike:

- $S \to \varepsilon$
- $A \to BC$, $A, B, C \in N$; továbbá $B, C \neq S$, ha $S \to \varepsilon \in P$.
- $A \to a$, $A \in N, a \in T$.

Tétel

Minden G környezetfüggetlen grammatikához létezik vele ekvivalens Chomsky normálformájú grammatika.

Bizonyítás: Több lépésben átalakítjuk G-t a megfelelő alakra ügyelve arra, hogy az egyes átalakítási lépések után kapott grammatika G-vel ekvivalens legyen.

Chomsky normálforma

1. lépés Új kezdőszimbólum bevezetése

Vezessünk be egy új S_0 kezdőszimbólumot és adjuk hozzá az $S_0 \rightarrow S$ szabályt P-hez. A generált nyelv nyilván nem változik, de a kezdőszimbólum már nem fordul elő szabály jobboldalán.

Kivétel: Amennyiben S nem fordul elő szabály jobboldalán, ez a lépés kihagyható.

Megjegyzés: A bizonyítás további része feltételezi, hogy S-sel van jelölve a kezdőszimbólum, ez a nemterminálisok átnevezésével biztosítható.

Chomsky normálforma

2. lépés Álterminálisok bevezetése

Minden $a \in T$ terminálisra végezzük el a következőt. Legyen $\bar{a}$ egy új nemterminális szimbólum és helyettesítsük minden P-beli szabály jobboldalán a előfordulásait $\bar{a}$-val. Adjuk hozzá ezen kívül P-hez a $\bar{a} \to a$ új szabályt.

Az így kapott grammatika az eredetivel ekvialens (ezt az előző félévben már meggondoltuk a zártsági tétel bizonyításánál).

Kivétel: Amennyiben $A \to a \in P$, akkor a-nak ezt az előfordulását nem szükséges átírni, hiszen ez a szabály már kellő alakú.

Chomsky normálforma

3. lépés Hosszredukció

Ekkor minden $X \to Y_1 Y_2 \cdots Y_k$, $k \geqslant 3$ alakú szabályt helyettesítünk egy

$$\{X \to Y_1 Z_1, Z_1 \to Y_2 Z_2, \ldots, Z_{k-2} \to Y_{k-1} Y_k\}$$

szabályhalmazzal, ahol $Z_1, \ldots, Z_{k-2}$ új, a szabályhoz bevezetett nemterminálisok.

Nyilván minden $X \to Y_1 Y_2 \cdots Y_k$ szabályalkalmazás helyettesíthető a fenti szabályokkal. Másrészt minden, valamelyik Z_i-t tartalmazó levezetés az átalakított grammatikában $Z_1, \ldots, Z_{k-2}$ sorrendben minden Z_i-t be kell hozzon majd át kell írjon. Mivel G környezetfüggetlen, így amennyiben a Z_i-t behozó lépést nem közvetlenül követi a Z_i-t átíró lépés, akkor az átíró lépés a levezetésben előrehozható. Tehát feltehető, hogy a Z_i-k a fenti sorrendben, közvetlenül egymást követő lépésekben jönnek be majd íródnak át vagyis az $X \to Y_1 Y_2 \cdots Y_k$ szabály alkalmazását szimulálják.

Chomsky normálforma

4. lépés ε-mentesítés

Definiáljuk az $U_i \subseteq N$ halmazokat a következőképpen:

$$U_1 := \{X \mid X \to \varepsilon \in P\},$$

$$U_{i+1} := U_i \cup \{X \mid X \to u \in P \text{ és } u \in U_i^*\}. \quad i \geqslant 1$$

Nyilvánvaló, hogy az U_i sorozat, $i = 1, 2, \ldots$ a tartalmazásra nézve hierarchiát alkot és van olyan k index, hogy $U_k = U_{k+1}$ és így $U_\ell = U_k$ minden $\ell \geqslant k$-ra.

$U := U_k$.

Állítás: $X \Rightarrow_G^* \varepsilon \iff X \in U$.

Az állítás következménye: $\varepsilon \in L(G) \iff S \in U$.

Chomsky normálforma

Az állítás bizonyítása:

($\Leftarrow$) Minden $1 \leqslant i \leqslant k$-ra (ahol k az első index, melyre $U_k = U_{k+1}$) és minden $X \in U_i$-ra $X \Rightarrow_G^* \varepsilon$.

Ez i-re vonatkozó teljes indukcióval könnyen látható. $i = 1$-re az állítás U_1 definíciójából azonnal következik.

Tegyük fel, hogy i-re teljesül. Ha $X \in U_{i+1}$, akkor van olyan $u \in U_i^*$, melyre $X \to u \in P$. Legyen $u = Z_1 \cdots Z_n$, ahol $Z_r \in U_i$ $(1 \leqslant r \leqslant n)$.

Ekkor az indukciós feltevés miatt $Z_i \Rightarrow_G^* \varepsilon$ $(1 \leqslant r \leqslant n)$. Tehát $X \Rightarrow_G Z_1 \cdots Z_n \Rightarrow_G^* \varepsilon$.

Mivel $U = U_k$, ezért az állításnak ezt az irányát bizonyítottuk.

Chomsky normálforma

Az állítás bizonyítása:

($\Rightarrow$) Ha $X \Rightarrow_G^* \varepsilon$ akkor létezik G-beli $X \Rightarrow w_n \Rightarrow w_{n-1} \Rightarrow \cdots \Rightarrow w_1 \Rightarrow \varepsilon$ levezetés ($n \geqslant 1$).

i-re vonatkozó teljes indukcióval ($1 \leqslant i \leqslant n$) könnyen látható, hogy $w_i \in U_i^*$.

Valóban, $w_1 \in U_1$ következik U_1 definíciójából, hiszen w_1 egyetlen olyan nemterminálisból kell álljon, amire van ε-szabály P-ben.

Tegyük fel, hogy i-re teljesül az állítás. Ekkor a $w_{i+1} \Rightarrow w_i$ levezetési lépésben egy olyan $X \to u \in P$ szabály került alkalmazásra, ahol $u \in U_i^*$, így $X \in U_{i+1}$. Mivel $U_i \subseteq U_{i+1}$, így $w_{i+1} \in U_{i+1}^*$.

Tehát $X \in U_{n+1} \subseteq U$, ezzel az állítást bizonyítottuk.

Chomsky normálforma

Hagyjuk el P-ből az összes $X \to \varepsilon$ alakú ún. **ε-szabályt** és adjuk hozzá a következőket

- $A \to BC \in P, B, C \in U$ esetén az $A \to B$ és $A \to C$ szabályokat
- $A \to BC \in P, B \in U, C \notin U$ esetén az $A \to C$ szabályt
- $A \to BC \in P, B \notin U, C \in U$ esetén az $A \to B$ szabályt

Legyen G' az így kapott grammatika. Ekkor $L(G_1) \subseteq L(G)\backslash\{\varepsilon\}$, hiszen minden új szabály alkalmazása megfelel egy régi szabály alkalmazásának amelyet egy $B \Rightarrow_G^* \varepsilon$ vagy $C \Rightarrow_G^* \varepsilon$ levezetés alkalmazásával kombinálunk.

Másrészt $L(G)\backslash\{\varepsilon\} \subseteq L(G_1)$. Ugyanis, ha $S \Rightarrow_G^* u$ és $u \neq \varepsilon$, akkor $S \Rightarrow_{G'}^* u$, hiszen az $X \to \varepsilon$ típusú szabályok alkalmazása elkerülhető egy megfelelő új szabály alkalmazásával.

Tehát **$S \notin U$ esetén** G_1 ekvivalens G-vel.

$S \in U$ esetén adjuk hozzá G szabályaihoz az $S \to \varepsilon$ szabályt.

Chomsky normálforma

5. lépés Láncmentesítés

Már csak az $X \to Y$ alakú $(X, Y \in N)$, ú.n. **láncszabályokat** kell eliminálni. Jelölje P_0 a P-beli láncszabályok halmazát.

Minden $A \in N$ esetén legyen $H(A) := \{B \in N \mid A \Rightarrow^* B\}$.

A G-vel ekvivalens, láncmentes G' grammatika szabályai:

$P' := \{A \to w \mid \exists B \in H(A), \text{ amelyre } B \to w \in P\} \setminus P_0$.

$L(G') \subseteq L(G)$, hiszen egy $A \to w$ új szabály alkalmazása megfelel az $A \Rightarrow^* B$ láncszabályok és a $B \to w$ eredeti szabály alkalmazásának $(B \in H(A))$.

Másrészt $L(G) \subseteq L(G')$, hiszen az új szabályokkal a láncszabályok alkalmazása elkerülhető.

Chomsky normálforma

Állítás: $H_k(A) = \{B \in N \mid A \Rightarrow^* B\}$.

Az állítás bizonyítása:

($\subseteq$) i-re vonatkozó teljes indukcióval könnyen látható, hogy $H_i(A) \subseteq \{B \in N \mid A \Rightarrow^* B\}$ $(1 \leqslant i \leqslant k)$.

$i = 0$-ra nyilvánvaló. Ha $B \in H_{i+1}(A)$, akkor van olyan $C \in H_i(A)$, hogy $C \to B \in P$. Mivel $C \in H_i(A)$, ezért az indukciós feltevés miatt $A \Rightarrow^* C$, és így $A \Rightarrow^* C \Rightarrow B$, tehát $H_{i+1}(A) \subseteq \{B \in N \mid A \Rightarrow^* B\}$.

($\supseteq$) Tegyük fel, hogy $A \Rightarrow^* B$, ekkor vagy $B = A$ és így $B \in H_0(A)$ vagy $\exists k \geqslant 1$, hogy $A \Rightarrow Z_1 \Rightarrow \cdots \Rightarrow Z_k = B$ valamely $Z_1 \ldots, Z_k \in N$-re.

i-re vonatkozó teljes indukcióval könnyen látható, hogy $Z_i \in H_i(A)$. ($i = 1$-re az indukció kezdőlépése illetve i-ről $i + 1$-re az indukciós lépés is $H_i(A)$ definíciójából azonnal adódik.)

Chomsky normálforma – példa

Példa: Hozzuk az alábbi grammatikát Chomsky normálformára! (a nagybetűk a nemterminálisok, S a kezdőszimólum)

$S \to AB$
$A \to aAa \mid C$
$B \to bBb \mid C$
$C \to Cabc \mid b \mid \varepsilon$

Megoldás:

1. lépés: Nincs S a jobboldalon, maradhat S a kezdőszimbólum.

2. lépés (Álterminálisok bevezetése):

$S \to AB$
$A \to DAD \mid C$
$B \to EBE \mid C$
$C \to CDEF \mid b \mid \varepsilon$
$D \to a$
$E \to b$
$F \to c$

Chomsky normálforma – példa

3. lépés (Hosszredukció):

$$
\begin{aligned}
S &\to AB \\
A &\to DAD \mid C \\
B &\to EBE \mid C \\
C &\to CDEF \mid b \mid \varepsilon \\
D &\to a \\
E &\to b \\
F &\to c
\end{aligned}
\quad \Longrightarrow \quad
\begin{aligned}
S &\to AB \\
A &\to DZ_1 \mid C \\
B &\to EZ_2 \mid C \\
C &\to CZ_3 \mid b \mid \varepsilon \\
D &\to a \\
E &\to b \\
F &\to c \\
Z_1 &\to AD \\
Z_2 &\to BE \\
Z_3 &\to DZ_4 \\
Z_4 &\to EF
\end{aligned}
$$

Chomsky normálforma – példa

4. lépés (ε-mentesítés):

$U_0 = \{C\},\ U_1 = \{C, A, B\}, U_2 = U_3 = \{S, A, B, C\} = U.$

$$
\begin{array}{lcl}
S \rightarrow AB & & S \rightarrow AB \mid A \mid B \mid \varepsilon \\
A \rightarrow DZ_1 \mid C & & A \rightarrow DZ_1 \mid C \\
B \rightarrow EZ_2 \mid C & & B \rightarrow EZ_2 \mid C \\
C \rightarrow CZ_3 \mid b \mid \varepsilon & & C \rightarrow CZ_3 \mid b \mid Z_3 \\
D \rightarrow a & & D \rightarrow a \\
E \rightarrow b & \Longrightarrow & E \rightarrow b \\
F \rightarrow c & & F \rightarrow c \\
Z_1 \rightarrow AD & & Z_1 \rightarrow AD \mid D \\
Z_2 \rightarrow BE & & Z_2 \rightarrow BE \mid E \\
Z_3 \rightarrow DZ_4 & & Z_3 \rightarrow DZ_4 \\
Z_4 \rightarrow EF & & Z_4 \rightarrow EF
\end{array}
$$

Chomsky normálforma – példa

5. lépés (Láncmentesítés):
$H_0(S) = \{S\}, H_1(S) = \{S, A, B\}, H_2(S) = \{S, A, B, C\}, H_3(S) = \{S, A, B, C, Z_3\} = H(S)$

Hasonlóan $H(A) = \{A, C, Z_3\}, H(B) = \{B, C, Z_3\}$,
$H(C) = \{C, Z_3\}$, $H(Z_1) = \{D, Z_1\}$, $H(Z_2) = \{E, Z_2\}$, a többi csak önmagát tartalmazza.

$$
\begin{array}{lcl}
S \to AB \mid A \mid B \mid \varepsilon & & S \to AB \mid DZ_1 \mid CZ_3 \mid b \mid DZ_4 \mid EZ_2 \mid \varepsilon \\
A \to DZ_1 \mid C & & A \to DZ_1 \mid CZ_3 \mid b \mid DZ_4 \\
B \to EZ_2 \mid C & & B \to EZ_2 \mid CZ_3 \mid b \mid DZ_4 \\
C \to CZ_3 \mid b \mid Z_3 & & C \to CZ_3 \mid b \mid DZ_4 \\
D \to a & & D \to a \\
E \to b & \Longrightarrow & E \to b \\
F \to c & & F \to c \\
Z_1 \to AD \mid D & & Z_1 \to AD \mid a \\
Z_2 \to BE \mid E & & Z_2 \to BE \mid b \\
Z_3 \to DZ_4 & & Z_3 \to DZ_4 \\
Z_4 \to EF & & Z_4 \to EF
\end{array}
$$

Chomsky normálforma – az algoritmus hatékonysága

Legyen $G = \langle N, T, P, S \rangle$ egy környezetfüggetlen grammatika. Ha $p : X \to Y_1 \cdots Y_m \in P$, akkor legyen $|p| := m + 1$, a p szabály bal- és jobboldalának összhossza. Jelölje $|G| := \sum_{p \in P} |p|$ a G grammatika **méretét**.

A Chomsky normálformára hozás hatékonysága:
A fenti algoritmus nagyságrendileg $|G|^2$ lépésben (precízebben: $O(|G|^2)$ lépésben, lásd később) előállít egy nagyságrendileg legfeljebb $|G|^2$ méretű (precízebben: $O(|G|^2)$ méretű) G-vel ekvivalens Chomsky normálformájú grammatikát.

Megjegyzés: A méretnövekedés a láncmentesítés kivételével lineáris.

Számításelmélet

2. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Hossz-nemcsökkentő grammatika

Definíció

Egy $G = \langle N, T, P, S \rangle$ grammatikát **hossz-nemcsökkentőnek** mondunk, ha minden szabályának alakjára az alábbiak valamelyike teljesül

- $S \to \varepsilon$, de ha van ilyen szabály, akkor más szabály nem tartalmazhatja jobboldalán S-et.
- $u \to v$, ahol $u, v \in (T \cup N)^+$ és $|u| \leqslant |v|$.

A környezetfüggő grammatikák nyilván hossz-nemcsökkentőek.

Hossz-nemcsökkentő grammatika

Tétel

Minden hossz-nemcsökkentő grammatika környezetfüggő nyelvet generál.

Bizonyítás: (vázlat) Minden hossz-nemcsökkentő $G = \langle N, T, P, S \rangle$ grammatikához megadható egy vele ekvivalens $G' = \langle N', T, P', S \rangle$ környezetfüggő grammatika.

1. lépés: Álterminálisok bevezetése
A szokásos módon feltehető, terminálisok csak $A \to a$ $(A \in N, a \in T)$ alakú szabályok jobboldalán fordulnak elő.

Hossz-nemcsökkentő grammatika

2. lépés: Környezetfüggő szabályokkal való helyettesítés

Legyen $X_1X_2\cdots X_n \to Y_1Y_2\cdots Y_m \quad (m \geqslant n)$ egy hossz-nemcsökkető szabály. Ezt az alábbi csupa 1-típusú szabályokkal szimulálhatjuk:

$$
\begin{gathered}
X_1X_2\cdots X_n \to Z_1X_2\cdots X_n,\\
Z_1X_2\cdots X_n \to Z_1Z_2X_3\cdots X_n,\\
\vdots\\
Z_1Z_2\cdots Z_{n-1}X_n \to Z_1Z_2\cdots Z_nY_{n+1}\cdots Y_m \quad (n \leqslant m),\\
Z_1Z_2\cdots Z_nY_{n+1}\cdots Y_m \to Y_1Z_2\ldots Z_nY_{n+1}\cdots Y_m,\\
\vdots\\
Y_1\cdots Y_{n-1}Z_nY_{n+1}\cdots Y_m \to Y_1Y_2\cdots Y_m,
\end{gathered}
$$

ahol $Z_1, Z_2, \ldots, Z_n$ új nemterminálisok.

Hossz-nemcsökkentő grammatika

Meggondolható, hogy a $Z_1, \ldots, Z_n$ új volta miatt a szabályokat csak ebben a sorrendben lehet és kell végrehajtani, ezért az új grammatika is ugyanazt a nyelvet generálja. Csináljuk meg ezt a szabálytranszformációt az összes „rossz” szabályra. Az így kapott G' grammatika már 1-típusú és $L(G)$-t generálja.

1. Példa (csak egyetlen hosszú szabály):

Az $ABC \to DEFGH$ szabály a következő környezetfüggő szabályokkal helyettesíthető:

$ABC \to XBC$

$XBC \to XYC$

$XYC \to XYZGH$

$XYZGH \to DYZGH$

$DYZGH \to DEZGH$

$DEZGH \to DEFGH$

(X, Y, Z új nemterminálisok)

Hossz-nemcsökkentő grammatika

2. Példa: $G = \langle \{S, B\}, \{a, b, c\}, P, S \rangle$

$P = \{S \to abc, S \to aSBc, cB \to Bc, bB \to abb\}$

Egy a G-vel ekvivalens 1-es típusú grammatika szabályai:

$S \to DEF$
$S \to DSBF$
$FB \to Z_1B$
$Z_1B \to Z_1F$
$Z_1F \to BF$

$EB \to Z_2B$
$Z_2B \to Z_2EE$
$Z_2EE \to DEE$
$D \to a$
$E \to b$
$F \to c$

(D, E, F, Z_1, Z_2 új nemterminálisok)

Kuroda normálforma

Definíció

Egy $G = \langle N, T, P, S \rangle$ grammatikát **Kuroda normálformájúnak** mondunk, ha minden szabályának alakjára az alábbiak valamelyike teljesül

- $S \to \varepsilon$, de ha van ilyen szabály, akkor más szabály nem tartalmazhatja jobboldalán S-et.
- $A \to a$, ahol $A \in N, a \in T$,
- $A \to BC$, ahol $A, B, C \in N$,
- $AB \to AC$, ahol $A, B, C \in N$,
- $BA \to CA$, ahol $A, B, C \in N$.

A Kuroda normálformájú grammatikák nyilván környezetfüggőek.

Tétel

Minden környezetfüggő grammatika G grammatikához van vele ekvivalens Kuroda normálformájú G' grammatika.

Kuroda normálforma

Bizonyítás: (vázlat)

1. lépés: Álterminálisok bevezetése

A szokásos módon feltehető, terminálisok csak $A \to a$ $(A \in N, a \in T)$ alakú szabályok jobboldalán fordulnak elő.

2. lépés: Környezetfüggetlen szabályok hosszredukciója

Szintén a Chomsky normálformánál látott módon.

3. lépés: Környezetfüggő láncmentesítés

Az A-ból láncszabályokkal elérhető nemterminálisok $H(A) = \{B \in N \mid A \Rightarrow^* B\}$ halmazának meghatározása a Chomsky normálformánál látott módon. A szabályrendszer módosítása:

$$P' := \{A_1 \cdots A_n \to w \mid w \notin N \wedge \exists B_1 \cdots B_n \to w \in P : B_i \in H(A_i)\ (\forall 1 \leqslant i \leqslant n)\}.$$

4. lépés: Környezetfüggő szabályok hosszredukciója

Az $X_1 \cdots X_m \to Y_1 \cdots Y_n$ alakú szabályok szimulációja, ahol $n \geqslant m \geqslant 2$.

Kuroda normálforma

Ha $n = m = 2$, akkor a következő lépésre ugorhatunk. Különben a szabály szimulációja a $Z_1, Z_2, \ldots, Z_{n-2}$ új nemterminálisok bevezetésével:

$$\begin{gathered} X_1X_2 \to Y_1Z_1, \\ Z_1X_3 \to Y_2Z_2, \\ \vdots \\ Z_{m-3}X_{m-1} \to Y_{m-2}Z_{m-2}, \end{gathered}$$

Továbbá ha $n = m$, akkor

$$Z_{m-2}X_m \to Y_{m-1}Y_m,$$

egyébként $(n > m)$ esetén:

$$\begin{gathered} Z_{m-2}X_m \to Y_{m-1}Z_{m-1}, \\ Z_{m-1} \to Y_mZ_m, \\ \vdots \\ Z_{n-3} \to Y_{n-2}Z_{n-2}, \\ Z_{n-2} \to Y_{n-1}Y_n. \end{gathered}$$

Kuroda normálforma

5. lépés: Az $AB \to CD, \ A \neq C, B \neq D$ szabályok eliminációja

Végül a nem Kuroda-normálformájú szabályok sémája ekkor $AB \to CD$ $(A, B, C, D \in N)$. Átalakításukhoz szabályonként egyedi W új nemterminálisokat vezetünk be és a fenti szabályt az alábbi szabályokkal szimuláljuk:

$$AB \to AW,$$
$$AW \to CW,$$
$$CW \to CD.$$

A kapott G' grammatika ekvivalens G-vel. Ugyanis az átalakított grammatikában ezen 3 szabály bármelyikének alkalmazása implikálja a másik 2 alkalmazását ebben a sorrendben. Meggondolható, hogy az $AB \to AW$ szabályalkalmazás hátratolható közvetlenül az $AW \to CW$ szabályalkalmazás elé, míg a $CW \to CD$ szabályalkalmazás előrehozható közvetlenül az $AW \to CW$ szabályalkalmazás utánra.

Kuroda normálforma – példa

Példa:

$S \rightarrow C \mid AABC$
$A \rightarrow ABC \mid a$
$B \rightarrow b$
$C \rightarrow B \mid bA$
$ABC \rightarrow ABaC$

1-2. lépés után:

$S \rightarrow C \mid AD$
$A \rightarrow AF \mid a$
$B \rightarrow b$
$C \rightarrow B \mid YA$
$ABC \rightarrow ABXC$
$D \rightarrow AE$
$E \rightarrow BC$
$F \rightarrow BC$
$X \rightarrow a$
$Y \rightarrow b$

Kuroda normálforma – példa

3. lépés: $H(S) = \{S, C, B\}$, $H(C) = \{C, B\}$, minden más Z nemterminálisra $H(Z) = \{Z\}$.

$S \to AD \mid YA \mid b$
$A \to AF \mid a$
$B \to b$
$C \to YA \mid b$
$ABC \to ABXC$
$ACC \to ABXC$
$ASC \to ABXC$
$ABS \to ABXC$
$ACS \to ABXC$
$ASS \to ABXC$
$D \to AE$
$E \to BC$
$F \to BC$
$X \to a$
$Y \to b$

Kuroda normálforma – példa

4. lépés:

$S \to AD \mid YA \mid b$

$A \to AF \mid a$

$B \to b$

$C \to YA \mid b$

$AB \to AZ_1 \quad Z_1C \to BZ_2 \quad Z_2 \to XC$

$AC \to AZ_3 \quad Z_3C \to BZ_4 \quad Z_4 \to XC$

$AS \to AZ_5 \quad Z_5C \to BZ_6 \quad Z_6 \to XC$

$AB \to AZ_7 \quad Z_7S \to BZ_8 \quad Z_8 \to XC$

$AC \to AZ_9 \quad Z_9S \to BZ_{10} \quad Z_{10} \to XC$

$AS \to AZ_{11} \quad Z_{11}S \to BZ_{12} \quad Z_{12} \to XC$

$D \to AE$

$E \to BC$

$F \to BC$

$X \to a$

$Y \to b$

Kuroda normálforma – példa

5. lépés:

$S \to AD \mid YA \mid b$
$A \to AF \mid a$
$B \to b$
$C \to YA \mid b$
$AB \to AZ_1 \quad Z_1C \to Z_1W_1 \quad Z_1W_1 \to BW_1 \quad BW_1 \to BZ_2 \quad Z_2 \to XC$
$AC \to AZ_3 \quad Z_3C \to Z_3W_2 \quad Z_3W_2 \to BW_2 \quad BW_2 \to BZ_4 \quad Z_4 \to XC$
$AS \to AZ_5 \quad Z_5C \to Z_5W_3 \quad Z_5W_3 \to BW_3 \quad BW_3 \to BZ_6 \quad Z_6 \to XC$
$AB \to AZ_7 \quad Z_7S \to Z_7W_4 \quad Z_7W_4 \to BW_4 \quad BW_4 \to BZ_8 \quad Z_8 \to XC$
$AC \to AZ_9 \quad Z_9S \to Z_9W_5 \quad Z_9W_1 \to BW_5 \quad BW_5 \to BZ_{10} \quad Z_{10} \to XC$
$AS \to AZ_{11} \quad Z_{11}S \to Z_{11}W_6 \quad Z_{11}W_6 \to BW_6 \quad BW_6 \to BZ_{12}$
$D \to AE \qquad Z_{12} \to XC$
$E \to BC$
$F \to BC$
$X \to a$
$Y \to b$

Egy 0-típusú normálforma

Tétel

Bármely $G = \langle N, T, P, S\rangle$ 0-típusú grammatikához van vele ekvivalens G' grammatika, ahol G' minden szabályának alakjára az alábbiak valamelyike teljesül

- $S \rightarrow \varepsilon$, de ha van ilyen szabály, akkor más szabály nem tartalmazhatja jobboldalán S-et.
- $A \rightarrow a$, ahol $A \in N, a \in T$,
- $A \rightarrow B$, ahol $A, B \in N$,
- $A \rightarrow BC$, ahol $A, B, C \in N$,
- $AB \rightarrow B$, ahol $A, B \in N$,
- $AB \rightarrow AC$, ahol $A, B, C \in N$,
- $BA \rightarrow CA$, ahol $A, B, C \in N$.

A tételt nem bizonyítjuk.

Megjegyzés: A 0-típusú grammatikáknak többfajta normálformája ismeretes. Ez egy ε-szabály mentes változat.

Algoritmikus problémák

Akkor beszélünk algoritmikus eldöntési problémáról, ha a bemenet egy olyan objektum, mely egy adott végtelen halmazból kerülhet ki (például egy természetes szám vagy adott ábécé feletti szó), kimenete pedig igen/nem.

Olyan algoritmust keresünk, amely kellően általános ahhoz, hogy a végtelen lehetséges bemenet bármelyike esetén alkalmazható legyen.

Egy probléma **eldönthető**, ha létezik olyan minden lehetséges bemenet esetén termináló algoritmus, amelyik a helyes választ adja.

Megjegyzés: Az eldönthetőség fogalmát a későbbiekben precízebben definiáljuk.
Előző szemeszterben tanultuk:

Tétel

$\mathcal{L}_i$ zárt a reguláris bűveletekre (unió, konkatenáció, Kleene-lezárt) $(i = 0, 1, 2, 3)$

$\mathcal{L}_3$ további zártsági tulajdonságai

Tétel

$\mathcal{L}_3$ zárt a komplementerre, a metszetre és a különbségre.

Bizonyítás:

- Legyen $L \in \mathcal{L}_3$, ekkor az előző tételek alapján L felismerhető egy $A = \langle Q, T, \delta, q_0, F \rangle$ véges determinisztikus automatával.

  Nyilván az $A' = \langle Q, T, \delta, q_0, Q \setminus F \rangle$ véges determinisztikus automata által felismert nyelv $\overline{L}$.

  Mivel $\overline{L}$ felismerhető véges determinisztikus automatával, ezért korábbi tételünk miatt $\overline{L} \in \mathcal{L}_3$.
- Mivel $L_1 \cap L_2 = \overline{\overline{L}_1 \cup \overline{L}_2}$, ezért a metszetre való zártság következik az unióra és a komplementerre való zártságból.
- Mivel $L_1 \setminus L_2 = L_1 \cap \overline{L}_2$, ezért a különbségre való zártság következik a metszetre és a komplementerre való zártságból.

Megjegyzés: $\mathcal{L}_2$ nem zárt a metszetre. Annyi viszont igaz, hogy egy 3-típusú és egy 2-típusú nyelv metszete 2-típusú.

3-as típusú grammatikák algoritmikus problémái

Állítás

Eldönthető, hogy egy G reguláris grammatika az üres nyelvet generálja-e.

Bizonyítás: Az FNyF-ben tanultak szerint G-hez algoritmikusan megadható egy $L(G)$-t felismerő A VDA. A pontosan akkor nem ismer fel egyetlen szót sem, ha a kezdőállapotából minden végállapota elérhetetlen. A átmeneti gráfja véges, így ez például A gráfján egy szélességi kereséssel algoritmikusan eldönthető.

Állítás

Eldönthető, hogy két reguláris grammatika által generált nyelv diszjunkt-e.

Bizonyítás: Generálják a G_1 és G_2 reguláris grammatikák rendre az L_1 és L_2 nyelveket. Az előbbi tétel szerint $L_1 \cap L_2$ is reguláris, így az előző állítás szerint ennek üressége eldönthető.

3-as típusú grammatikák algoritmikus problémái

Állítás

Eldönthető, hogy két reguláris grammatika ugyanazt a nyelvet generálja-e vagy sem.

Bizonyítás:
Generálják a G_1 és G_2 reguláris grammatikák rendre az L_1 és L_2 nyelveket. Az $L_3 = (L_1 \cap \overline{L}_2) \cup (\overline{L}_1 \cap L_2)$ nyelv szintén reguláris, így van olyan G_3 reguláris grammatika, amely L_3-at generálja. Ekkor azonban $L_1 = L_2$ akkor és csak akkor, ha $L_3 = \emptyset$, amely a fentiek szerint eldönthető.

Állítás

Eldönthető, hogy egy reguláris grammatika bővebb vagy egyenlő nyelvet generál-e mint egy másik.

Bizonyítás: Generálják a G_1, G_2 reguláris grammatikák rendre az L_1, L_2 nyelveket. Az $L_3 = (L_1 \cap \overline{L}_2) = L_1 \setminus L_2$ szintén reguláris. Ennek üressége eldönthető, ami ekvivalens azzal, hogy $L_1 \subseteq L_2$.

Lineáris algoritmus a 3-as típusú szóproblémára

Szóprobléma (membership problem):

Adott a G grammatika és $u \in T^*$. $u \stackrel{?}{\in} L(G)$.

Tétel

A 3-as típusú grammatikák szóproblémája (hatékonyan) eldönthető.

Bizonyítás: Legyen $G = \langle N, T, P, S\rangle$ egy 3-as típusú grammatika normálformában adva és $u = t_1 \cdots t_n$ a levezetendő szó.

Az algoritmus rekurzívan kiszámol egy N részhalmazaiból álló H_i $(0 \leqslant i \leqslant n)$ sorozatot. H_i-ból H_{i+1} (csak G-től függő) konstans időben számolható.

$H_0 := \{S\} \quad H_{i+1} := \{A \in N \mid \exists\, B \in H_i \wedge B \to t_{i+1}A \in P\}$.

Könnyen látható, hogy H_i azon nemterminálosok halmaza, melyek pontosan i levezetési lépés után a mondatforma végén állhatnak.

Tehát ha $F = \{A \in N \mid A \to \varepsilon \in P\}$, akkor nyilván

$$u \in L(G) \Leftrightarrow H_n \cap F \neq \emptyset.$$

Lineáris algoritmus a 3-as típusú szóproblémára

Példa:

$S \to aA \,|\, bS \,|\, \varepsilon$
$A \to aA \,|\, aS$

abb-re $H_0 = \{S\}, H_1 = \{A\}, H_2 = \emptyset, H_3 = \emptyset$,

míg aab-re $H_0 = \{S\}$, $H_1 = \{A\}, H_2 = \{A, S\}, H_3 = \{S\}$

lesz a H_i-k sorozata. Mivel most $F = \{S\}$, ezért aab generálható, abb viszont nem.

Vegyük észre, hogy $H_i \in \mathcal{P}(N)$ és minden $1 \leqslant i \leqslant n-1$-re H_{i+1} csak H_i-től és t_{i+1}-től függ.Azaz a $\{H_i\}_{0 \leqslant i \leqslant n}$ halmazok meghatározása egy VDA segítségével automatizálható.

	a	b
$\leftrightarrows \{S\}$	$\{A\}$	$\{S\}$
$\{A\}$	$\{S, A\}$	$\{\}$
$\leftarrow \{S, A\}$	$\{S, A\}$	$\{S\}$
$\{\}$	$\{\}$	$\{\}$

Lineáris algoritmus a 3-as típusú szóproblémára

Megjegyzések:

- Ez a VDA nem más, mint a G-hez készített VNDA determinizáltja. $\mathcal{P}(N)$ állapothalmazzal, $H_0 = \{S\}$ kezdőállapottal, $\delta(H, a) = \{B \in N \mid \exists\, A \in H : A \to aB \in P\}$ állapotátmenet-függvénnyel ($H \in \mathcal{P}(N)$ és $a \in T$) és $F' = \{H \subseteq N \mid H \cap F \neq \emptyset\}$ elfogadó állapothalmazzal, ahol $F = \{A \in N \mid A \to \varepsilon \in P\}$.
- Az algoritmus **lineáris** időben eldönti, $u \overset{?}{\in} L(G)$, hiszen minden egyes H_i halmaz kiszámításának ideje csak G-től ($|u|$-tól nem) függő konstans.
- Amennyiben G méretéhez képest hosszú szóra vagy több szóra szeretnénk a kérdést eldönteni érdemes lehet előfeldolgozó lépésként a fenti automatát elkészíteni.
- Másrészt, kevés, kisméretű szó és nagyméretű grammatika esetén hatékonyabb lehet a H_i halmazok közvetlen kiszámítása.

A környezetfüggetlen grammatikák szóproblémája

Állítás: Legyen $G = \langle T, N, P, S\rangle$ környezetfüggetlen grammatika és egy $u \in T^*$ egy szó. Eldönthető, hogy $u \overset{?}{\in} L(G)$.

Bizonyítás: Tanultuk, hogy algoritmikusan előállítható egy G-vel ekvivalens Chomsky normálformájú grammatika. Legyen $n = |u|$.

$n = 0$ esetén $u \in L(G) \Leftrightarrow S \to \varepsilon \in P$.

Teljes indukcióval könnyen látható, hogy $n \geqslant 1$ esetén u levezetése $n-1$ darab „$A \to BC$" típusú és n darab „$A \to a$" típusú szabály alkalmazásából kell álljon. Mivel adott grammatika és n esetén véges sok ilyen levezetés van ezek megvizsgálása után el tudjuk dönteni, hogy $u \overset{?}{\in} L(G)$.

A környezetfüggetlen grammatikák szóproblémája

Az előző algoritmus alapján csak exponenciális felső becslést adhatunk a szóprobléma eldöntésének hatékonyságára.

u levezetéseinek keresését azonban hatékonyabban is megszervezhetjük egy „bottom-up" ún. „dinamikus programozás" segítségével, az így kapott algoritmus már polinomiális (azaz hatékony) lesz.

John Cocke, Daniel Younger és Tadao Kasami algoritmusa (CYK algoritmus) egy Chomsky normálformában adott grammatika esetében $|G| \cdot n^3$ nagyságrendű lépésben eldönti a szóproblémát.

Tetszőleges környezetfüggetlen grammatika esetén tehát a Chomsky normálformára hozással kombinálva a CYK algoritmus hatékonyságára $|G|^2 \cdot n^3$ nagyságrendű felső becslés adható.

CYK algoritmus 2-es típusú szóproblémára

Cocke-Younger-Kasami (CYK) algoritmus

Input: Egy környezetfüggetlen $G = \langle T, N, P, S \rangle$ grammatika **Chomsky-normálformában** adva és egy $u \in T^*$ szó.

Output: $u \overset{?}{\in} L(G)$

Legyen $u = t_1 \ldots t_n,\ t_i \in T, n \geqslant 1$. Legyen A_i a $P_i \in P$ szabály bal-, β_i pedig a jobboldala. ($1 \leqslant i \leqslant |P|$, $A_i \in N$, $\beta_i \in T \cup N^2$.)

A CYK algoritmus rekurzíven definiál $H_{i,j},\ 1 \leqslant i \leqslant j \leqslant n$ halmazokat $(j-i)$ szerint növekvő sorrendben.

$$H_{i,i} := \{A_j \mid \beta_j = t_i\}$$

$$H_{i,j} := \{A_k \mid \beta_k \in \bigcup_{h=i}^{j-1} H_{i,h} H_{h+1,j}\} \quad (i < j)$$

$u \in L(G) \iff S \in H_{1,n}$.

A CYK algoritmus helyessége

Az algoritmus helyességéhez elég belátni, hogy minden $1 \leqslant i \leqslant j \leqslant n$ esetén $H_{i,j} = \{X \in N \mid X \Rightarrow_G^* t_i \cdots t_j\}$.

Ezt $(j-i)$-re vonatkozó teljes indukcióval bizonyítjuk.

$j - i = 0$ esetén világos, hogy t_i éppen $H_{i,i}$ nemterminálisaiból vezethető le.

Tegyük fel, hogy az állítás igaz minden olyan $1 \leqslant k \leqslant \ell \leqslant n$-re, melyre $\ell - k < j - i$ és legyen $1 \leqslant i < j \leqslant n$.

Tekintsük egy $X \Rightarrow^* t_i \cdots t_j$ levezetést. Mivel a levezetendő szó legalább 2 hosszú ezért a levezetés első lépése $X \Rightarrow YZ$ valamely $Y, Z \in N$-re. Ekkor létezik egy olyan $i \leqslant h < j$, melyre $Y \Rightarrow^* t_i \cdots t_h$ és $Z \Rightarrow^* t_{h+1} \cdots t_j$. Mivel $h - i < j - i$ és $j - (h+1) < j - i$ ezért az indukciós feltevés szerint $Y \in H_{i,h}, Z \in H_{h+1,j}$ és így $X \in H_{i,j}$.

Fordítva, ha $X \in H_{i,j}$ $(j > i)$, akkor van olyan $i \leqslant h < j$ és $Y \in H_{i,h}, Z \in H_{h+1,j}$, melyre $X \to YZ \in P$, azaz $Y \Rightarrow^* t_i \cdots t_h$ és $Z \Rightarrow^* t_{h+1} \cdots t_j$. Ekkor $X \Rightarrow YZ \Rightarrow^* t_i \cdots t_h Z \Rightarrow^* t_i \cdots t_j$.

CYK algoritmus

Példa: A $G = \langle\{S, A, B, C, U, V, W, X, Y, Z\}, \{a, b, c\}, P, S\rangle$ Chomsky normálformájú grammatika esetén CYK algoritmussal döntsük el, hogy az *aabbcc* szót generálja-e G, ahol P:

$S \rightarrow AB \mid BC$
$A \rightarrow XA \mid a$
$X \rightarrow a$
$C \rightarrow YC \mid c$
$Y \rightarrow c$
$B \rightarrow UV \mid VW \mid XS$
$U \rightarrow XX$
$W \rightarrow YY \mid XS$
$V \rightarrow ZZ$
$Z \rightarrow b$

$$
\begin{array}{cccccc}
 & & & H_{1,n} & & \\
 & & H_{1,n-1} & & H_{2,n} & \\
 & & & \vdots & & \\
 & H_{1,3} & H_{2,4} & & & \\
H_{1,2} & H_{2,3} & & & \cdots & H_{n-1,n} \\
H_{1,1} & H_{2,2} & & & \cdots & H_{n,n} \\
\hline
t_1 & t_2 & & \cdots & & t_n
\end{array}
$$

CYK algoritmus

$$
\begin{aligned}
S &\rightarrow AB \mid BC \\
A &\rightarrow XA \mid a \\
X &\rightarrow a \\
C &\rightarrow YC \mid c \\
Y &\rightarrow c \\
B &\rightarrow UV \mid VW \mid XS \\
U &\rightarrow XX \\
W &\rightarrow YY \mid XS \\
V &\rightarrow ZZ \\
Z &\rightarrow b
\end{aligned}
$$

$\{S, B, W\}$					
$\{S\}$	$\{S\}$				
$\{B\}$	$\{\ \}$	$\{B\}$			
$\{\ \}$	$\{\ \}$	$\{\ \}$	$\{\ \}$		
$\{A, U\}$	$\{\ \}$	$\{V\}$	$\{\ \}$	$\{C, W\}$	
$\{A, X\}$	$\{A, X\}$	$\{Z\}$	$\{Z\}$	$\{Y, C\}$	$\{Y, C\}$
a	a	b	b	c	c

$\{AS, XS\}$ $\{AB, UB\}$ $\{\ \}$ $\{BC, BW\}$ $\{SY, SC\}$

CYK algoritmus

$$
\begin{aligned}
S &\to AB \mid BC \\
A &\to XA \mid a \\
X &\to a \\
C &\to YC \mid c \\
Y &\to c \\
B &\to UV \mid VW \mid XS \\
U &\to XX \\
W &\to YY \mid XS \\
V &\to ZZ \\
Z &\to b
\end{aligned}
$$

$$
\begin{array}{ccccccccccc}
 & & & & & \{\boxed{S}, B, W\} & & & & & \\
 & & & & \{S\} & & \{S\} & & & & \\
 & & & \{B\} & & \{\ \} & & \{B\} & & & \\
 & & \{\ \} & & \{\ \} & & \{\ \} & & \{\ \} & & \\
 & \{A, U\} & & \{\ \} & & \{V\} & & \{\ \} & & \{C, W\} & \\
\{A, X\} & & \{A, X\} & & \{Z\} & & \{Z\} & & \{Y, C\} & & \{Y, C\} \\
\hline
a & & a & & b & & b & & c & & c
\end{array}
$$

Mivel $S \in H_{1,6}$, ezért $aabbcc \in L(G)$.
Visszafejthető például a következő levezetés:
$S \Rightarrow AB \Rightarrow XAB \Rightarrow XAVW \Rightarrow XAZZW \Rightarrow XAZZYY \Rightarrow^* aabbcc$.

2-típusú grammatikák algoritmikus kérdései

Következmény

Az alábbi algoritmikus kérdések eldönthetők

1. Adott egy G 2-es típusú grammatika és egy L véges nyelv. $L \overset{?}{\subseteq} L(G)$.
2. Adott egy G 2-es típusú grammatika és egy L véges nyelv. $L \cap L(G) \overset{?}{=} \emptyset$.

FNyF-en tanultuk, hogy az alábbi algoritmikus kérdések szintén eldönthetők (nagy Bar-Hillel lemma következményei):

Tétel

1. Eldönthető, hogy egy környezetfüggetlen grammatika végtelen nyelvet generál-e vagy sem.
2. Eldönthető, hogy egy környezetfüggetlen grammatika az üres nyelvet generálja-e vagy sem.

Algoritmikus eldönthetetlen 2-típusú kérdések

A 2-es típusú grammatikák esetében **vannak algoritmikusan eldönthetetlen problémák**, azaz olyan eldöntési („igen”/„nem” kimenetű) problémák, amelyekre nem létezik olyan mindig termináló algoritmus, amelyik éppen a probléma „igen”-példányaira ad igen választ. Ezekre még visszatérünk egy későbbi előadáson.

1. **Input:** G_1, G_2 2-típusú grammatikák
   Output: $L(G_1) \stackrel{?}{=} L(G_2)$
2. **Input:** G_1, G_2 2-típusú grammatikák
   Output: $L(G_1) \stackrel{?}{\subseteq} L(G_2)$
3. **Input:** G_1, G_2 2-típusú grammatikák
   Output: $L(G_1) \cap L(G_2) \stackrel{?}{=} \varnothing$
4. **Input:** $G = \langle N, T, P, S \rangle$ 2-típusú grammatika
   Output: $L(G) \stackrel{?}{=} T^*$
5. **Input:** G 2-típusú grammatika
   Output: G egyértelmű grammatika-e.

1-típusú grammatikák szóproblémája

Tétel

Eldönthető, hogy egy $G = \langle N, T, P, S \rangle$ hossz-nemcsökkentő grammatika és $u \in T^*$ szó esetén $u \in L(G)$ teljesül-e.

Bizonyítás: Ha $u = \varepsilon$, akkor $u \in L(G) \Leftrightarrow S \to \varepsilon \in P$.

$n = |u| \geqslant 1$ esetén legyen $r = \sum_{i=1}^{n} |T \cup N|^i$. Ekkor r a $T \cup N$ halmaz legfeljebb n hosszú, nemüres szavainak száma.

Mivel G hossz-nemcsökkentő, ezért u levezetései nem tartalmaznak n-nél hosszabb mondatformát,így u minden r-nél hosszabb levezetése tartalmaz ismétlődő mondatformát.

Ebből következően ha $S \Rightarrow_G^* u$, akkor u-nak létezik legfeljebb r hosszú levezetése is, hiszen egy levezetésben az ismétlődő mondatformák közötti levezetést kihagyva ugyanannak a szónak egy rövidebb levezetését kapjuk.

Tehát $u \in L(G)$, akkor és csak akkor, ha G legfeljebb r hosszú levezetéssel generálható. Ezek viszont algoritmikusan előállíthatók.

0-típusú grammatikák szóproblémája

Kevés pozitív eredmény mondható 0-típusú nyelvek algoritmikus kérdései kapcsán.

Bizonyítás nélkül megemlítjük, hogy a 0-típusú grammatikák szóproblémája algoritmikusan eldönthetetlen.

Egy parciálisan eldöntő (helyes választ adó, azonban nem mindig termináló) algoritmust azonban készíthetünk.

Legyen $G = \langle N, T, P, S \rangle$ 0-típusú grammatika és $u \in T^*$. Készíthetünk egy végtelen gráfot, melynek csúcsai $(N \cup T)^*$ elemeivel címkézettek és $x \in (N \cup T)^*$-ból akkor és csak akkor van irányított él $y \in (N \cup T)^*$-ba, ha $x \Rightarrow_G y$. A gráf minden csúcsa véges kifokú, hiszen P véges és egy szabály baloldala legfeljebb annyiszor fordulhat elő a mondatformában, mint amennyi a szó hossza.

Tehát $u \in L(G)$ akkor és csak akkor, ha ebben a gráfban egy S-ből indított szélességi keresés megtalálja u-t. (Ha $u \notin L(G)$, akkor tipikusan nem terminál az algoritmus.)

Számításelmélet

3. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Vázlatos tematika

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

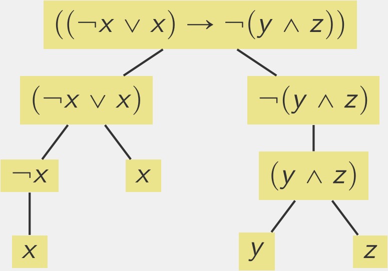
Ítéletkalkulus (nulladrendű logika)

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

Beláthatók olyan következtetések, mint:

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

Szerkezeti fa, részformula, fő logikai összekötő



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

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

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

Zárójelelhagyás

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

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

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

Láncformulák zárójelelhagyása:

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

Interpretáció

Definíció

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

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

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

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

A formulák igazságértéke

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

Definíció

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

ahol a műveleteket eredményét az alábbi táblázat definiálja.

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

Az ítélettábla

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

Definíció

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

Példa:

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

Formulák szemantikus tulajdonságai

Definíció

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

Formulák szemantikus tulajdonságai

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

Fontosabb logikai törvények

$\top$: tautológia, $\perp$: kielégíthetetlen formula.

(a) $\neg\neg\varphi \sim_0 \varphi$,

(b) $\varphi \vee \varphi \sim_0 \varphi$ valamint $\varphi \wedge \varphi \sim_0 \varphi$,

(c) $\varphi \vee \psi \sim_0 \psi \vee \varphi$ valamint $\varphi \wedge \psi \sim_0 \psi \wedge \varphi$,

(d) $(\varphi \vee \psi) \vee \xi \sim_0 \varphi \vee (\psi \vee \xi)$ valamint
$(\varphi \wedge \psi) \wedge \xi \sim_0 \varphi \wedge (\psi \wedge \xi)$,

(e) $(\varphi \vee \psi) \wedge \xi \sim_0 (\varphi \wedge \xi) \vee (\psi \wedge \xi)$ valamint
$(\varphi \wedge \psi) \vee \xi \sim_0 (\varphi \vee \xi) \wedge (\psi \vee \xi)$,

(f) $(\varphi \vee \psi) \wedge \psi \sim_0 \psi$ valamint $(\varphi \wedge \psi) \vee \psi \sim_0 \psi$,

(g) $\varphi \to \psi \sim_0 \neg\varphi \vee \psi$,

(h) $\neg(\varphi \wedge \psi) \sim_0 \neg\varphi \vee \neg\psi$ valamint $\neg(\varphi \vee \psi) \sim_0 \neg\varphi \wedge \neg\psi$,

(i) $\varphi \vee \neg\varphi \sim_0 \top$ valamint $\varphi \wedge \neg\varphi \sim_0 \perp$,

(j) $\varphi \vee \top \sim_0 \top$ valamint $\varphi \wedge \perp \sim_0 \perp$,

(k) $\varphi \vee \perp \sim_0 \varphi$ valamint $\varphi \wedge \top \sim_0 \varphi$.

Formulák szemantikus tulajdonságai

Állítás

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

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

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

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

Formulahalmazok szemantikus tulajdonságai

Definíció

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

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

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

A szemantikus fogalmak egymással való kapcsolata

Tétel

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

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

Bizonyítás:

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

A konjunktív normálforma

Definíció

- **Literálnak** nevezünk egy x vagy $\neg x$ alakú formulát, ahol $x \in \mathrm{Var}$. x és $\neg x$ **komplemens literálpár**. Egy literál **alapja** az az ítéletváltozó, amelyik a literálban szerepel.
- **Elemi diszjunkciónak** (vagy röviden **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 $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ ($m \geqslant 1$) alakú formulát, ahol minden $1 \leqslant i \leqslant m$-re C_i egy klóz (a KNF egy **tagja**).
- Az **elemi konjunkciót** és a **diszjunktív normálformát** (DNF) ezzel analóg módon definiáljuk $\wedge$ és $\vee$ szerepének felcserélésével.

Példa:

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

A diszjunktív normálforma

Tétel

Minden φ ítéletkalkulusbeli formulához megadható egy vele tautológikusan ekvivalens DNF.

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

Ekkor minden $I \in \varphi^i$ esetén

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

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

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

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

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

A konjunktív normálforma

Tétel

Minden φ ítéletkalkulusbeli formulához megadható egy vele tautologikusan ekvivalens KNF.

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

Ekkor minden $I \in \varphi^h$ esetén

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

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

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

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

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

A konjunktív normálforma

Példa: Legyen $\varphi = (x \to y) \to z$. φ ítélettáblája:

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

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

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

A konjunktív normálforma

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

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

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

Példa:
$(\neg x \rightarrow y) \rightarrow (x \wedge \neg(\neg y \wedge z)) \sim_0 \neg(x \vee y) \vee (x \wedge (y \vee \neg z)) \sim_0$
$(\neg x \wedge \neg y) \vee (x \wedge (y \vee \neg z)) \sim_0 (\neg x \vee x) \wedge (\neg x \vee y \vee \neg z) \wedge$
$(\neg y \vee x) \wedge (\neg y \vee y \vee \neg z) \sim_0 (\neg x \vee y \vee \neg z) \wedge (\neg y \vee x)$.

Ez KNF. A DNF ebből egy disztributív szabályalkalmazással adódik:
$(\neg x \wedge \neg y) \vee (\neg x \wedge x) \vee (y \wedge \neg y) \vee (y \wedge x) \vee (\neg z \wedge \neg y) \vee (\neg z \wedge x) \sim_0$
$(\neg x \wedge \neg y) \vee (y \wedge x) \vee (\neg z \wedge \neg y) \vee (\neg z \wedge x)$.

KNF szerepe következmények bizonyításában

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

Ha $\mathcal{F} = \{F_1, \ldots, F_n\}$ véges formulahalmaz, akkor $\mathcal{F} \cup \{\neg\varphi\}$ kielégíthetetlensége ekvivalens $F_1 \wedge \cdots \wedge F_n \wedge \neg\varphi$ kielégíthetetlenségével.

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

Rezolvens

Rezolvens

Legyenek C_1 és C_2 pontosan 1 komplemens literálpárt tartalmazó klózok. Tehát $C_1 = C_1' \vee \ell_1$, $C_2 = C_2' \vee \ell_2$, ahol ℓ_1 és ℓ_2 komplemens literálpár, C_1' és C_2' viszont nem tartalmaz ilyet. A $\mathrm{res}(C_1, C_2) := C_1' \vee C_2'$ klózt (esetleges egyszerűsítés után) a (C_1, C_2) klózpár rezolvensének nevezzük. (Ha $C_1 = \ell_1$, $C_2 = \ell_2$, akkor $\mathrm{res}(C_1, C_2) = \Box$.)

Példa: Mi a rezolvensük?

klózpár	rezolvens
$(x \vee y, \neg y \vee z)$	$x \vee z$
$(x \vee y \vee z, \neg y \vee z)$	$x \vee z$
$(x \vee \neg y, \neg y \vee z)$	nincs: mindkét azonos alapú literál negált
$(x \vee \neg y, z \vee \neg v)$	nincs: nincs két azonos alapú literál
$(x \vee y \vee z, \neg y \vee \neg z)$	nincs: két komplemens literálpár van
$(x, \neg x)$	$\Box$

Rezolúció

Rezolúciós levezetés

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

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

és $K_m = C$.

Tétel

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

A bizonyítást nem részletezzük. Az egyik fontos lemma:

Lemma

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

Rezolúció

Példa: Rezolúciós levezetéssel igazoljuk, hogy az alábbi klózhalmaz kielégíthetetlen!
$\{y \vee z, \neg x \vee w \vee \neg z, \neg y, y \vee \neg z \vee \neg w, x \vee y\}$

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

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

Számításelmélet

4. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Elsőrendű logika

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

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

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

Elsőrendű logika – szintaxis

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

Definíció

Egy elsőrendű logika szimbólumhalmaza a következőkből áll

- Pred, a **predikátumszimbólumok** véges halmaza,
- Func, a **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 – szintaxis

Definíció

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

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

Definíció

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

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

Elsőrendű logika – szintaxis

Példa

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

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

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

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

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

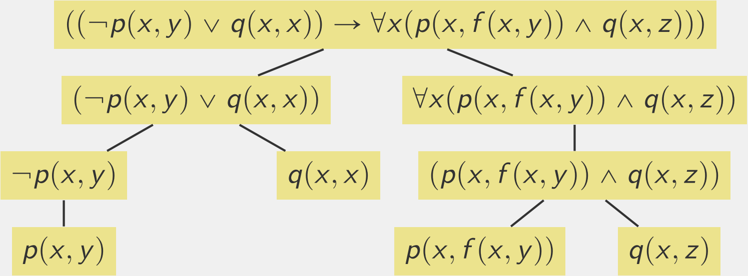
$p(x),\ \forall x f(x, y),\ p(x, q(y, z)) \notin$ Form

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

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

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

Szerkezeti fa, részformula, fő logikai összekötő



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

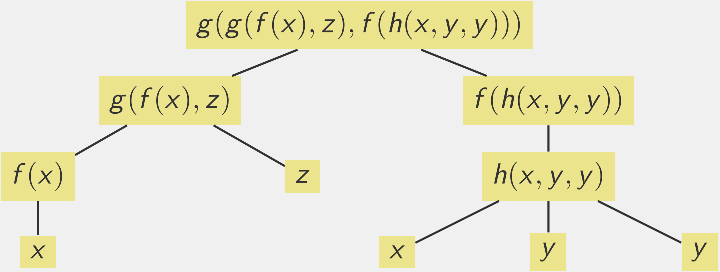
Az előforduló címkék a formula **részformulái**. (A példában a sárgával megjelölt formulák.) A levelek atomi formulák.

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

Term szerkezeti fája, zárójelelhagyás

Term szerkezeti fája:

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



Zárójelelhagyás:

Ugyanúgy, mint a nulladrendű logika esetén.

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

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

Mindent elhagyva, amit lehet:

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

Elsőrendű logika – interpretáció, változókiértékelés

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

Definíció

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

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

Definíció

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

Vegyük észre, hogy κ függ az U univerzumtól.

Elsőrendű logika – a termek szemantikája

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

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

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

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

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

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 $|c|^{I,\kappa} := c^I$,
- $|f(t_1, t_2, \ldots, t_{\text{ar}(f)})|^{I,\kappa} := f^I(|t_1|^{I,\kappa}, |t_2|^{I,\kappa}, \ldots, |t_{\text{ar}(f)}|^{I,\kappa})$.

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

Elsőrendű logika – a formulák szemantikája

Definíció

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

Definíció

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

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

Példa Az előző példát folytatva

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

$\varphi_1 = \forall x p(x,a)$,
Minden természetes szám $\geqslant 0$. $|\varphi_1|^{I,\kappa} = i$,

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

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

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

Elsőrendű logika – szabad és kötött előfordulás

Definíció

Legyen φ egy formula, és tekintsük $x \in \text{Ind}$ egy előfordulását φ-ben. (A kvantorokat közvetlenül követő változókat nem tekintjük ezen változó előfordulásának.) Azt mondjuk, hogy x ezen előfordulása **kötött**, ha x a φ egy $\exists x\psi$ vagy $\forall x\psi$ alakú részformulájába esik. Ellenkező esetben x ezen előfordulása **szabad**. Ha φ minden individuumváltozójának minden előfordulása kötött, akkor **zárt** formuláról beszélünk. Egyébként a 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 $\forall x p(x,x) \to q(x,x)$ nyitott, mert x 3. és 4. előfordulását nem tartalmazza kvantált részformula. (A formula részformulái: $\forall x p(x,x) \to q(x,x)$, $\forall x p(x,x)$, $p(x,x)$, $q(x,x)$.)

Az elsőrendű logika szemantikus alapfogalmai

Definíció

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

Elsőrendű logikai törvények

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

Elsőrendű logikai törvények

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

Megoldás:

$$|\neg\exists xA|^{I,\kappa} = h$$
$$\Updownarrow$$
$$|\exists xA|^{I,\kappa} = i$$
$$\Updownarrow$$
$$\kappa\text{-nak van olyan } \kappa^* \ x\text{-variánsa, amelyre } |A|^{I,\kappa^*} = i$$
$$\Updownarrow$$
$$\kappa\text{-nak van olyan } \kappa^* \ x\text{-variánsa, amelyre } |\neg A|^{I,\kappa^*} = h$$
$$\Updownarrow$$
$$|\forall x\neg A|^{I,\kappa} = h$$

Ugyanazon (I,κ) (interpretáció,változókiértékelés)-párokra hamis a forma, tehát valóban logikailag ekvivalensek.

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

Elsőrendű következmény

Példa: Igazoljuk formálisan a bevezetőben említett következtetést!

Megoldás:

Először is, a formalizáláshoz legyenek
$E(x)$: x ember
$H(x)$: x halandó
s : Szókrátész (konstans)

„Minden ember halandó.”	$\forall x(E(x) \to H(x))$
„Szókrátész ember.”	$E(s)$
„Szókrátész halandó.”	$H(s)$

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

Elsőrendű következmény

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

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

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

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

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

Függvények aszimptotikus nagyságrendje

Legyenek $f, g : \mathbb{N} \to \mathbb{R}_0^+$ 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 $f(n) \leqslant c \cdot g(n)$ minden $n \geqslant N$-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 $f(n) \geqslant c \cdot g(n)$ minden $n \geqslant N$-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 $c_1 \cdot g(n) \leqslant f(n) \leqslant c_2 \cdot g(n)$ minden $n \geqslant N$-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:

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

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

Számításelmélet

5. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Algoritmikus problémák

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

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

Példák:

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

Algoritmikus eldöntési problémák

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

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

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

Eldöntési problémák és matematikai gépek

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

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

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

Eldöntési problémák és matematikai gépek

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

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.

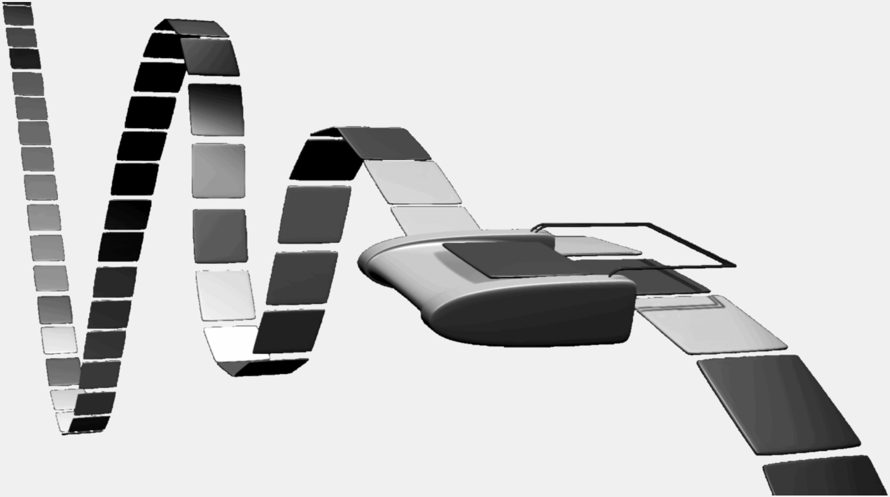
(illetve bármilyen, a Turing géppel azonos számítási teljesítményű absztrakt modellel)

NEM TÉTEL!!!

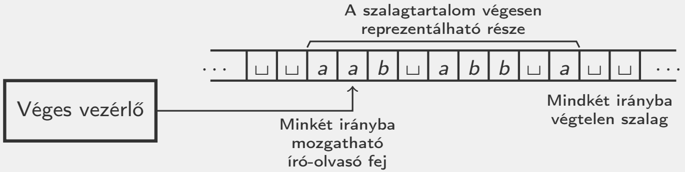
A Church-Turing tézis nem bizonyítható, hiszen nem egy formális matematikai állítás, az algoritmus intuitív fogalmát használja.

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

Turing gépek

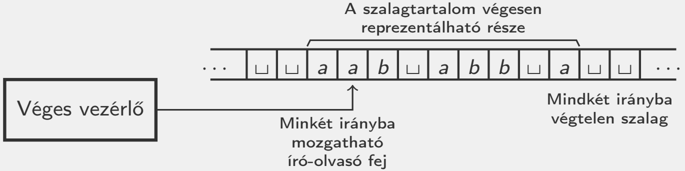


Turing gépek – Informális kép



- 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 $\sqcup \in \Gamma \backslash \Sigma$.
- $\delta : (Q \backslash \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény. δ az egész $(Q \backslash \{q_i, q_n\}) \times \Gamma$-n értelmezett függvény.

$\{L, S, R\}$ elemeire úgy gondolhatunk mint a TG lépéseinek irányai (balra, helyben marad, jobbra). (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 $q_0 u \sqcup$ szó. (Vagyis $q_0 u$, ha $u \neq \varepsilon$ és $q_0 \sqcup$, ha $u = \varepsilon$).

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

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

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

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

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

Egylépéses konfigurácóátmenet

Jelölés C_M: az M TG konfigurációinak halmaza.

Definíció

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

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

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

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

Többlépéses konfigurációátmenet

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

Definíció

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

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

Ekvivalens definíció:

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

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

A Turing gép által felismert nyelv; felismerhető/eldönthető nyelv

Az M TG által felismert nyelv

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

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

Definíció

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

Definíció

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

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

RE és R

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

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

Definíció

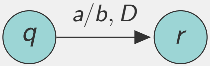
Legyen $f : \mathbb{N} \to \mathbb{N}$ egy függvény. Azt mondjuk, hogy M egy $\boldsymbol{f(n)}$ **időkorlátos** gép (vagy M $f(n)$ időigényű), ha minden $u \in \Sigma^*$ input szóra M futási ideje az u szón legfeljebb $f(|u|)$.

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

Turing gépek – Példa

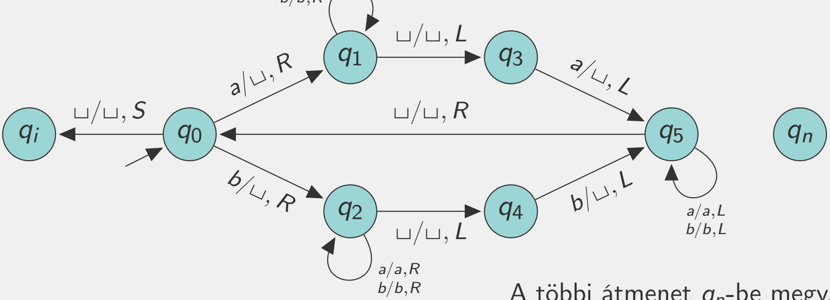
Feladat: Készítsünk egy M Turing gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Az **átmenetdiagram**.



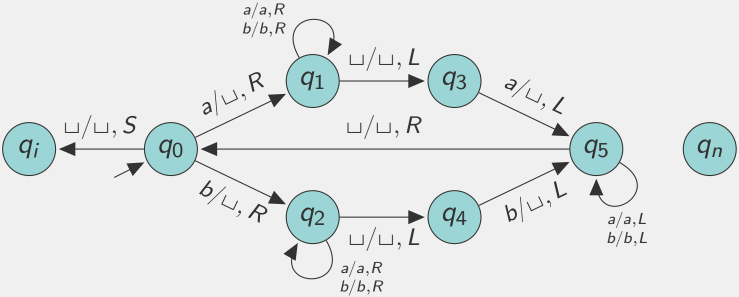
megfelel a $\delta(q, a) = (r, b, D)$ átmenetnek
$(q, r \in Q, a, b \in \Gamma, D \in \{L, S, R\})$

Egy **megoldás**:



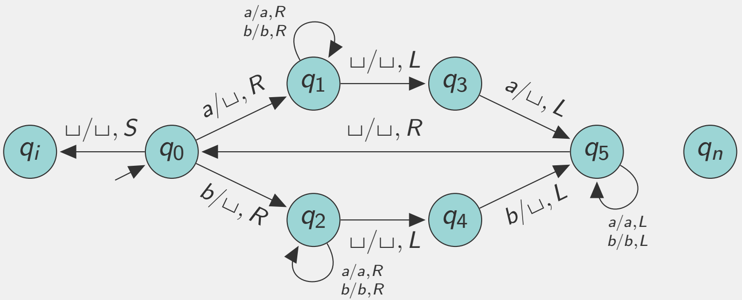
A többi átmenet q_n-be megy.

Turing gépek – Példa (folyt.)



Példa. Konfigurációátmenetek sorozata az *aba* inputra:
$q_0 aba \vdash q_1 ba \vdash b q_1 a \vdash ba q_1 \sqcup \vdash b q_3 a \vdash q_5 b \vdash q_5 \sqcup b \vdash q_0 b \vdash q_2 \sqcup \vdash q_4 \sqcup \vdash q_n \sqcup$.
Az *aba* inputra 10 lépésben jut a gép megállási konfigurációba.
Ebben a példában tetszőleges n-re ki tudjuk számolni a pontos időigényt is, de egyszerűbb (és gyakran elegendő) egy jó aszimptotikus felső korlát megadása.

Turing gépek – Példa (folyt.)



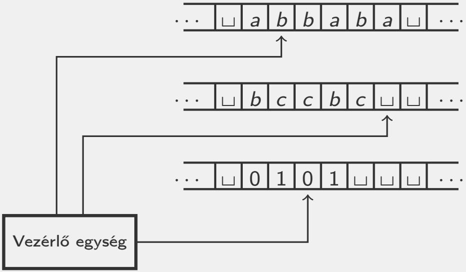
A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$-et lépünk, $+1$ lépés q_i-be vagy q_n-be.

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

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

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

Többszalagos Turing gép – Informális kép



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

k-szalagos Turing gép

Definíció

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

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

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

k-szalagos Turing gépek konfigurációi

Definíció

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

Ez azt reprezentálja, hogy

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

Definíció

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

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

k-szalagos Turing gépek megállási konfigurációi

Definíció

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

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

k-szalagos TG – egylépéses konfigurációátmenet

Definíció

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

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

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

k-szalagos TG – többlépéses konfigurációátmenet

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

Példa:

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

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

Definíció

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

k-szalagos TG – felismert nyelv és időigény

Definíció

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

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

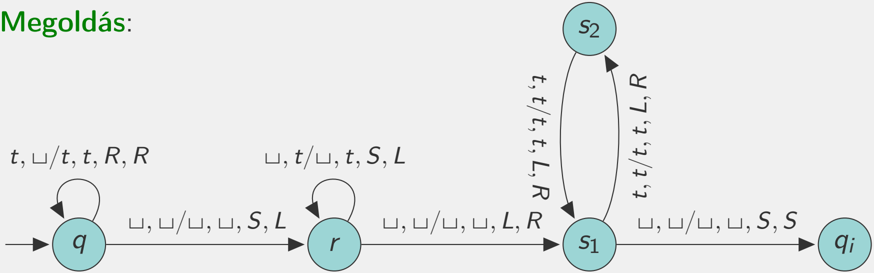
Definíció

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

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

k-szalagos Turing gép – 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

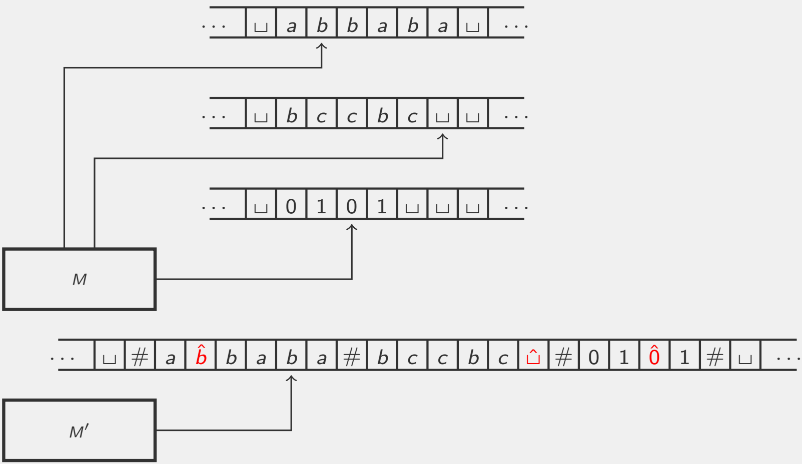
Két TG **ekvivalens**, ha ugyanazt a nyelvet ismerik fel.

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 $a_1 \cdots a_n$ bemeneteten:

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

k-szalagos TG szimulálása egyszalagossal

Meggondolható, hogy M egyetlen lépésének szimulálásakor

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

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

Turing gép egy irányban végtelen szalaggal

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

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

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

Turing gép egy irányban végtelen szalaggal

Tétel

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

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)

Számításelmélet

6. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

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

Azaz míg a **determinisztikus** esetben a δ átmenetfüggvény minden egyes $(Q \backslash \{q_i, q_n\}) \times \Gamma$-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 $\forall\, 1 \leqslant i \leqslant n-1$-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**
$L(M) = \left\{ u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon \text{ -ra} \right\}$.

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.

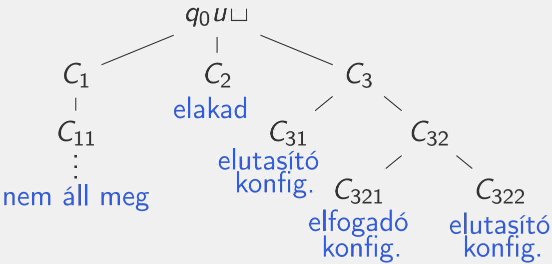
Egy NTG-nek azonban **több számítása is lehet ugyanarra a szóra**. Ezek között lehetnek elfogadó és elutasító (sőt nem termináló!) számítások is. Egy NTG akkor fogad el egy szót, ha az adott szóra **legalább egy számítása q_i-ben ér véget** (hiszen ekkor a kezdőkonfiguráció és ez az elfogadó konfiguráció $\vdash^*$ 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. $q_0 u \sqcup$ a gyökér címkéje. Ha C egy csúcs címkéje, akkor $|\{C' \mid C \vdash C'\}|$ gyereke van és ezek címkéi éppen $\{C' \mid C \vdash C'\}$ elemei.

Példa:



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

Nemdeterminisztikus számítási fa

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

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

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

NTG-vel való eldönthetőség, időigény

Definíció

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

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

Definíció

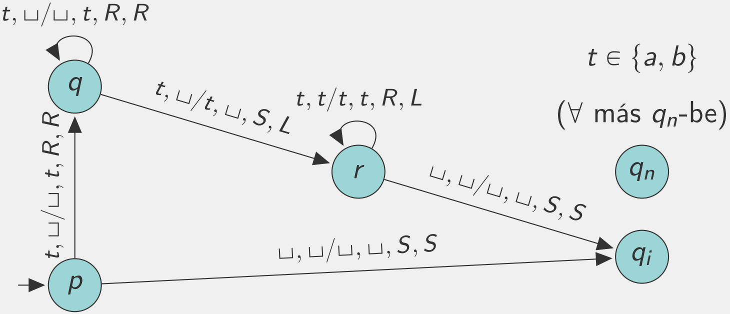
Az M NTG $\boldsymbol{f(n)}$ **időkorlátos** (időigényű), ha minden $u \in \Sigma^*$ n hosszú szóra u számítási fája legfeljebb $f(n)$ magas.

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

1. Példa: Ha $X = \{a, b\}$ és $a \prec b$, 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 $X = \{0 \prec 1 \prec 2 \prec 3 \prec 4 \prec 5 \prec 6 \prec 7 \prec 8 \prec 9\}$ rendezett ábécé feletti szavaknak tekintve őket $n \prec_{\text{shortlex}} m$ teljesül.

Nemdeterminisztikus Turing gép

Szimulálás determinisztikus TG-pel

Tétel

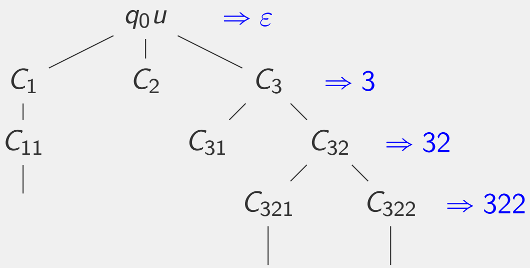
Minden $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ $f(n)$ időkorlátos NTG-hez megadható egy ekvivalens, $2^{O(f(n))}$ időkorlátos M' determinisztikus TG.

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

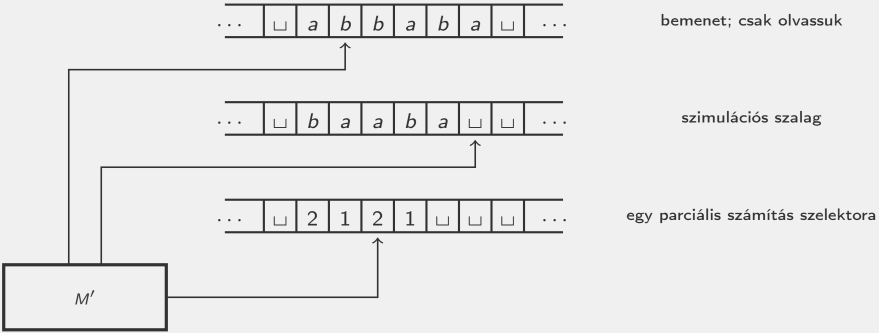
NTG szimulálása determinisztikus TG-pel

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 $\exists$ k-adik eleme, akkor
      - M' szimulálja M egy lépését ezen elem szerint
      - Ha ez q_i -be vezet, akkor M' is elfogad
      - Ha ez q_n-be vezet, akkor M' kilép ebből a ciklusból

      különben (ha $\delta(q, a)$-nak $\nexists$ k-adik eleme) kilép a ciklusból
    - M' a 3-ik szalagon eggyel jobbra lép
  - M' törli a 2. szalagot és előállítja a 3. szalagon a hossz-lexikografikus (shortlex) rendezés szerinti következő szót T felett (a fejet a szó elejére állítva)

NTG szimulálása determinisztikus TG-pel

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

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

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

Megjegyzés:

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

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

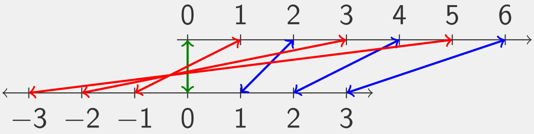
- A és B halmazoknak **megegyezik a számosságuk**, ha $\exists$ bijekció köztük. Jelölése: $|A| = |B|$.
- A-nak **legalább annyi a számossága**, mint B-nek, ha $\exists$ B-ből injekció A-ba. Jelölése: $|A| \geqslant |B|$.
- A-nak **nagyobb a számossága, mint B-nek**, ha $\exists$ B-ből A-ba injekció,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

Cantor-Bernstein-Schröder tétel

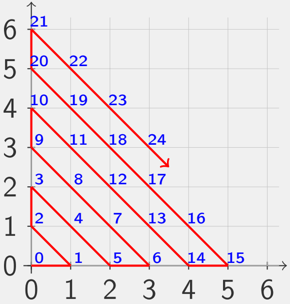
Ha $\exists$ injekció A-ból B-be és B-ből A-ba is, akkor $\exists$ bijekció A és B között, azaz ha $|A| \leqslant |B|$ és $|A| \geqslant |B|$, akkor $|A| = |B|$.

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

$\mathbb{Q}^+ := \{\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+,$ 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 $|\mathbb{Q}^+| \leqslant |\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$.
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: $\operatorname{tg}(\pi(x - \frac{1}{2}))\big|_{(0,1)} : (0,1) \to \mathbb{R}$ 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 $|[0,1)| \leqslant |\{0,1\}^{\mathbb{N}}|$.

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 $|\{0,1\}^{\mathbb{N}}| \leqslant |[0,1)|$.

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 $|\{0,1\}^{\mathbb{N}}| > |\mathbb{N}|$.

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

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

$H_0 \subset \{0,1\}^{\mathbb{N}}$, 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]

$|\mathcal{P}(H)| \geqslant |H|$, 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: $\quad \forall x \in H: \quad x :\in A \Leftrightarrow x \notin f^{-1}(x)$

$f(A) \in A$ 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

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

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

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

Létezik nem Turing-felismerhető nyelv

Tétel

Létezik nem Turing-felismerhető nyelv.

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

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

Jelölés: Minden $i \geqslant 1$-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 $T(i,j) = 1 \Leftrightarrow w_j \in L(M_i)$ $(i,j \geqslant 1)$.

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 $i \geqslant 1$-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.

Számításelmélet

7. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

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

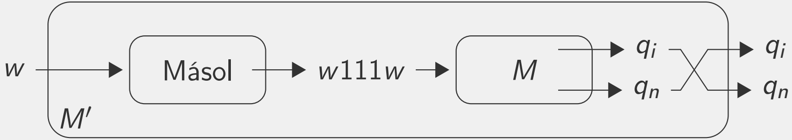
Eldönthetetlenség

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.

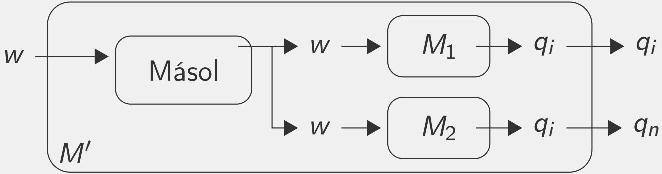
RE és R tulajdonságai

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

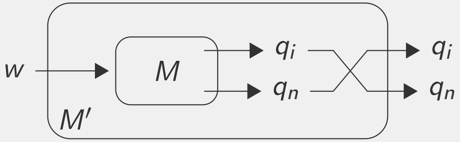
RE nem zárt a komplementer-képzésre.

Bizonyítás:
Legyen $L \in RE \backslash R$ (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.

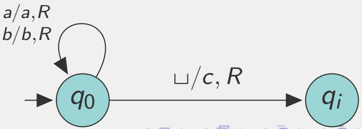
Definíció

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:

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



Visszavezetés

Definíció

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

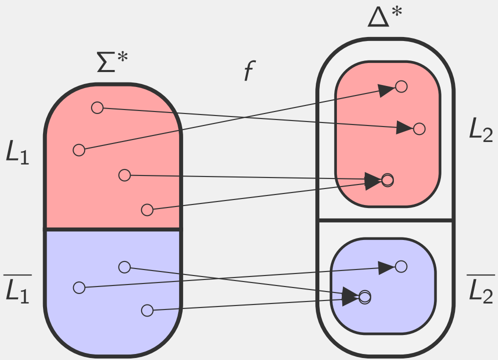
Definíció

$L_1 \subseteq \Sigma^*$ **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: $L_1 \leqslant L_2$

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

Visszavezetés

$L_1 \leqslant L_2$



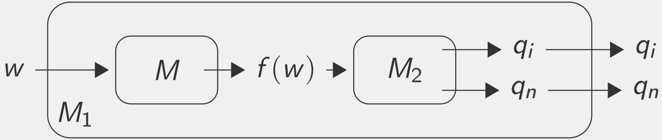
f kiszámítható, az egész Σ^*-on értelmezett, $f(L_1) \subseteq L_2$ valamint $f(\overline{L}_1) \subseteq \overline{L}_2$. f nem kell hogy injektív legyen és az se kell, hogy szürjektív.

Tétel

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

Visszavezetés

Bizonyítás:
Legyen $L_2 \in RE$ ($\in R$) és tegyük fel, hogy $L_1 \leqslant L_2$. 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 $L_1 \leqslant L_2$ és $L_1 \notin RE$, akkor $L_2 \notin RE$.
- Ha $L_1 \leqslant L_2$ 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?

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

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 $L_u \leqslant L_h$, 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 $L_h \leqslant L_u$, 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.

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

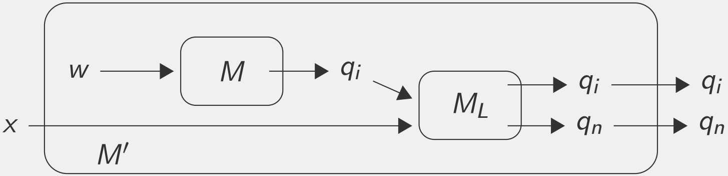
Bizonyítás:

1. eset $\emptyset \notin \mathcal{P}$.
Mivel tudjuk, hogy $L_u \notin R$, elég belátni, hogy $L_u \leqslant L_{\mathcal{P}}$.
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 $L_u \leqslant L_{\mathcal{P}}$ és így $L_{\mathcal{P}} \notin R$.

Rice tétel

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

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

Rice tétel

Alkalmazások

Következmények:

Eldönthetetlen, hogy egy M TG

- az üres nyelvet 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 $u_1, \ldots, u_n, v_1 \ldots, v_n \in \Sigma^+$ $(n \geqslant 1)$. 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 $(m \geqslant 1, 1 \leqslant i_1, \ldots, i_m \leqslant n)$ 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 keresésekor egy végtelen keresési térrel van dolgunk.

Post Megfelelkezési Probléma (PMP):

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

Post Megfelelkezési Probléma

Tétel

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

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

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

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.

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

Post Megfelelkezési Probléma

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

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

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 $d_i = \frac{u_i}{v_i}$ $(1 \leqslant i \leqslant n)$.
D' legyen a következő $|D| + 2$ méretű készlet $(\# \notin \Sigma)$:

$$d_i' = \frac{\text{balcsillag}(u_i)}{\text{jobbcsillag}(v_i)} \qquad (1 \leqslant i \leqslant n)$$

$$d_0' = \frac{\text{balcsillag}(u_1)}{\text{baljobbcsillag}(v_1)}, \qquad d_{n+1}' = \frac{*\#}{\#}$$

Post Megfelelkezési Probléma

Példa: Ha

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

akkor

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

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

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 $L_{\text{MPMP}} \leqslant L_{\text{PMP}}$.

Post Megfelelkezési Probléma

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

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

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

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

- $d := \frac{\#}{\#q_0 a_1 \cdots a_n \#}$ (ahol $\# \notin \Sigma$) $\quad d :\in D$
- – 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

$$\frac{\#}{\#q_0bab\#} \;\Bigg|\; \frac{q_0b}{aq_2}\,\frac{a}{a}\,\frac{b}{b}\,\frac{\#}{\#} \;\Bigg|\; \frac{a}{a}\,\frac{q_2a}{q_ib}\,\frac{b}{b}\,\frac{\#}{\#} \;\Bigg|\; \frac{aq_i}{q_i}\,\frac{b}{b}\,\frac{b}{b}\,\frac{\#}{\#} \;\Bigg|\; \frac{q_ib}{q_i}\,\frac{b}{b}\,\frac{\#}{\#} \;\Bigg|\; \frac{q_ib}{q_i}\,\frac{\#}{\#} \;\Bigg|\; \frac{q_i\#\#}{\#}$$

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

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

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

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

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

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

Post Megfelelkezési Probléma

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

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

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

Post Megfelelkezési Probléma

Állítás: A visszavezetés tranzitív.

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

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

Innen a tétel bizonyítása: $L_u \leqslant L_{\mathrm{MPMP}}$, $L_{\mathrm{MPMP}} \leqslant L_{\mathrm{PMP}}$ 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.)

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

Tétel

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

Bizonyítás: Megmutatjuk, hogy $L_{\mathrm{PMP}} \leqslant \overline{L_{\mathrm{ECF}}}$.
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 $L_{\mathrm{PMP}} \notin R$, következik, hogy $\overline{L_{\mathrm{ECF}}} \notin R$, amiből kapjuk, hogy $L_{\mathrm{ECF}} \notin R$. $\square$

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

Lemma

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

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

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

Ötlet: Amíg Σ-beli betűk jönnek az inputon pakoljuk őket bele a verembe. Ha $a_i \in \Delta$-beli betű jön, akkor próbáljuk meg kivenni u_i^{-1}-et a veremből. Megvalósítás:
$A = \langle \Sigma \cup \{\#\}, Q, \Sigma \cup \Delta, \delta, q_0, \#, \{s\} \rangle$, ahol

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

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

és M_δ:

$\#q_0 t \to \#tq_0 \quad (t \in \Sigma)$

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

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

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

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

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

$\#r \to \#s$

Az veremautomata $L(G_A)$-t ismeri fel és determinisztikus (teljessé is tehető egy zsákutcaállapot és a hiányzó átmenetek hozzávételével.)

Környezetfüggetlen grammatikákkal kapcsolatos eldönthetetlen algoritmikus problémák

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

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

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

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

$\square$

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_{PMP}-t vezethetjük vissza rá, ehhez tekintsük ismét G_A-t és G_B-t.

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

Legyenek G_1 és G_2 olyan környezetfüggetlen grammatikák, amelyekre $L(G_1) = L$ és $L(G_2) = (\Sigma \cup \Delta)^*$.
$L(G_1) = L(G_2) \Leftrightarrow L(G_A) \cap L(G_B) = \emptyset$, így ha (2) eldönthető volna, akkor az (1)-beli érvelés alapján L_{PMP} is az lenne, de láttuk, hogy nem az.

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

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

Számításelmélet

8. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

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 ítélettáblá(ka)t a szóban forgó formulá(k)ra és olvassuk le belőlük. $\square$

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

Eldönthetetlen problémák az elsőrendű logikában

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

Definíció

$\textsc{ValidityPred} := \{\langle\varphi\rangle \mid \varphi$ logikailag igaz elsőrendű formula$\}$.
$\textsc{UnsatPred} := \{\langle\varphi\rangle \mid \varphi$ kielégíthetetlen elsőrendű formula$\}$.
$\textsc{SatPred} := \{\langle\varphi\rangle \mid \varphi$ kielégíthető elsőrendű formula$\}$.
$\textsc{EqivPred} := \{\langle\varphi,\psi\rangle \mid \varphi,\psi$ elsőrendű formulák, melyekre $\varphi \sim \psi\}$.
$\textsc{ConsPred} := \{\langle\mathcal{F},\varphi\rangle \mid \mathcal{F}$ véges elsőrendű formulahalmaz, φ elsőrendű formula, $\mathcal{F} \models \varphi\}$.

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

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

Eldönthetetlen problémák az elsőrendű logikában

Tétel

$\textsc{ValidityPred} \notin \mathrm{R}$

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

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

Tekintsük azt az elsőrendű logikai nyelvet ahol $\mathrm{Pred} = \{p\}, \mathrm{ar}(p) = 2, \mathrm{Func} = \{f_{a_1}, \ldots, f_{a_n}\}$, $\mathrm{ar}(f_{a_i}) = 1$ $(\forall\, 1 \leqslant i \leqslant n), \mathrm{Cnst} = \{c\}$.

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

Eldönthetetlen problémák az elsőrendű logikában

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

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

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

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

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

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

$$\begin{aligned} &\text{van olyan } m \geqslant 1 \text{ és } 1 \leqslant i_1, \ldots i_m \leqslant k, \text{ hogy} \\ &u_{i_1} \cdots u_{i_m} = u \text{ és } v_{i_1} \cdots v_{i_m} = v. \qquad (*) \end{aligned}$$

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

Eldönthetetlen problémák az elsőrendű logikában

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

Tegyük fel, hogy $p^I(u, v) =$ igaz valamely $u, v \in \Sigma^*$-ra. Ekkor p^I definíciója szerint van olyan $m \geqslant 1$ és $1 \leqslant i_1, \dots i_m \leqslant k$, hogy $u_{i_1} \cdots u_{i_m} = u$ és $v_{i_1} \cdots v_{i_m} = v$.
Ekkor nyilván minden $1 \leqslant i \leqslant k$-ra az $u_i u$ és $v_i v$ szavakra is teljesül a (*) feltétel (az $i, i_1, \dots i_m$ index sorozat jó), tehát $p^I(u_i u, v_i v) =$ igaz.
Mivel $|f_{u_i}(u)|^I = u_i u$ és $|f_{v_i}(v)|^I = v_i v$, ezért bármely $u, v \in \Sigma^*$-ra ha $p^I(u, v) =$ igaz, akkor
$p^I(|f_{u_1}(u)|^I, |f_{v_1}(v)|^I) \wedge \cdots \wedge p^I(|f_{u_k}(u)|^I, |f_{v_k}(v)|^I) =$ igaz. Tehát $I \models \varphi_2$.

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

Eldönthetetlen problémák az elsőrendű logikában

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

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

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

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

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

Eldönthetetlen problémák az elsőrendű logikában

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

Tehát $I \vDash \varphi_3$ és így $I \vDash \varphi_D$.
D alapján a φ_D formula nyilván kiszámítható, így $L_{\text{PMP}} \leqslant$ ValidityPred. A tétel állítása következik egy a visszavezetésről tanult tételből és abból, hogy $L_{\text{PMP}} \notin \mathrm{R}$. $\square$.

Következmény

UnsatPred, SatPred, EquivPred, ConsPred $\notin \mathrm{R}$

Bizonyítás: φ kielégíthetetlen $\Leftrightarrow\ \vDash \neg\varphi$. Tehát az előző tétel alapján UnsatPred $\notin \mathrm{R}$.

Eldönthetetlen nyelv komplementere eldönthetetlen, tehát SatPred $\notin \mathrm{R}$.

φ kielégíthetetlen $\Longleftrightarrow \varphi \sim \bot$, így EquivPred $\notin \mathrm{R}$.

$\mathcal{F} \vDash \varphi \Longleftrightarrow \mathcal{F} \cup \{\neg\varphi\}$ kielégíthetetlen, így ConsPred $\notin \mathrm{R}$. $\square$

Eldönthetetlen problémák az elsőrendű logikában

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

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

Tétel

$\textsc{UnsatPred} \in \mathrm{RE}$.

Következmény

$\textsc{SatPred} \notin \mathrm{RE}$

Bizonyítás: Korábbi tételünk volt hogy $L, \overline{L} \in \mathrm{RE} \Rightarrow L \in \mathrm{R}$. Mivel $\overline{\textsc{UnsatPred}} = \textsc{SatPred}$ és $\textsc{UnsatPred} \in \mathrm{RE} \setminus \mathrm{R}$, ezért $\textsc{SatPred} \notin \mathrm{RE}$. $\square$

$\mathcal{L}_0$ és RE kapcsolata

Tétel

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

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

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

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

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

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

$\mathcal{L}_0$ és RE kapcsolata

Tétel

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

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

$\mathcal{L}_0$ és RE kapcsolata

Legyen $G = \langle (\Gamma \backslash \Sigma) \cup Q \cup \{S, A, \rhd, \lhd\}, \Sigma, P, S \rangle$.
P szabályai:

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

1-2. generálunk egy tetszőleges elfogadó konfigurációt

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

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

$\mathcal{L}_0$ és RE kapcsolata

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

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

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

Tehát $q_0 w \vdash^* \alpha q_i \beta$ valamely $\alpha, \beta \in \Gamma^*$-ra akkor és csak akkor, ha $S \Rightarrow^* \rhd \alpha q_i \beta \lhd \Rightarrow^* \rhd \sqcup^{i} q_0 w \sqcup^{j} \lhd \Rightarrow^* w$. $\square$

$\mathcal{L}_1$ és R kapcsolata

Definíció

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

- a bemenetek $\rhd(\Sigma\backslash\{\rhd,\lhd\})^*\lhd$-beliek,
- $\rhd$ és $\lhd$ nem írhatók felül
- $\rhd$-tól balra illetve $\lhd$-től jobbra nem állhat a fej.
- a fej kezdőpozíciója a $\rhd$ tartalmú cella jobb-szomszédja

Magyarán az LKA egy korlátos munkaterülettel rendelkező NTG.

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

$\mathcal{L}_1$ és R kapcsolata

Tétel

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

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

Bizonyítás (vázlat):

(1) Az előző előtti tételben láttuk, hogy minden 0. típusú grammatikához lehet konstruálni $L(G)$-t felismerő NTG-t.
A konstrukció a 3. szalagján nemdeterminisztikusan szimulált egy G-beli levezetést, az iterációk végén ellenőrizte, hogy a 3. szalagon lévő mondatforma megegyezik-e az első szalagon lévő u inputtal.

Amennyiben G 1-es típusú, azaz hossz-nemcsökkentőek a szabályai, akkor a 3. szalagon lévő mondatforma hossza nem haladhatja meg $|u|$-t, így ez az NTG egy LKA.

$\mathcal{L}_1$ és R kapcsolata

(2) Ha az előző bizonyításban konstruált grammatika szabályai hossz-nemcsökkentőek lennének, akkor ez megfelelő is lenne.

Azonban a hosszcsökkentő szabályok kiküszöbölhetőek:

- az elfogadó konfiguráció generálásakor használtuk az $A \to \varepsilon$ szabályt. Mivel itt valójában S-ből egy hosszabb szót építünk fel, ennek a szabálynak az alkalmazása könnyen megkerülhető (az egyszerűség miatt szerepelt),
- $\rhd$, $\lhd$, q_0-t eltüntető szabályok. Mindegyiket csak egyszer használjuk levezetésenként. $(N \cup T) \times (N \cup T)$-beli jeleknek a nemterminálisokhoz való hozzáadásával ezek használata elkerülhető (példa: az $AB \to C$ hosszcsökkentő szabályt $(A, B) \to C$-vel helyettesíthetjük),
- a lineáris korlátoltság miatt M konfigurációi egy adott u inputra nem lehetnek hosszabbak, mint az u-hoz tartozó kezdőkonfiguráció hossza, így nincs szükség $\sqcup$-eket eltüntető szabályokra. $\square$

$\mathcal{L}_1$ és R kapcsolata

Tétel

Ha A LKA, akkor $L(A)$ eldönthető.

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

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

Következmény

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

$\mathcal{L}_1$ és R kapcsolata

Tétel

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

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

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

A Cantor-féle átlós módszerrel megmutatható, hogy $L_{\text{LKA-átló}} \in \mathrm{R} \setminus \mathcal{L}_1$. Ezt nem bizonyítjuk.

Tehát az algoritmikus és a Chomsky nyelvosztályok kapcsolata így foglalható össze:

Következmény

$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathrm{R} \subset \mathcal{L}_0 = \mathrm{RE}$.

R,RE és a Chomsky nyelvosztályok – Összefoglaló

Az alábbi táblázatban adott sorban minden nyelvleíró eszköz egyforma erejű és erősebb a korábbi sorokban felsoroltaknál.

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

Számításelmélet

9. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

BONYOLULTSÁGELMÉLET

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

A problémakat a legtakarékosabb megoldásuk erőforrásigénye alapján osztályozhatjuk.

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

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

Definíció

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

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

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

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 „megsejt" (azaz nemdeterminisztikusan generál) egy T egy bizonyítékot (vagy „tanút"), majd polinom időben leellenőrzi (már determinisztikusan), hogy T alapján $I \in L$ teljesül-e. A NTG definíciója alapján elég ha a számításai közül egy produkál egy ilyen „tanút".

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

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

Polinom idejű visszavezetés

Definíció

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

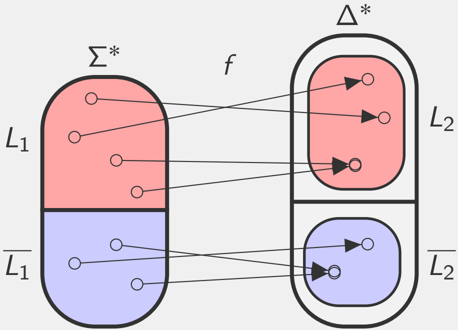
Definíció

$L_1 \subseteq \Sigma^*$ **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: $L_1 \leqslant_p L_2$.

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

$L_1 \leqslant_p L_2$



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 $L_1 \leqslant_p L_2$ és $L_2 \in \mathrm{P}$, akkor $L_1 \in \mathrm{P}$.
- Ha $L_1 \leqslant_p L_2$ és $L_2 \in \mathrm{NP}$, akkor $L_1 \in \mathrm{NP}$.

Polinom idejű visszavezetés

Bizonyítás:

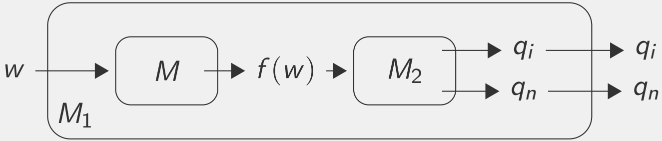
Az elsőt bizonyítjuk, a második analóg.

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

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

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

Konstruáljuk meg M_1-et:



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

$\mathcal{C}$-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 $L' \leqslant_p L$.

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 $L' \leqslant_p L$.

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 $L' \leqslant_p L$, 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. $\square$

Egy NP-teljes nyelv

Ha $L \leqslant_p L'$, 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

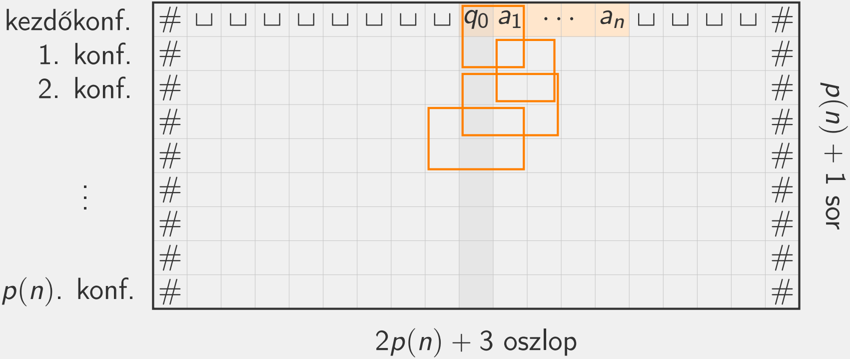
Bizonyítás:

- 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, $L \leqslant_p$ 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 $p(n) \geqslant n$.)
  • 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, $\leqslant p(n)$ 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:

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

– 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:

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

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

b_1	b_2	b_3
b_4	b_5	b_6

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

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

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 $k = |\Delta|$.

$\varphi_0 : \quad (p(n)+1)(2p(n)+3)(k+k(k-1)) = O(p^2(n))$,

$\varphi_{\text{start}} : \quad 2p(n)+3 = O(p(n))$,

$\varphi_{\text{move}} : \quad \leqslant p(n)(2p(n)+1)k^6 \cdot 6 = O(p^2(n))$,

$\varphi_{\text{accept}} : \quad 2p(n)+1 = O(p(n))$,

azaz φ_w $O(p^2(n))$ méretű, így polinom időben megkonstruálható

• tehát $w \mapsto \langle\varphi_w\rangle$ pol. idejű visszavezetés, így $L \leqslant_p$ SAT.

• Ez tetszőleges $L \in$ NP nyelvre elmondható. Így SAT NP-nehéz. Mivel NP-beli, ezért NP-teljes is. $\square$

Polinom idejű visszavezetés tranzitivitása

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

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.

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

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

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

További NP-teljes problémák

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

Tétel

Ha L NP-teljes, $L \leqslant_p L'$ és $L' \in$ NP, akkor L' NP-teljes.

Bizonyítás:

Legyen $L'' \in$ NP tetszőleges. Mivel L NP-teljes, ezért $L'' \leqslant_p L$. Mivel a feltételek szerint $L \leqslant_p L'$, ezért a polinom idejű visszavezetések tranzitivitása miatt L' NP-nehéz. Ebből és a 3. feltételből kövezkezik az állítás. $\square$

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

$\varphi \mapsto \varphi'$:

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

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

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

3SAT NP-teljessége

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

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

$n = 1$: ($\Rightarrow$): 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.

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

$n \geqslant 5$: ($\Leftarrow$): 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$\leqslant_p$ 3SAT. $\square$

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 $\ell_i \vee \ell_j \sim_0 \neg\ell_i \to \ell_j \sim_0 \neg\ell_j \to \ell_i$.

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. $\square$

HORNSAT P-beli

Tétel

HORNSAT $\in$ P.

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

1. $x_k \quad (1 \leqslant k \leqslant n)$,
2. $x_k \vee \neg x_{i_1} \vee \cdots \vee \neg x_{i_j} \quad (j \geqslant 1, 1 \leqslant k, i_1 \ldots, i_j \leqslant n)$,
3. $\neg x_{i_1} \vee \cdots \vee \neg x_{i_j} \quad (j \geqslant 1, 1 \leqslant i_1 \ldots, i_j \leqslant n)$.

Definiálja az $I_{\min}$ interpretációt a következő algoritmus:

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

HORNSAT P-beli

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

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

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

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

Számításelmélet

10. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

3 színezhetőség

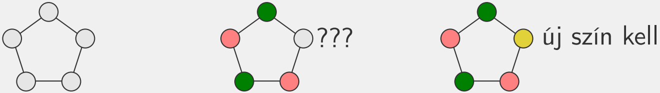
Definíció

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-színezhető$\}$

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

Tétel

3SZÍNEZÉS NP-teljes.

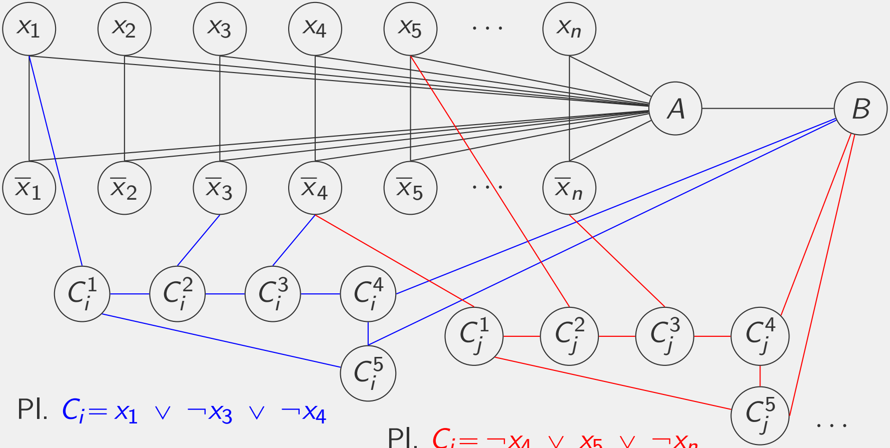
- kSZÍNEZÉS NP-beli: egy $\langle G \rangle$ inputra az M NTG egy számítási ága állítson elő egy $f : V(G) \to \{1, \ldots, k\}$ k-színezést. Mind egy konkrét k-színezés előállítása, mind pedig a konkrét színezés helyességének ellenőrzése polinom időben megtehető. M egy számítása végződjön q_i-ben, ha 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 $\Leftrightarrow G_\varphi$ 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:



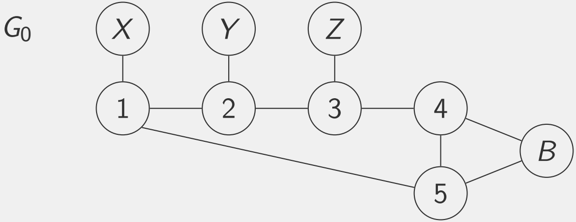
Pl. $C_i = x_1 \vee \neg x_3 \vee \neg x_4$

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.



A lemma bizonyítása:

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

3 színezhetőség

- Ha X, Y, Z, B nem egyszínű, akkor megadható egy színezés.

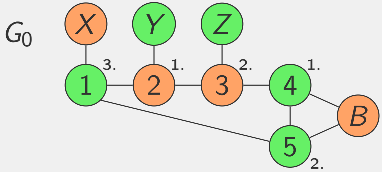
  1. lépés: első körben 2 színt használunk, 1,2,3,4,5-öt színezzük az $\{X, Y, Z, B\}$-beli szomszédjával ellentétes színűre.

  Ez persze még nem jó, lehetnek azonos színű szomszédok.

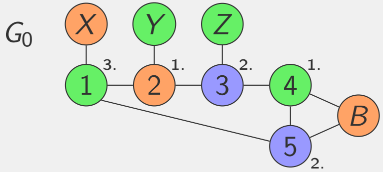
  2. lépés: bevetjük a 3. színt: ha 1,2,3,4,5 között van valahány egymás utáni azonos színű csúcs (az óramutató járása szerint és ciklikusan), akkor ezen egymás utáni azonos színű csúcsok közül minden párosadikat színezzük át a 3. színre.

Példa:

1. lépés utáni színezés



2. lépés utáni színezés



3 színezhetőség

A visszavezetés bizonyítása:

- Tegyük fel hogy φ kielégíthető, ekkor meg kell adnunk G_φ egy 3-színezését. Legyenek a színek piros, zöld és kék. Ha x_i igaz, akkor legyen az x_i csúcs zöld, az $\overline{x}_i$ 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 $x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n$ 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 φ ismeretében G_φ mérete és elkészítésének ideje az input (φ) méretének polinomja, ezért a visszavezetés polinom idejű.

Mivel $\mathrm{3SAT}$ NP-teljes, korábbi tételünk miatt $\mathrm{3S\small{ZÍNEZÉS}}$ is az. $\square$

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

$\mathrm{2S\small{ZÍNEZÉS}} \in \mathrm{P}$

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. $\square$

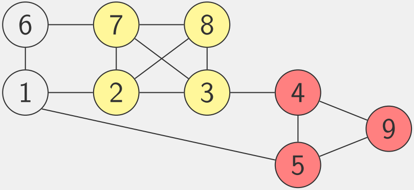
Klikk

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

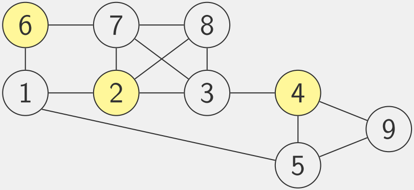
Definíció

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

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

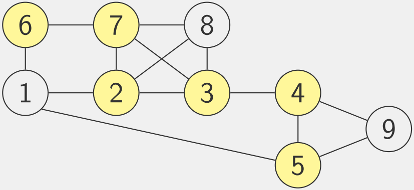
Legyen $S \subseteq V(G)$ és $E \in E(G)$. Ha $S \cap E \neq \emptyset$, akkor a csúcshalmaz **lefogja** E-t. Ha S minden $E \in E(G)$ élt lefog, akkor S egy **lefogó ponthalmaz**.

Megjegyzés: A fenti fogalom csúcsfedés néven is ismeretes.

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.

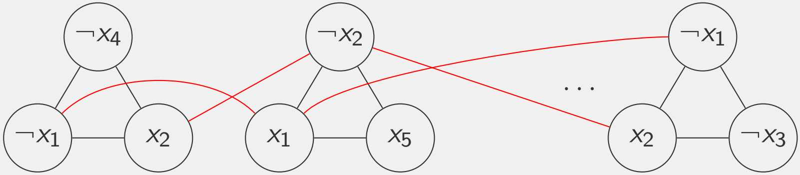
- Egy NTG egy számítási ágán vizsgálja meg a csúcsoknak egy konkrét, k elemű részhalmazát. Egy k elemű ponthalmaz előállítása illetve annak ellenőrzése, hogy ez egy klikk/független ponthalmaz/lefogó ponthalmaz az input méretének polinomiális függvénye. Tehát mindhárom nyelv NP-ben van.
- 3SAT $\leqslant_p$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



$\neg x_1 \vee x_2 \vee \neg x_4$ $\quad x_1 \vee \neg x_2 \vee x_5$ $\quad \neg x_1 \vee x_2 \vee \neg x_3$

$*$ 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.

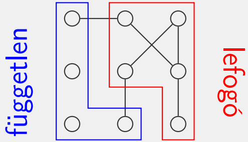
KLIKK, LEFOGÓ PONTHALMAZ

- 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

Definíció

$\mathcal{S}$ 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.

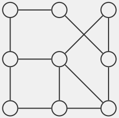
Lefogó ponthalmaz a Hipergráf lefogó ponthalmaz speciális esete. Minden gráf hipergráf is egyben, a megfeleltetés $U = V(G)$, $\mathcal{S} = E(G)$. k ugyanaz, mivel a lefogó ponthalmaz szintén speciális esete a hipergráf lefogó ponthalmaznak. $\square$

Irányítatlan/irányított Hamilton út/kör

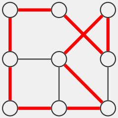
Definíció

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.

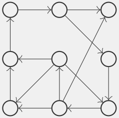
Rövidítés: H-út/ H-kör: Hamilton út/ Hamilton kör.



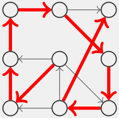
G



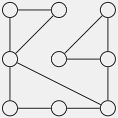
H-kör G-ben



G'



H-út G'-ben



nincs H-kör

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

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

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

Irányított $s \rightsquigarrow t$ Hamilton út NP teljessége

Tétel

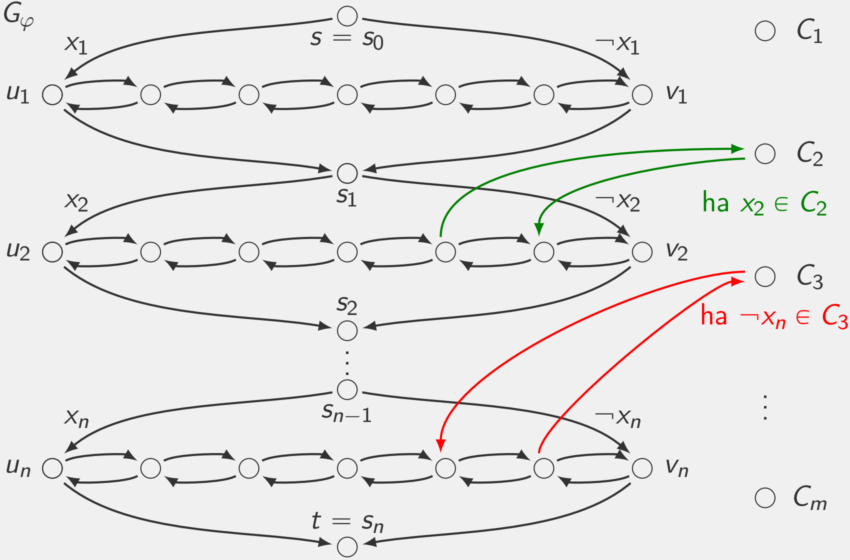
HÚ NP-teljes

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 $3m - 1$ belső pontú kétirányú út $w_{i,1}, \ldots, w_{i,3m-1}$.
- Minden $w_{i,k}$ legfeljebb egy C_j-vel lehet összekötve, ha $3|k$, akkor eggyel se.
- Ha $x_i \in C_j$, akkor $(w_{i,3j-2}, C_j)$ és $(C_j, w_{i,3j-1}) :\in E(G_\varphi)$. (pozitív bekötés)
- Ha $\neg x_i \in C_j$, akkor $(w_{i,3j-1}, C_j)$ és $(C_j, w_{i,3j-2}) :\in E(G_\varphi)$. (negatív bekötés)

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

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. $\square$

Irányítatlan $s \rightsquigarrow t$ Hamilton út NP teljessége

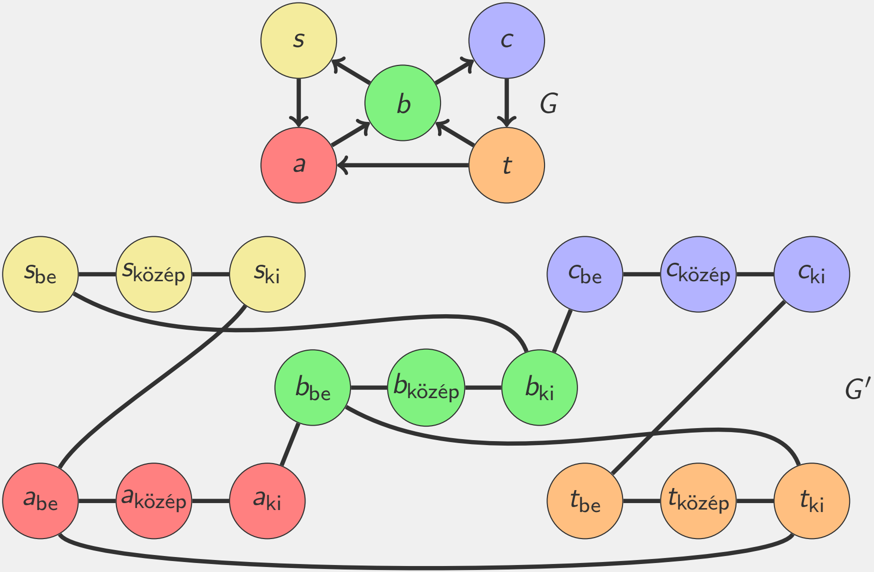
Megjegyzés: IHÚ és IHK NP-belisége az előzőekhez hasonlóan adódik.

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 $s \rightsquigarrow t$ Hamilton út NP teljessége



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

Könnyen meggondolható, hogy ez egy polinomiális visszavezetés:

- G' mérete G méretének polinomja és G' G-ből nyilván polinom időben előállítható.
- ha G-ben van egy $P : s \rightsquigarrow t$ irányított H-út, akkor G' konstrukciója miatt a következő G'-beli csúcssorozat H-út G'-ben: P szerint haladva minden v csúcsot sorra a v_{be}, $v_{\text{közép}}$, v_{ki} csúcsokkal helyettesítsük.
- ha G'-ben van egy $P' : s' \rightsquigarrow t'$ H-út, akkor P'-ben minden v-re v_{be}, $v_{\text{közép}}$, 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. $\square$

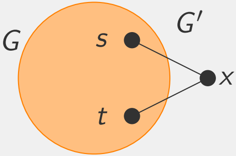
Irányítatlan Hamilton kör NP teljessége

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. $\square$

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. $\square$

Számításelmélet

11. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

Lineáris diophantoszi egyenlőtlenségrendszer

DIOPHANTOSZI EGYENLŐTLENSÉGRENDSZER$=$

$\{\langle \mathbf{A}, \mathbf{b} \rangle \mid \mathbf{A}\mathbf{x} \leqslant \mathbf{b}$ egészegyütthatós egyenlőtlenségrendszernek van egész megoldása$\}$.

Tétel

DIOPHANTOSZI EGYENLŐTLENSÉGRENDSZER NP-nehéz.

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

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

Lineáris diophantoszi egyenlőtlenségrendszer

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

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

Részletösszeg probléma

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

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

Tétel

RÉSZLETÖSSZEG NP-teljes.

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

Megmutatjuk, hogy 3SAT$\leqslant_p$RÉSZLETÖSSZEG.
Legyen φ 3KNF n változóval és m klózzal.

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

Részletösszeg probléma

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

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

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

A számok:

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

Részletösszeg probléma

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

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

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

Részletösszeg probléma

Fordítva, tegyük fel, hogy az S' részhalmazbeli számok összege K. Ez csak úgy lehetséges, ha az első $2n$ számból pontosan n-et választottunk, minden változóra 1-et. Ha az i. változóhoz rendelt 2 szám közül az első S'-beli, akkor legyen x_i igaz, különben hamis. S' minden $1 \leqslant j \leqslant m$-re pontosan 1-et tartalmaz a j. klózhoz rendelt 3 számból, hiszen ha egyet se tartalmazna, akkor legfeljebb 3, ha legalább 2-t akkor legalább 11 lenne a j. számjegy. Ez azt jelenti, hogy 8-at csak úgy kaphatunk, hogy 1,2 vagy 3 darab 1-es szerepel az összegben. Ez viszont azt jelenti, hogy minden klózban 1,2 vagy 3 literál igaz, tehát φ kielégíthető.

Mivel S $O(n+m)$ darab $O(n+m)$ jegyű számból áll és K is $O(n+m)$ jegyű ezért a $\varphi \mapsto (S, K)$ függvény polinom időben kiszámítható, tehát 3SAT polinom időben visszavezethető Részletösszeg-re, így Részletösszeg NP-teljes. $\square$

Hátizsák probléma

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

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

Tétel

HÁTIZSÁK NP-teljes.

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

Hátizsák probléma

$\textsc{Részletösszeg} \leqslant_p \textsc{Hátizsák}$:
Legyen (S, K) $\textsc{Részletösszeg}$ egy bemenete, ahol $S = \{s_1, \ldots, s_n\}$.

$a_i := s_i, p_i := s_i, b := K, k := K$ minden $1 \leqslant i \leqslant n$-re.

Ha valamely $I \subseteq \{1, \ldots n\}$-re $\sum_{i \in I} s_i = K$, akkor $\sum_{i \in I} a_i = K = b \leqslant b$ és $\sum_{i \in I} p_i = K = k \geqslant k$.

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

(S, K)-ból $a_1, \ldots, a_n, b, p_1, \ldots p_n, k$ polinom időben kiszámítható, tehát $\textsc{Részletösszeg} \leqslant_p \textsc{Hátizsák}$, így $\textsc{Hátizsák}$ NP-nehéz, és NP-belisége miatt NP-teljes is. $\square$

Partíció probléma

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

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

Tétel

PARTÍCIÓ NP-teljes.

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

RÉSZLETÖSSZEG $\leqslant_p$ PARTÍCIÓ. Legyen $S = \{s_1, \ldots, s_m\}$ és b a RÉSZLETÖSSZEG egy bemenete, feltehető, hogy $b \leqslant s = \sum_{i=1}^{m} s_i$.
$B := \{s_1, \ldots, s_m, s+1-b, b+1\}$. Ekkor
$\langle B\rangle \in$ PARTÍCIÓ $\Longleftrightarrow \langle S, b\rangle \in$ RÉSZLETÖSSZEG.

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

A visszavezetés nyilván polinomiális. $\square$

Ládapakolás

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

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

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

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

Ládapakolás

Tétel

Ládapakolás NP-teljes.

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

Partíció $\leqslant_p$ Ládapakolás:

Legyenek $b_1, \ldots, b_n$ a B multihalmaz elemei és $b = \sum_{i=1}^{n} b_i$. Legyenek az L multihalmaz elemei $2b_1/b, \ldots, 2b_n/b$. Ekkor könnyen láthatóan $\langle B \rangle \in$ Partíció $\iff \langle L, 2 \rangle \in$ Ládapakolás.

A visszavezetés nyilván polinomiális. $\square$

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 az intenzív próbálkozások ellenére sem, így erős NP-köztes jelölteknek számítanak.

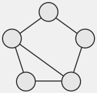
NP-köztes jelöltek – Gráfizomorfizmus

Definíció

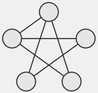
A $G_i = (V_i, E_i)$ $(i = 1, 2)$ irányítatlan gráfok **izomorfak**, ha van olyan $f : V_1 \to V_2$ bijekció, hogy $\forall u, v \in V_1$ esetén $\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$.

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

Példa:



és



izomorfak.

Megjegyzés: A gráfizomorfizmus probléma számos, gyakorlatban előforduló speciális esete P-beli. Például:

- fák
- síkba rajzolható gráfok
- korlátos fokszámú gráfok

NP-köztes jelöltek – Gráfizomorfizmus

Nehéz eset például, ha nagyméretű gráfokban minden fokszám $\sqrt{|V|}$ körüli.

Egy új eredmény: Babai László, magyar matematikus 2017-es eredménye:

Tétel: $\textsc{Gráfizomorfizmus} \in$ QP, ahol

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

a „kvázipolinom időben" megoldható problémák osztálya.

A gráfizomorfizmus probléma alábbi általánosítása viszont már NP-teljes.

$\textsc{Részgráfizomorfizmus} = \{\langle G_1, G_2 \rangle \mid$ G_1 és G_2 irányítatlan gráfok és G_1 izomorf G_2 egy részgráfjával$\}$.

Részgráfizomorfizmus NP-teljes

Megjegyzés: A részgráf nem feszítetten értendő, tehát például $G_1 = (\{a, b, c\}, \{\{a, b\}, \{b, c\}\})$ 2 élből álló út részgráfja $G_2 = (\{A, B, C, D\}, \{\{A, B\}, \{A, C\}, \{A, D\}, \{B, C\}, \{B, D\}, \{C, D\})$ teljes 4 csúcsú gráfnak, hiszen G_1 izomorf $G_3 = (\{A, B, C\}, \{\{A, B\}, \{B, C\}\})$-mal, ami részgráfja G_2-nek annak ellenére, hogy $\{A, C\} \in E(G_2)$.

Tétel

Részgráfizomorfizmus NP-teljes.

Bizonyítás: Részgráfizomorfizmus NP-beli, hiszen egy NTG polinom időben előllíthat egy f bijekciót $V(G_1)$ és $V(G_2)$ egy $|V(G_1)|$ méretű részhalmaza között, majd polinom időben ellenőrizhető, hogy f izomorfizmus-e.

IHK$\leqslant_p$Részgráfizomorfizmus. Legyen G egy irányítatlan gráf. G_1 legyen egy $|V(G)|$ csúcsú kör, $G_2 := G$. Ekkor G-ben van Hamilton kör, akkor és csak akkor, ha G_2-nek van G_1-gyel izomorf részgráfja. A visszavezetés nyilván polinomiális. $\square$

NP-köztes jelöltek – Prímfaktorizáció

Számítási feladat:

Prímfaktorizáció: adjuk meg egy egész szám prímtényezős felbontását!

A probléma eldöntési változata:

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

Alkalmazás:

RSA eljárás: 1976-ban Ron Rivest, Adi Shamir és Len Adleman kifejlesztett egy nyílt kulcsú titkosító algoritmust, amely két nagy prím összeszorzásával azt használja ki, hogy nem ismeretes polinomiális algoritmus egy összetett szám prímtényezőinek meghatározására.

Bár bizonyítani nem tudjuk, hogy $\textsc{Prímfaktorizáció}$ nem P-beli, mégis az RSA algoritmus adatok biztonságos továbbítására a mai napig az egyik leggyakrabban használt algoritmus.

co$\mathcal{C}$ bonyolultsági osztályok

Definíció

Ha $\mathcal{C}$ egy bonyolultsági osztály 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 $L_2 \in \text{co}\mathcal{C}$ 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 $L_1 \in \text{co}\mathcal{C}$. □

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. $\square$

Példák coNP teljes nyelvekre

$\textsc{Unsat} := \{\langle\varphi\rangle \,|\, \varphi$ kielégíthetetlen nulladrendű formula$\}$.

$\textsc{Taut} := \{\langle\varphi\rangle \,|$ a φ nulladrendű formula tautológia$\}$.

Tétel

$\textsc{Unsat}$ és $\textsc{Taut}$ coNP-teljesek.

Bizonyítás: $\textsc{Áltsat} = \{\langle\varphi\rangle \,|\, \varphi$ kielégíthető nulladrendű formula$\}$ is NP-teljes (NP-beli és $\textsc{Sat}$ speciális esete neki.)

$\overline{\textsc{Áltsat}} := \textsc{Unsat}$, az előző tétel alapján $\textsc{Unsat}$ coNP-teljes.

$\textsc{Unsat} \leqslant_p \textsc{Taut}$, hiszen $\varphi \mapsto \neg\varphi$ polinom idejű visszavezetés. $\square$

Informálisan: coNP olyan nyelveket tartalmaz, amelyekbe való tartozás **polinom időben cáfolható**.

Például egy φ-t kielégítő interpretáció cáfolja, hogy φ kielégíthetetlen lenne. Egy φ-t hamisra értékelő interpretáció cáfolja, hogy φ tautológia lenne. Egy interpretáció polinom időben előállítható és adott interpretációban a formula igazságértéke polinom időben kiszámítható.

coNP-teljes nyelvek egy tulajdonsága

Tétel

Ha L coNP-teljes és $L \in \mathrm{NP}$, akkor NP $=$ coNP.

Bizonyítás: Legyen $L' \in \mathrm{coNP}$ tetszőleges. Mivel L coNP-teljes, ezért $L' \leqslant_p L$, de mivel L NP-beli, ezért korábbi tételünk szerint $L' \in \mathrm{NP}$. Tehát $\mathrm{coNP} \subseteq \mathrm{NP}$.

Legyen most $L \in \mathrm{NP}$ tetszőleges, ekkor coNP definíciója miatt $\overline{L} \in \mathrm{coNP}$. $\mathrm{coNP} \subseteq \mathrm{NP}$ miatt $\overline{L} \in \mathrm{NP}$, majd ismét coNP definíciója miatt $L \in \mathrm{coNP}$. Tehát $\mathrm{NP} \subseteq \mathrm{coNP}$ is teljesül. $\square$

Sejtés: NP $\neq$ coNP.

Amennyiben a sejtés igaz, akkor Unsat és Taut még csak nem is NP-beliek. Mindenesetre nem ismeretes ezen nyelvekhez való tartozásra polinom időben kiszámítható és ellenőrizhető bizonyíték (tanú).

NP $\cap$ coNP

Ha NP $\neq$ coNP egy érdekes osztály lehet NP $\cap$ coNP. Nyilván P $\subseteq$ NP $\cap$ coNP. Először nézzünk néhány példát NP $\cap$ coNP-beli nyelvre.

1. példa

ÖSSZEFÜGGŐ:= $\{\langle G\rangle \mid G = (V, E)$ összefüggő, irányítatlan gráf$\}$

ÖSSZEFÜGGŐ NP-beli, hiszen egy n csúcsú gráf összefüggőségére bizonyíték $\binom{n}{2}$ út a pontpárok között. Egy út hossza legfeljebb n. Így egy NTG előállíthat $\binom{n}{2}$ darab legfeljebb n csúcsból álló sorozatot polinom idő alatt, majd polinom időben ellenőrizheti, hogy ezek tényleg utak-e a pontpárok között.

ÖSSZEFÜGGŐ coNP-beli is, hiszen a csúcsok egy olyan X részhalmaza cáfolat, hogy X és $V\backslash X$ között nem megy él. V egy X részhalmazának előállítása és annak ellenőrése hogy X és $V\backslash X$ között nem megy él polinom időben megvalósítható.

ÖSSZEFÜGGŐ P-ben van, például szélességi kereséssel polinom időben ellenőrizhető egy gráf összefüggősége.

$\mathrm{NP} \cap \mathrm{coNP}$

2. példa

Definíció

Legyen $G = (A, B, E)$ (irányítatlan) páros gráf. Egy $M \subseteq E$ élhalmaz **teljes párosítás**, ha az $(A \cup B, M)$ gráfban minden csúcs foka pontosan 1.

$\textsc{Teljes Párosítás} := \{\langle G \rangle \mid G = (A, B, E)$ páros gráfban van teljes párosítás$\}$

$\textsc{Teljes Párosítás}$ NP-beli, hiszen $(a, b) \in A \times B$ párok egy $|A|$ méretű listája polinom időben előállítható, és polinom időben ellenőrizhető, hogy ez teljes párosítást ad.

NP $\cap$ coNP

TELJES PÁROSÍTÁS coNP-belisége Frobenius tételéből adódik.

Tétel: A $G = (A, B, E)$ páros gráfban akkor és csak akkor létezik teljes párosítás, ha $|A| = |B|$ és minden $X \subseteq A$ halmazra legalább $|X|$ olyan B-beli csúcs van, amelyik valamelyek X-beli csúccsal szomszédos.

A egy X részhalmaza polinom időben előállítható és szintén polinom időben ellenőrizhető, hogy nem teljesül-e a rá a tétel feltétele.

TELJES PÁROSÍTÁS P-ben van, amit a **magyar módszer** nevű Kőnig Dénes és Egerváry Jenő munkássága nyomán Harold Kuhn által adott polinomiális algoritmus mutat, mely egy páros gráfban keres maximális méretű (részleges) párosítást.

Ötlete: vegyünk független éleket, amíg tudunk, majd keressünk javító alternáló utat, azaz olyan utat ami egy A-beli és egy B-beli párosításon kívüli csúcs között fut és az élei váltakozva párosításon kívüliek illetve belüliek.

$\mathrm{NP} \cap \mathrm{coNP}$

3. példa Prímtesztelés

$\textsc{Prímek}:= \{p \mid p \text{ prím}\}$.

Fontos észrevétel, hogy egy $p \in \textsc{Prímek}$ hossza p számjegyeinek száma, azaz $\Theta(\log p)$.

$\textsc{Prímek}$ coNP-belisége könnyen látható, hiszen egy n szám legfeljebb $\sqrt{n}$ méretű osztója cáfolja, hogy n prím lenne. A cáfolat mérete $O(\log n)$, a maradékos osztás $O(\log^3 n)$ időben végrehajtható.

Megjegyzés: $\sqrt{n}$-ig determinisztikusan kipróbálni minden számot túl lassú, $\log n$-ben exponenciális.

$\textsc{Prímek}$ NP-belisége már nem ilyen egyszerű, szükség van egy gyorsan ellenőrizhető prímtesztre.

NP $\cap$ coNP

Tétel (Lucas prímtesztje) n prím $\Leftrightarrow$ létezik $1 \leqslant x \leqslant n-1$, melyre $x^{n-1} \equiv 1 \pmod{n}$, de $x^{(n-1)/p} \not\equiv 1 \pmod{n}$ $(n-1)$-nek minden p prímosztójára.

Ez alapján a következő $(\log n)$-ben polinom idejű rekurzív nemdeterminisztikus algoritmus készíthető:

Nemdeterminisztikus prímfelismerés(n)
if $n = 2$ **then return** 'igen';
if $n = 1$ vagy $n > 2$ páros **then return** 'nem';
if $n > 2$ páratlan **then**
legyen $1 < x < n$;
ellenőrizzük, hogy $x^{n-1} \equiv 1 \pmod{n}$ igaz-e
tippelünk $n-1$ prímfelbontására: $p_1, \ldots p_k$
ellenőrizzük, hogy $n-1 = p_1 \cdots p_k$
ellenőrizzük, minden $1 \leqslant i \leqslant k$-ra, hogy p_i prím és
hogy $x^{(n-1)/p_i} \not\equiv 1 \pmod{n}$
return 'igen', ha minden ellenőrzés rendben

NP $\cap$ coNP

Egy potenciális prímfelbontás $O(\log n)$ prímből áll, melyek mindegyike $O(\log n)$ hosszú. Egy mod n hatványozás végrehajtható $O(\log^3 n)$, az összes $O(\log^4 n)$, időben.

Összességében az algoritmus lépésszámára $m = \lceil \log_2 n \rceil$ jegyű bemenet esetén

$T(m) \leqslant cm^4 + \sum_{i=1}^{k} T(m_i)$ alakú rekurzió adódik, ahol m_i a p_i számjegyeinek száma $(1 \leqslant i \leqslant k)$.

Nyilván $\sum_{i=1}^{k} m_i \leqslant m$ és $m_i \leqslant m - 1$ minden $1 \leqslant i \leqslant k$-ra.

Innen teljes indukcióval $T(m) \leqslant cm^5$ bizonyítható:

$T(m) \leqslant cm^4 + \sum_{i=1}^{k} cm_i^5 \leqslant cm^4 + c(m-1)^4 \sum_{i=1}^{k} m_i = cm(m^3 + (m-1)^4) \leqslant cm^5$

Az algoritmus $O(\log^5 n)$ időkorlátos, így Prímek NP-beli.

NP $\cap$ coNP

Sokáig nyitott kérdés volt a számjegyek számában polinomiális determinisztikus prímteszt létezése, azaz, hogy PRÍMEK P-beli-e.

AKS-prímteszt: 2002-ben M. Agrawal, N. Kayal, N. Saxena indiai tudósok készítettek egy $O(\log^{12} n \log^k \log n)$ idejű determinisztikus prímtesztet (k egy konstans), amivel bebizonyították, hogy PRÍMEK P-beli. Munkájukkal elnyerték a Gödel- és Fulkerson-díjakat 2006-ban. Később a hatékonyságot $O(\log^6 n \log^k \log n)$-re sikerült javítani.

A fenti példák azt sugallják, hogy egy NP $\cap$ coNP-beli problémáról végül mindig kiderül, hogy P-beli, ám ez valószínűleg nem igaz.

Sejtés: P $\neq$ NP $\cap$ coNP.

Számításelmélet

12. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

A tárbonyolultság mérésének problémája

Első megközelítésben a tárigény a működés során felhasznált, pontosabban a fejek által meglátogatott cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

Egy megoldási javaslat: Bevezethetjük az többlet tárigény fogalmát, ami az **input tárolására használt cellákon felül** igénybevett cellák száma.

Vannak olyan TG-ek, melyek csak az input területét használják, ám azt akár többször is átírják. Ezt beszámítsuk?

Eldöntési 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

- $\text{SPACE}\,(f(n)) := \{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos determinisztikus offline TG-pel$\}$
- $\text{NSPACE}\,(f(n)) := \{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos nemdeterminisztikus offline TG-pel$\}$
- PSPACE:=$\bigcup_{k\geqslant 1}\text{SPACE}\,(n^k)$.
- NPSPACE:=$\bigcup_{k\geqslant 1}\text{NSPACE}\,(n^k)$.
- 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 $\Longleftrightarrow$ ÚT$(s, t, \lceil \log_2 n \rceil)$=igaz.
- ÚT(x, y, i)=igaz $\Longleftrightarrow$ $\exists z($ 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.

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 $\textsc{Elér} \in \mathrm{TIME}(n^2)$ vagy $\textsc{Elér} \in \mathrm{SPACE}(\log^2 n)$ 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 $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

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

$\mathrm{NL} \subseteq \mathrm{P}$

Bizonyítás

Legyen $L \in \mathrm{NL}$ 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 $L \in \mathrm{P}$.

ELÉR eldöntése nemdeterminisztikus log. tárral

ELÉR fontos szerepet tölt be az L$\stackrel{?}{=}$NL kérdés vizsgálatában is.

Tétel

ELÉR $\in$ NL

Bizonyítás: Az M 3-szalagos NOTG a (G, s, t) inputra ($n = |V(G)|$) a következőt teszi:

- ráírja s-t a második szalagra
- ráírja a 0-t a harmadik szalagra
- Amíg a harmadik szalagon n-nél kisebb szám áll
  – Legyen u a második szalagon lévő csúcs
  – Nemdeterminisztikusan kiválasztja v egy ki-szomszédját és felírja u helyére a második szalagra
  – Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot (binárisan) eggyel
- Ha n-nél nagyobb szám áll a 3. szalagon, akkor elutasítja a bemenetet.

Mindkét szalag tartalmát $O(\log n)$ bittel kódolhatjuk.

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 $L' \in \mathrm{NL}$ nyelvre, $L' \leqslant_\ell L$. Ha ezen felül $L \in \mathrm{NL}$ 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 $L_2 \in \mathrm{L}$.
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

(bizonyítás nélkül)

Hierarchia tétel

EXPTIME:=$\bigcup_{k\in\mathbb{N}}$TIME(2^{n^k}).

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

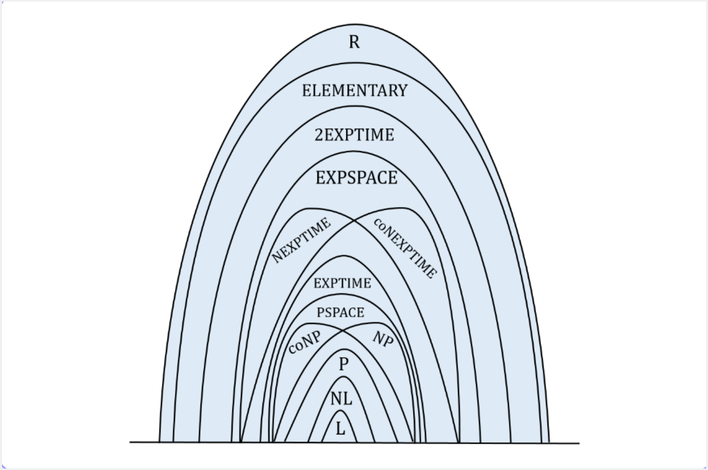
(2): Immerman- Szelepcsényi

(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]

Számításelmélet

13. előadás

előadó: Kolonits Gábor
kolomax@inf.elte.hu

PSPACE-teljes problémák

A hierarchia tétel alapján tudjuk, hogy NP $\subseteq$ PSPACE $\subseteq$ EXPTIME.

Az a sejtésünk, hogy a fenti tartalmazások valódiak.

Számos érdekes probléma tartozik ebbe az osztályba, köztük számos olyan akármekkora táblán játszható kétszemélyes játék nyerő állásainak meghatározása, ahol a lépésszám polinomiálisan korlátozott, a játékosok felváltva lépnek és aktuális lépésük függ az ellenfél előző lépésétől.

A PSPACE-teljes problémák a problémaosztály legnehezebb problémái.

Teljesen kvantifikált Boole formulák

Definíció

Teljesen kvantifikált Boole formulának (TKBF-nek) nevezünk egy olyan formulát, melyben minden ítéletváltozó univerzális vagy egzisztenciális kvantorral kötött, ráadásul úgy, hogy a formula két részre osztható: elől vannak a kvantorok és utána egy kvantormentes ítéletlogikai formula.

A kvantorok hatásköre a saját pozíciójuktól a formula végéig tart.

Példák:

$\varphi_1 = \exists x_1 \forall x_2 (x_1 \to x_2)$

$\varphi_2 = \exists x_1 \forall x_2 ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))$

A TKBF-ek zártak, igazságértékük változókiértékeléstől független.

φ_1 igaz, hiszen bármi is x_2 igazságértéke, ha x_1-et hamisra állítjuk az ítéletlogikai rész igaz lesz.

φ_2 hamis, hiszen x_1 akár igaz akár hamis, van x_2-nek olyan igazságértéke, hogy az egyik tag hamis lesz.

QSAT PSPACE-teljes

Definíció

$\textsc{QSat} = \{\langle\varphi\rangle \mid \varphi$ igaz teljesen kvantifikált Boole formula$\}$.

Az alábbi tétel szerint QSAT a legnehezebb problémák közé tartozik a PSPACE osztályon belül.

Tétel

QSAT PSPACE-teljes.

A PSPACE-teljességet a polinom idejű visszavezetésre nézve értjük.

QSAT PSPACE-teljes

Bizonyítás: QSAT PSPACE-beliségéhez tekintsük az ÉRTÉK(φ) függvényt, mely visszaadja a φ TKBF igazságértékét. ÉRTÉK(φ)-t kiszámíthatjuk a következő rekurzív algoritmussal.

function ÉRTÉK(φ)
  if $\varphi = \forall x \psi$ **then return** ÉRTÉK(ψ^i) $\wedge$ ÉRTÉK(ψ^h)
  else if $\varphi = \exists x \psi$ **then return** ÉRTÉK(ψ^i) $\vee$ ÉRTÉK(ψ^h)
  else return $B(\varphi)$

ahol ha ψ egyetlen szabad változóval rendelkező formula

ψ^i : ψ egyetlen szabad változóját igazzal helyettesítjük.

ψ^h : ψ egyetlen szabad változóját hamissal helyettesítjük.

továbbá ha φ változómentes formula

$B(\varphi)$: a φ kiértékelésével kapott igazságérték.

Példa: Ha $\psi = \forall x(x \wedge \neg y)$, akkor $\psi^h = x \wedge \neg h$.
Ha $\varphi = i \wedge \neg h \rightarrow h$, akkor $B(\varphi) = h$.

QSAT PSPACE-teljes

Például $\varphi = \exists x_1 \forall x_2((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))$-re:

$\varphi_{ii} = ((\neg i \vee i) \wedge (i \vee \neg i))$. $\text{ÉRTÉK}(\varphi_{ii}) = B(\varphi_{ii}) = i$.
$\varphi_{ih} = ((\neg i \vee h) \wedge (i \vee \neg h))$. $\text{ÉRTÉK}(\varphi_{ih}) = B(\varphi_{ih}) = h$.

$\varphi_i = \forall x_2((\neg i \vee x_2) \wedge (i \vee \neg x_2))$

$((\neg i \vee x_2) \wedge (i \vee \neg x_2))^i = \varphi_{ii}$. $((\neg i \vee x_2) \wedge (i \vee \neg x_2))^h = \varphi_{ih}$.

$\text{ÉRTÉK}(\varphi_i) = \text{ÉRTÉK}(\varphi_{ii}) \wedge \text{ÉRTÉK}(\varphi_{ih}) = i \wedge h = h$.

$\varphi_{hi} = ((\neg h \vee i) \wedge (h \vee \neg i))$ $\text{ÉRTÉK}(\varphi_{hi}) = B(\varphi_{hi}) = h$.
$\varphi_{hh} = ((\neg h \vee h) \wedge (h \vee \neg h))$ $\text{ÉRTÉK}(\varphi_{hh}) = B(\varphi_{hh}) = i$.

$\varphi_h = \forall x_2((\neg h \vee x_2) \wedge (h \vee \neg x_2))$.

$((\neg h \vee x_2) \wedge (h \vee \neg x_2))^i = \varphi_{ii}$. $((\neg h \vee x_2) \wedge (h \vee \neg x_2))^h = \varphi_{ih}$.

$\text{ÉRTÉK}(\varphi_h) = \text{ÉRTÉK}(\varphi_{hi}) \wedge \text{ÉRTÉK}(\varphi_{hh}) = h \wedge i = h$.

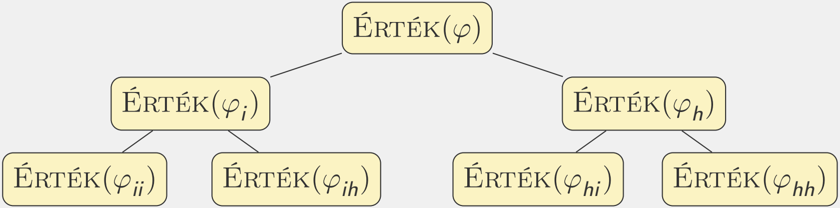
$\varphi = \exists x_1 \forall x_2((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))$.

$\forall x_2((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))^i = \varphi_i$.
$\forall x_2((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2))^h = \varphi_h$.

$\text{ÉRTÉK}(\varphi) = \text{ÉRTÉK}(\varphi_i) \vee \text{ÉRTÉK}(\varphi_h) = h \vee h = h$.

QSAT PSPACE-teljes

A rekurzív függvényhívások fája:



Az algoritmus tárigénye a formula méretébe lineáris, hiszen a rekurzív hívási fa mélysége $O(n)$ és elég a rekurzió minden szintjén egy változó igazságértékét tárolni (a második gyerekre csak akkor hívjuk meg az eljárást, ha a szülő értéke nem következik az első gyerek értékéből, ha pedig meghívtuk a második gyerekre, abból következik az első gyerek visszaadott értéke). Tehát QSAT $\in$ PSPACE.

$\textsc{Qsat}$ PSPACE-teljes

A PSPACE-nehézség bizonyításához legyen L tetszőleges PSPACE-beli nyelv. Megmutatjuk, hogy $L \leqslant_p \textsc{Qsat}$. Legyen $M = \langle Q, \{0,1\}, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy $p(n)$ többlet tárigényű L-et eldöntő offline Turing gép és w az M egy tetszőleges $n (n \in \mathbb{N})$ hosszú bemenete. Feltehető, hogy $p(n)$ polinom és $p(n) \geqslant n$. Megadunk egy φ TKBF-et úgy, hogy $w \in L(M)$ akkor és csak akkor, ha $\langle \varphi \rangle \in \textsc{Qsat}$.

A tárigényre vonatkozó korlátból adódik, hogy M-nek w-re vonatkozó számítása legfeljebb $h = 2^{cp(n)}$ különböző konfigurációt tartalmaz valamely $c > 0$-ra. Így feltehető, hogy M legfeljebb h lépésben elfogadja vagy elutasítja w-t, és az előző előadáson látottak alapján az is, hogy egyetlen elfogadó állapot van.

QSAT PSPACE-teljes

A bizonyítás során azzal az egyszerűsítéssel élünk, hogy a konfigurációkat az egyetlen munkaszalag tartalma, ezen szalag fejének helyzete és az aktuális állapot együttesen meghatározzák.

A lehetséges konfigurációk legyenek $c_1, \ldots, c_h$, köztük $c_1 = c_{\text{kezdő}}$, $c_h = c_{\text{elfogadó}}$.

Ítéletváltozók: $x_{i,j,s}$ $(1 \leqslant i \leqslant h, 1 \leqslant j \leqslant p(n), s \in Q \cup \Gamma)$.
$x_{i,j,s}$ annak az állításnak felel meg, hogy c_i j-edik betűje s.

Egy I interpretáció akkor és csak akkor írja le azt, hogy a c_i konfiguráció legális, ha

- minden $1 \leqslant j \leqslant p(n)$-re pontosan egy $s \in Q \cup \Gamma$ van, amelyre $x_{i,j,s}$ igaz
- pontosan egy $1 \leqslant j \leqslant p(n)$-hez van olyan $s \in Q$, hogy $x_{i,j,s}$ igaz

QSAT PSPACE-teljes

Jelölés: ha X ítéletváltozók egy halmaza, akkor $\exists X$-szel rövidítjük az X-beli változók egzisztenciális kvantifikálását, azaz a $\exists x_1 \cdots \exists x_m$-et, ha $X = \{x_1, \ldots x_m\}$. $\forall X$-re analóg módon.

Mi csak konfigurációkat azonosító változóhalmazokat fogunk tekinteni, azaz az $X_k = \{x_{k,j,s} \mid 1 \leqslant j \leqslant p(n), s \in Q \cup \Gamma\}$ változóhalmaz megfelel a c_k konfigurációnak minden $1 \leqslant k \leqslant h$-ra.

Példa: Legyen $c_k = aqbb$ egy konfiguráció.

A következő formula azt írja le, hogy az X_k-beli változók értékei úgy vannak beállítva, hogy azok a c_k konfigurációt határozzák meg.

$$\exists X_k (x_{k,1,a} \wedge x_{k,2,q} \wedge \; x_{k,3,b} \wedge x_{k,4,b} \wedge \bigwedge_{x \in X_k \setminus H} \neg x),$$

ahol $H = \{x_{k,1,a}, x_{k,2,q}, x_{k,3,b}, x_{k,4,b}\}$.

A formula persze igaz és mérete $O(p(n))$.

Qsat PSPACE-teljes

EQ(X_k, X_ℓ) egy olyan formulát jelöl, ami akkor és csak akkor igaz, ha minden $1 \leqslant j \leqslant p(n)$, $s \in Q \cup \Gamma$-ra $x_{k,j,s}$ és $x_{\ell,j,s}$ igazságértéke ugyanaz.

$\varphi_{k,\ell,t}$ jelölje azt a TKBF-et, ami akkor és csak akkor igaz, ha c_k-ból legfeljebb 2^t konfigurációátmenettel el lehet jutni c_ℓ-be.

Ekkor az Elér $\in$ SPACE$(\log^2 n)$ bizonyításánál látott módon $\varphi_{k,\ell,t}$ igaz, akkor és csak akkor, ha van olyan $1 \leqslant m \leqslant h$, hogy $\varphi_{k,m,t-1}$ és $\varphi_{m,\ell,t-1}$ is igaz.

$\varphi_{k,\ell,0}$ leírható egy $O(p(n))$ hosszú EQ$(X_k, X_\ell) \vee \varphi'_{k,\ell,0}$ formulával, ahol $\varphi'_{k,\ell,0}$ azt írja le, hogy $c_k \vdash c_\ell$. Ugyanis a konfigurációk $O(p(n))$ hosszúak, a változás 2x3-as „ablaka" $O(p(n))$ pozícióban lehet és konstans féleképpen tölthető ki, a többi változó egyenlősége szintén egy $O(p(n))$ hosszú részformulával leírható.

Azonban a $\varphi_{k,\ell,t} = \exists X_m(\varphi_{k,m,t-1} \wedge \varphi_{m,\ell,t-1})$ rekurzió nem jó, mivel ez exponenciális sok rekurzív hívást és így exponenciális méretű formulát és időigényt eredményezne.

$\textsc{Qsat}$ PSPACE-teljes

Ehelyett:

$$\varphi_{k,\ell,t} = \exists X_m \forall X_i \forall X_j ((\mathsf{EQ}(X_i, X_k) \wedge \mathsf{EQ}(X_j, X_m)) \vee$$
$$(\mathsf{EQ}(X_i, X_m) \wedge \mathsf{EQ}(X_j, X_\ell)) \to \varphi_{i,j,t-1}),$$

ami összesen csak $O(\log h) = O(p(n))$ rekurzív hívást eredményez (hiszen $w \in L$ akkor és csak akkor, ha $\varphi_{1,h,\lceil \log_2 h \rceil}$ igaz) és $\varphi_{k,\ell,t}$ $\varphi_{i,j,t-1}$-n kívüli része $O(p(n))$ méretű ($\forall\, 1 \leqslant t \leqslant \lceil \log_2 h \rceil$-re).

Tehát $\varphi' := \exists X_1 \exists X_h (\varphi_K \wedge \varphi_{1,h,\lceil \log_2 h \rceil} \wedge \varphi_V)$ polinom időben kiszámítható, ahol a φ_K és φ_V részformulák literálok 1-1 olyan konjunkciója, melyek M w-hez tartozó c_1 kezdő- és c_h elfogadó konfigurációját írják le.

A kapott formulában nem feltétlen van minden kvantor a formula elején, de ez a kvantorok kiemelésével orvosolható, így $L \leqslant_p \textsc{Qsat}$.

Ezzel $\textsc{Qsat}$ PSPACE-beli és PSPACE-nehéz, tehát PSPACE-teljes.

QSAT mint kétszemélyes játék

Megjegyzés: QSAT-nak az a változata is PSPACE teljes, amikor a kvantormentes résznek KNF-nek kell lennie. Ez nem teljesen nyilvánvaló, hiszen egy ítéletlogikai formulával tautológikusan ekvivalens KNF megadása akár exponenciális méretű formulát is eredményezhet.

A QSAT azon megszorítása amikor csak olyan bemeneteket tekintünk ahol a kvantorok alternálnak, az első és az utolsó kvantor a $\exists$, és a kvantormentes rész egy KNF, felfogható úgy, mint az alábbi **kétszemélyes játék**.

Tegyük fel, hogy adott egy ilyen alakú φ TKBF. Az első játékos választja meg a páratlan sorszámú változók értékét, és a célja a formula igazzá tétele. A második választja meg a páros sorszámú változók értékét, és a célja a formula hamissá tétele. Nyilvánvaló, hogy az első játékosnak akkor és csak akkor van nyerő stratégiája ebben a játékban, ha $\langle\varphi\rangle \in \text{QSAT}$.

Ez a változat is PSPACE-teljes.

Földrajzi játék

Definíció

A **Földrajzi játék** a következő kétszemélyes játék. Adott egy G irányított gráf és annak egy p csúcsa. A két játékos p-ből kiindulva felváltva jelöli meg G még meg nem jelölt csúcsait, mindig az utoljára megjelölt csúcsból elérhető csúcsok közül választva. Az veszít, aki nem tud további csúcsot megjelölni.

1. példa: Alíz és Borcsa a következő játékot játssza. Megyeszékhelyeket mondanak felváltva, a következő játékosnak az előző játékos által mondott város utolsó betűjével kell kezdenie a sajátját.

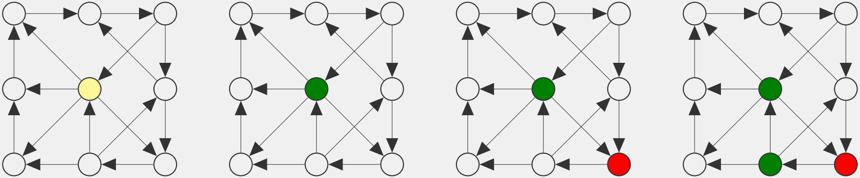
Alíz: Szolnok
Borcsa: Kecskemét
Alíz: Tatabánya

Alíz nyert, mert Borcsa nem tud mondani A-val új megyeszékhelyt.

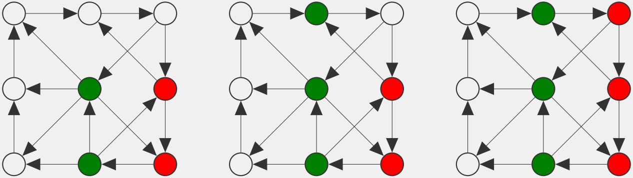
Ez egy földrajzi játék a megyeszékhelyek a gráfján, ahol u városból akkor vezet irányított út v városba, ha u utolsó betűje megegyezik v első betűjével (amennyiben a kezdőváros rögzített).

Földrajzi játék

2. példa: Az előző játék általánosítása, ha **Alíz** és **Borcsa** egy tetszőleges gráfon játszanak. Alíz kezd.



: startcsúcs



Borcsa nyert, Alíz nem tud lépni.

Erre a gráfra és kezdőcsúcsra Borcsának van nyerő stratégiája.
Az előző játék bizonyítja, Alíznak mindig csak 1 lépése volt.

Földrajzi játék PSPACE-teljes

Definíció

Földrajzi játék$=\{\langle G, p\rangle \mid$ a G irányított gráfban p kezdőcsúccsal az 1. játékosnak van nyerő stratégiája$\}$

Tétel

Földrajzi játék PSPACE-teljes.

Bizonyítás: A játék **állása** alatt G csúcsainak egy olyan 2 színnel színezését értjük, ahol az egyik színosztály a jelölt, a másik a jelöletlen csúcsok, és még az az információ, hogy ki következik soron.

Földrajzi játék PSPACE-beliségének belátásához készítsük el a játék fáját, ahol a csúcsok az állások és egy állás gyerekei a lehetséges következő állások. Irányítsuk a fát a levelektől a gyökér felé.

FÖLDRAJZI JÁTÉK PSPACE-teljes

A leveleket értékeljük ki igazra vagy hamisra aszerint, hogy az az 1. játékosnak a győzelmét vagy vereségét jelenti-e.
Bármely más, nem levél állás tekinthető egy VAGY kapunak ha az 1. játékos van soron, illetve egy ÉS kapunak, ha a 2. játékos következik. A kettőnél több bemenettel rendelkező kapuk helyettesíthetők bináris kapukkal.
A kiértékelés polinom tárral elvégezhető, mert

- minden játék polinomiális hosszúságú (legfeljebb a gráf csúcsszáma)
- adott állásból polinomiális tárral megkonstruálható minden lehetséges következő állás, illetve ha nincs következő állás akkor eldönthető ki nyert
- a játékfa polinomiális magasságú, és ez akkor is igaz, ha csak bináris kapukat használunk (az eredeti játékfában a fokszám nem nagyobb, mint G-ben a maximális ki-fok, ami konstans).

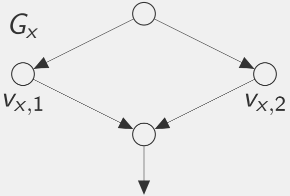
Ezekből a QSAT-nál már látott érvelés alapján következik, hogy a játékfa polinom tárral kiértékelhető.

Földrajzi játék PSPACE-teljes

Földrajzi játék PSPACE-nehézségének bizonyításához Qsat alternáló kvantoros, KNF magú változatát vezetjük vissza rá polinom időben.

Legyen tehát $\varphi = \exists x_1 \forall x_2 \cdots \exists x_k \psi$ TKBF, ahol $\psi = c_1 \wedge \cdots \wedge c_m$, ahol $m \geqslant 1$ és a c_i-k klózok, azaz ψ KNF.

Megkonstruáljuk a G_φ gráfot a következőképpen. Először minden φ-beli x változóhoz elkészítjük az alábbi G_x részgráfot.



A G_x részgráfok „egymás alatt" vannak és össze vannak kötve.

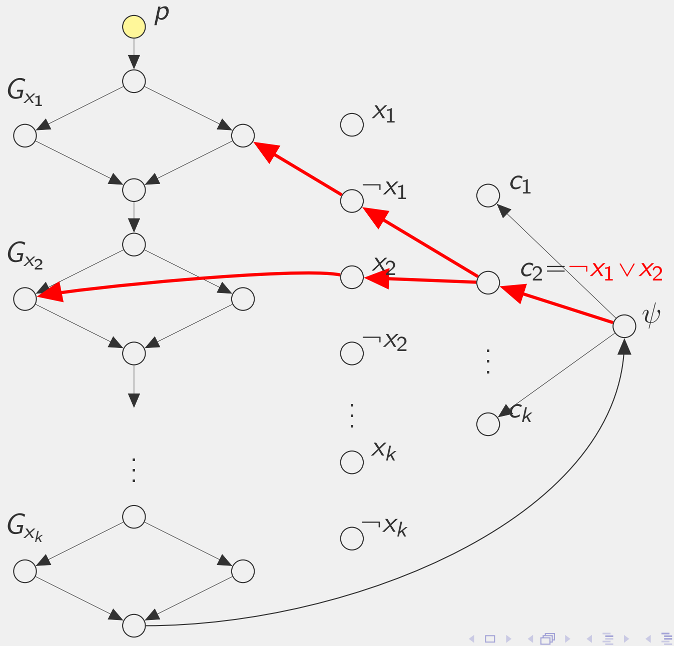
Földrajzi játék PSPACE-teljes

G_φ-nek van egy csúcsa ami ψ-vel van címkézve. Az ebből kiinduló élek olyan csúcsokba vezetnek, melyek rendre ψ klózaival vannak címkézve. Tartozik egy csúcs minden literálhoz is. Minden klóznak megfelelő csúcsból vezessen él azoknak a literáloknak megfelelő csúcsokhoz, amelyekből az adott klóz áll.

Minden literállal címkézett csúcsból egyetlen él induljon ki a következőképpen. Ha a csúcs x-el van címkézve valamely x változóra, akkor a belőle kiinduló él G_x $v_{x,1}$ csúcsába, ha $\neg x$-el van címkézve, akkor G_x $v_{x,2}$ csúcsába vezessen.

A p csúcsból induljon egyetlen él G_{x_1} felső csúcsába, valamint vegyünk hozzá egy élt G_{x_k} alsó csúcsából a ψ címkéjű csúcsba.

Földrajzi játék PSPACE-teljes



Földrajzi játék PSPACE-teljes

Nézzük meg, hogyan zajlik egy játék G_φ-ben.

A páratlan sorszámú változóknak megfelelő G_x részgráfok legfelső csúcsaiban az első játékos választja az irányt (azaz kezdetben ő választ egy csúcsot a p-ből kiindulva. A páros sorszámúaknak megfelelőkben viszont a második játékos. G_{x_k}-ban az első játékos választja az irányt és így a 2. játékos a legalsó csúcsot.

Az első játékosnak kell választania a ψ címkéjű csúcsot. A második játékos választ egy c klózt, majd az első játékos kiválasztja c egy ℓ literálját.

Itt két lehetőség van: vagy tud a második játékos még korábban nem választott csúcsot választani vagy nem.

G_φ konstrukciója miatt a második játékos pontosan akkor tud csúcsot választani, ha a következők egyike teljesül:

- $\ell = x$ valamely x változóra és $v_{x,1}$-et még nem jelölték meg,
- $\ell = \neg x$ valamely x változóra és $v_{x,2}$-et még nem jelölték meg.

Földrajzi játék PSPACE-teljes

Belátható, hogy az 1. játékosnak pontosan akkor van nyerő stratégiája G_φ-ben, ha a φ-beli kétszemélyes játék során is van, azaz pontosan akkor ha az 1. játékos meg tudja adni a páratlan indexű ítéletváltozók értékét úgy, hogy később akárhogyan is választ a második játékos egy ψ-beli klózt, az 1. játékos ki tud választani abban a klózban egy olyan literált, ami igaz lesz a kettejük játéka által meghatározott interpretációban.

Ebből következik, hogy a $\varphi \mapsto G_\varphi$ megfeleltetés Qsat visszavezetése a Földrajzi játék-ra.

Az is könnyen látható, hogy G_φ a φ méretében polinom időben megkonstruálható, és ezzel a tételt bebizonyítottuk. $\square$

További PSPACE-teljes problémák

1-típusú szóprobléma

Reguláris grammatikák esetében lineáris, környezetfüggetlen grammatikák esetében pedig köbös időigényű algoritmust (CYK) tanultunk is ismert a szóprobléma eldöntésére az előző félévben.

Mivel $\mathcal{L}_0 =$RE, ezért a 0. típusú szóprobléma eldönthetetlen (de RE-beli).

Előző félévben tanultunk egy, az 1-típusú szóproblémát eldöntő exponenciális időigényű algoritmust. Bizonyítás nélkül megjegyezzük, hogy az 1-típusú szóprobléma PSPACE-teljes.

További PSPACE-teljes problémák

Általánosított, korlátos idejű go játék

Tétel: Annak eldöntése, hogy egy $n \times n$-es táblán legfeljebb n^2 lépésig játszott go játékban egy adott állás fehér számára nyerő állás-e PSPACE-teljes.

Megjegyzés: A konkrét méretben (például 8x8-as vagy 19x19-es) táblán játszott kétszemélyes játékok (go, dáma, sakk) véges játékok, a PSPACE-teljesség azonban aszimptotikus fogalom. Ezekben a játékokban a keresési tér véges, bár hatalmas.

Megjegyzés: Ha elhagyjuk a lépésszámkorlátot, akkor a probléma EXPTIME-teljessé válik.

További PSPACE-teljes problémák

Helyben elfogadás problémája

HELYBEN ELFOGADÁS$=\{\langle M, w\rangle \mid$ az M determinisztikus TG úgy fogadja el w-t, hogy a fej nem hagyja el w területét$\}$

Tétel

HELYBEN ELFOGADÁS PSPACE-teljes.

Bizonyítás: HELYBEN ELFOGADÁS PSPACE-beli:
M lehetséges konfigurációinak száma egy w bemenetre legfeljebb $m(w) = |Q| \cdot |w| \cdot |\Gamma|^{|w|}$, ahol Q az M állapothalmaza és Γ a szalagábécéje.
HELYBEN ELFOGADÁS a következő S Turing géppel polinom tárral eldönthető.

- S szimulálja M-et w-n
- ha M elutasítja w-t akkor S lépjen q_n-be
- ha M feje elhagyja w területét, akkor S lépjen q_n-be
- ha M elfogadja w-t, akkor S lépjen q_i-be
- ha M $m(w)$-nél többet lép, akkor S lépjen q_n-be

További PSPACE-teljes problémák

Számolni kell a megtett lépéseket. A lépések száma $O(|w|)$ bittel leírható.

Az S determinisztikus TG működése során tehát $O(|w|) = O(|\langle M, w\rangle|)$ tárat használ és $L(S) =$ HELYBEN ELFOGADÁS.

Tehát HELYBEN ELFOGADÁS $\in$ PSPACE

Legyen $L \in$ PSPACE tetszőleges. Ekkor van olyan M determinisztikus TG, mely L-et $p(n)$ tárral eldönti. Tetszőleges w szóra legyen $f(w) := \langle M', \#^{p(n)} w \#^{p(n)}\rangle$, ahol $\#$ egy olyan szimbólum, ami nincs benne M szalagábécéjében, M' pedig w első betűjére teker, majd ugyanúgy működik, mint M $\#$-t $\sqcup$-ként kezelve.

Ekkor f polinom idejű visszavezetés, hiszen $w \in L(M)$ akkor és csak akkor ha $\langle M', \#^{p(n)} w \#^{p(n)}\rangle \in$ HELYBEN ELFOGADÁS.

Így $L \leqslant_p$ HELYBEN ELFOGADÁS bizonyítva, hogy HELYBEN ELFOGADÁS PSPACE-nehéz. Mivel láttuk, hogy PSPACE-beli, ezért PSPACE-teljes is. $\square$