Számítási modellek

4. előadás

A logika két modellje

Nulladrendű logika:

A logika **nulladrendű modelljében** a formulák ítéletváltozókból $X, Y, \ldots$ épülnek fel logikai műveletek $\neg$, $\wedge$, $\vee$ segítségével. Példa: $\neg(X \vee \neg(Y \wedge \neg Z))$.

A logika két modellje

Nulladrendű logika:

A logika **nulladrendű modelljében** a formulák ítéletváltozókból $X, Y, \ldots$ épülnek fel logikai műveletek $\neg, \wedge, \vee$ segítségével. Példa: $\neg(X \vee \neg(Y \wedge \neg Z))$.

A változókat **kiértékelhetjük** igazra/hamisra. Adott változókiértékelés mellett a formula szerkezete alapján rekurzíve kiszámítható a **formula Boole-értéke** ($\neg$: negáció, $\wedge$: logikai és, $\vee$: megengedő vagy).

A logika két modellje

Nulladrendű logika:

A logika **nulladrendű modelljében** a formulák ítéletváltozókból $X, Y, \ldots$ épülnek fel logikai műveletek $\neg, \wedge, \vee$ segítségével. Példa: $\neg(X \vee \neg(Y \wedge \neg Z))$.

A változókat **kiértékelhetjük** igazra/hamisra. Adott változókiértékelés mellett a formula szerkezete alapján rekurzíve kiszámítható a **formula Boole-értéke** ($\neg$: negáció, $\wedge$: logikai és, $\vee$: megengedő vagy).

Egy formula **kielégíthető**, ha a 2^n lehetséges változókiértékelés közül legalább egy esetben a formula Boole-értéke igaz, **kielégíthetetlen** különben (n a változók száma).

A logika két modellje

Nulladrendű logika:

A logika **nulladrendű modelljében** a formulák ítéletváltozókból $X, Y, \ldots$ épülnek fel logikai műveletek $\neg, \wedge, \vee$ segítségével. Példa: $\neg(X \vee \neg(Y \wedge \neg Z))$.

A változókat **kiértékelhetjük** igazra/hamisra. Adott változókiértékelés mellett a formula szerkezete alapján rekurzíve kiszámítható a **formula Boole-értéke** ($\neg$: negáció, $\wedge$: logikai és, $\vee$: megengedő vagy).

Egy formula **kielégíthető**, ha a 2^n lehetséges változókiértékelés közül legalább egy esetben a formula Boole-értéke igaz, **kielégíthetetlen** különben (n a változók száma). φ **tautologikusan igaz**, ha minden változókiértékelés esetén igaz ($\models \varphi$).

A logika két modellje

Nulladrendű logika:

A logika **nulladrendű modelljében** a formulák ítéletváltozókból $X, Y, \ldots$ épülnek fel logikai műveletek $\neg$, $\wedge$, $\vee$ segítségével. Példa: $\neg(X \vee \neg(Y \wedge \neg Z))$.

A változókat **kiértékelhetjük** igazra/hamisra. Adott változókiértékelés mellett a formula szerkezete alapján rekurzíve kiszámítható a **formula Boole-értéke** ($\neg$: negáció, $\wedge$: logikai és, $\vee$: megengedő vagy).

Egy formula **kielégíthető**, ha a 2^n lehetséges változókiértékelés közül legalább egy esetben a formula Boole-értéke igaz, **kielégíthetetlen** különben (n a változók száma). φ **tautologikusan igaz**, ha minden változókiértékelés esetén igaz ($\models \varphi$). Egy Φ formulahalmaz **tautologikus következménye** φ , ha minden olyan változókiértékelésben, amiben Φ összes formulája igaz, igaz φ is ($\Phi \models \varphi$).

A logika két modellje

Ha X egy ítéletváltozó, X-et és $\neg X$-et **literálnak**. Literálok diszjunkcióját **elemi diszjunkciónak** vagy más néven **klóznak**. Ilyenek konjunkcióját **konjunktív normálformának** (KNF) nevezzük. Példa: $(\neg X \vee Y) \wedge (X \vee \neg Y \vee Z) \wedge \neg Z$.

A logika két modellje

Ha X egy ítéletváltozó, X-et és $\neg X$-et **literálnak**. Literálok diszjunkcióját **elemi diszjunkciónak** vagy más néven **klóznak**. Ilyenek konjunkcióját **konjunktív normálformának** (KNF) nevezzük. Példa: $(\neg X \vee Y) \wedge (X \vee \neg Y \vee Z) \wedge \neg Z$.

Állítás: Minden formulához van vele ekvivalens KNF.

A logika két modellje

Ha X egy ítéletváltozó, X-et és $\neg X$-et **literálnak**. Literálok diszjunkcióját **elemi diszjunkciónak** vagy más néven **klóznak**. Ilyenek konjunkcióját **konjunktív normálformának** (KNF) nevezzük. Példa: $(\neg X \vee Y) \wedge (X \vee \neg Y \vee Z) \wedge \neg Z$.

Állítás: Minden formulához van vele ekvivalens KNF.

Elsőrendű logika:

A logika két modellje

Ha X egy ítéletváltozó, X-et és $\neg X$-et **literálnak**. Literálok diszjunkcióját **elemi diszjunkciónak** vagy más néven **klóznak**. Ilyenek konjunkcióját **konjunktív normálformának** (KNF) nevezzük. Példa: $(\neg X \vee Y) \wedge (X \vee \neg Y \vee Z) \wedge \neg Z$.

Állítás: Minden formulához van vele ekvivalens KNF.

Elsőrendű logika:

Adott **függvényszimbólumok**, **predikátumszimbólumok** és **konstansok** egy-egy véges halmaza. Az előbbiek az **aritásukkal** együtt (hány változósak). Továbbá legyenek $x, y, \ldots$ ún. **individuumváltozók**.

A logika két modellje

Ha X egy ítéletváltozó, X-et és $\neg X$-et **literálnak**. Literálok diszjunkcióját **elemi diszjunkciónak** vagy más néven **klóznak**. Ilyenek konjunkcióját **konjunktív normálformának** (KNF) nevezzük. Példa: $(\neg X \vee Y) \wedge (X \vee \neg Y \vee Z) \wedge \neg Z$.

Állítás: Minden formulához van vele ekvivalens KNF.

Elsőrendű logika:

Adott **függvényszimbólumok**, **predikátumszimbólumok** és **konstansok** egy-egy véges halmaza. Az előbbiek az **aritásukkal** együtt (hány változósak). Továbbá legyenek $x, y, \dots$ ún. **individuumváltozók**.

A **termek** konstansokból, változókból és függvényjelekből épülnek fel figyelembe véve az aritást. Például $g(f(x, a), y)$, itt a konstans, f, g 2 aritású függvényjelek.

A logika két modellje

Ha X egy ítéletváltozó, X-et és $\neg X$-et **literálnak**. Literálok diszjunkcióját **elemi diszjunkciónak** vagy más néven **klóznak**. Ilyenek konjunkcióját **konjunktív normálformának** (KNF) nevezzük. Példa: $(\neg X \vee Y) \wedge (X \vee \neg Y \vee Z) \wedge \neg Z$.

Állítás: Minden formulához van vele ekvivalens KNF.

Elsőrendű logika:

Adott **függvényszimbólumok**, **predikátumszimbólumok** és **konstansok** egy-egy véges halmaza. Az előbbiek az **aritásukkal** együtt (hány változósak). Továbbá legyenek $x, y, \ldots$ ún. **individuumváltozók**.

A **termek** konstansokból, változókból és függvényjelekből épülnek fel figyelembe véve az aritást. Például $g(f(x, a), y)$, itt a konstans, f, g 2 aritású függvényjelek.

Az **atomi formula** egyetlen predikátumszimbólum aritásnyi term argumentummal. A **formulák** atomi formulákból épülnek fel $\neg, \wedge, \vee, \exists x, \forall x$ segítségével.

A logika két modellje

Példa: $P(x, f(x)) \vee \exists y Q(y)$, itt P 2 aritású, Q 1 aritású predikátumszimbólum, f 1 aritású függvényjel.

A logika két modellje

Példa: $P(x, f(x)) \vee \exists y Q(y)$, itt P 2 aritású, Q 1 aritású predikátumszimbólum, f 1 aritású függvényjel.

A **formulák Boole-értékének** meghatározásához először keresünk egy matematikai struktúrát: egy U alaphalmazt, majd ezen a függvényszimbólumoknak illetve predikátumszimbólumoknak aritásuk szerint függvényeket illetve relációkat feleltetünk meg U-n. A konstansoknak szintén megfeleltetünk 1-1 U-beli elemet (interpretáció). Majd a változóknak U-beli értéket adunk (változókiértékelés).

A logika két modellje

Példa: $P(x, f(x)) \vee \exists y Q(y)$, itt P 2 aritású, Q 1 aritású predikátumszimbólum, f 1 aritású függvényjel.

A **formulák Boole-értékének** meghatározásához először keresünk egy matematikai struktúrát: egy U alaphalmazt, majd ezen a függvényszimbólumoknak illetve predikátumszimbólumoknak aritásuk szerint függvényeket illetve relációkat feleltetünk meg U-n. A konstansoknak szintén megfeleltetünk 1-1 U-beli elemet (interpretáció). Majd a változóknak U-beli értéket adunk (változókiértékelés).

Így a termeknek lesz U-beli értékük. Egy n argumentumú atomi formula akkor legyen igaz, ha a termeiből álló érték n-es a predikátumszimólumnak megfeleltetett relációban áll. Innen $\neg, \wedge, \vee$: szokásos, $\exists x$: létezik U-beli elem..., $\forall x$: minden U-beli elemre ...

A logika két modellje

Példa: $P(x, f(x)) \vee \exists y Q(y)$, itt P 2 aritású, Q 1 aritású predikátumszimbólum, f 1 aritású függvényjel.

A **formulák Boole-értékének** meghatározásához először keresünk egy matematikai struktúrát: egy U alaphalmazt, majd ezen a függvényszimbólumoknak illetve predikátumszimbólumoknak aritásuk szerint függvényeket illetve relációkat feleltetünk meg U-n. A konstansoknak szintén megfeleltetünk 1-1 U-beli elemet (interpretáció). Majd a változóknak U-beli értéket adunk (változókiértékelés).

Így a termeknek lesz U-beli értékük. Egy n argumentumú atomi formula akkor legyen igaz, ha a termeiből álló érték n-es a predikátumszimólumnak megfeleltetett relációban áll. Innen$\neg$, $\wedge$, $\vee$: szokásos, $\exists x$: létezik U-beli elem..., $\forall x$: minden U-beli elemre ...

Egy formula **kielégíthető**, ha van olyan interpretáció és változókiértékelés melyre a formula Boole-értéke igaz.

A logika két modellje

Példa: $P(x, f(x)) \vee \exists y Q(y)$, itt P 2 aritású, Q 1 aritású predikátumszimbólum, f 1 aritású függvényjel.

A **formulák Boole-értékének** meghatározásához először keresünk egy matematikai struktúrát: egy U alaphalmazt, majd ezen a függvényszimbólumoknak illetve predikátumszimbólumoknak aritásuk szerint függvényeket illetve relációkat feleltetünk meg U-n. A konstansoknak szintén megfeleltetünk 1-1 U-beli elemet (interpretáció). Majd a változóknak U-beli értéket adunk (változókiértékelés).

Így a termeknek lesz U-beli értékük. Egy n argumentumú atomi formula akkor legyen igaz, ha a termeiből álló érték n-es a predikátumszimólumnak megfeleltetett relációban áll. Innen$\neg$, $\wedge$, $\vee$: szokásos, $\exists x$: létezik U-beli elem..., $\forall x$: minden U-beli elemre ...

Egy formula **kielégíthető**, ha van olyan interpretáció és változókiértékelés melyre a formula Boole-értéke igaz. φ **logikailag igaz**, ha minden interpretáció és változókiértékelés esetén igaz. **Logikai következmény:** mint nulladrendben.

A Turing gépek egy elkódolása

Feltehető, hogy $\Sigma = \{0, 1\}$.

A Turing gépek egy elkódolása

Feltehető, hogy $\Sigma = \{0,1\}$.

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

A Turing gépek egy elkódolása

Feltehető, hogy $\Sigma = \{0, 1\}$.

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

A Turing gépek egy elkódolása

Feltehető, hogy $\Sigma = \{0, 1\}$.

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.

A Turing gépek egy elkódolása

Feltehető, hogy $\Sigma = \{0, 1\}$.

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.

A Turing gépek egy elkódolása

Feltehető, hogy $\Sigma = \{0,1\}$.

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 $0^i10^j10^r10^s10^t$.
- $\langle M \rangle$ az átmenetek kódjainak felsorolása 11-el elválasztva.

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

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

Létezik nem Turing-felismerhető nyelv

Jelölés: Minden $i \geqslant 1$-re,

Létezik nem Turing-felismerhető nyelv

Jelölés: Minden $i \geqslant 1$-re,

- jelölje w_i a $\{0,1\}^*$ halmaz i-ik elemét a hossz-lexikografikus rendezés szerint.

Létezik nem Turing-felismerhető nyelv

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)

Létezik nem Turing-felismerhető nyelv

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)

Tétel

Létezik nem Turing-felismerhető nyelv.

Létezik nem Turing-felismerhető nyelv

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)

Tétel

Létezik nem Turing-felismerhető nyelv.

Bizonyítás: Két különböző nyelvet nem ismerhet fel ugyanaz a TG. A TG-ek számossága megszámlálható (a fenti kódolás injekció $\{0, 1\}^*$-ba, ami megszámlálható). Másrészt viszont a $\{0, 1\}$ feletti nyelvek számossága continuum. $\square$

Létezik nem Turing-felismerhető nyelv

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)

Tétel

Létezik nem Turing-felismerhető nyelv.

Bizonyítás: Két különböző nyelvet nem ismerhet fel ugyanaz a TG. A TG-ek számossága megszámlálható (a fenti kódolás injekció $\{0,1\}^*$-ba, ami megszámlálható). Másrészt viszont a $\{0,1\}$ feletti nyelvek számossága continuum. $\square$

Az **átlós nyelv**: $L_d = \{w_i \mid w_i \notin L(M_i)\}$

Tétel

$L_d \notin RE$.

R és $\mathcal{L}_1$ viszonya

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 $\rhd(\Sigma \setminus \{\rhd, \lhd\})^* \lhd$-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

R és $\mathcal{L}_1$ viszonya

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 olyan NTG, amely korlátos munkaterülettel rendelkezik.

R és $\mathcal{L}_1$ viszonya

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 olyan NTG, amely korlátos munkaterülettel rendelkezik.

Nevét egy vele ekvivalens modellről kapta, amelyben a rendelkezésre álló tár az input hosszának konstansszorosa (lineáris függvénye).

R és $\mathcal{L}_1$ viszonya

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 $L(G) = L(A)$.

R és $\mathcal{L}_1$ viszonya

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 előadáson láttuk, hogy minden 0. típusú grammatikához lehet konstruálni $L(G)$-t felismerő NTG-t.

R és $\mathcal{L}_1$ viszonya

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 előadáson 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 az 1. és 3. szalag tartalma megegyezik-e.

R és $\mathcal{L}_1$ viszonya

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 előadáson 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 az 1. és 3. szalag tartalma megegyezik-e.

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

R és $\mathcal{L}_1$ viszonya

(2) Az előző előadás konstrukciójának (eléggé technikai jellegű) kis módosításával elérhető, hogy az LKA-hoz (ott NTG-hez) készített grammatika hossz-nemcsökkentő legyen, ehhez viszont ismert, hogy $\exists$ vele ekvivalens 1-típusú grammatika. $\square$

R és $\mathcal{L}_1$ viszonya

(2) Az előző előadás konstrukciójának (eléggé technikai jellegű) kis módosításával elérhető, hogy az LKA-hoz (ott NTG-hez) készített grammatika hossz-nemcsökkentő legyen, ehhez viszont ismert, hogy $\exists$ vele ekvivalens 1-típusú grammatika. $\square$

Tétel

Ha A LKA, akkor $L(A)$ eldönthető.

R és $\mathcal{L}_1$ viszonya

(2) Az előző előadás konstrukciójának (eléggé technikai jellegű) kis módosításával elérhető, hogy az LKA-hoz (ott NTG-hez) készített grammatika hossz-nemcsökkentő legyen, ehhez viszont ismert, hogy $\exists$ vele ekvivalens 1-típusú grammatika. $\square$

Tétel

Ha A LKA, akkor $L(A)$ eldönthető.

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

R és $\mathcal{L}_1$ viszonya

(2) Az előző előadás konstrukciójának (eléggé technikai jellegű) kis módosításával elérhető, hogy az LKA-hoz (ott NTG-hez) készített grammatika hossz-nemcsökkentő legyen, ehhez viszont ismert, hogy $\exists$ vele ekvivalens 1-típusú grammatika. $\square$

Tétel

Ha A LKA, akkor $L(A)$ eldönthető.

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

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

R és $\mathcal{L}_1$ viszonya

Tétel

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

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq$ R.

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M\rangle \mid M \text{ LKA és } \langle M\rangle \notin L(M)\}$.

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M \rangle \mid M \text{ LKA és } \langle M \rangle \notin L(M)\}$.

- $L_{d,\mathrm{LKA}}$ eldönthető.

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M\rangle \mid M \text{ LKA és } \langle M\rangle \notin L(M)\}$.

- $L_{d,\mathrm{LKA}}$ eldönthető.

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

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M\rangle \mid M \text{ LKA és } \langle M\rangle \notin L(M)\}$.

- $L_{d,\mathrm{LKA}}$ eldönthető.

  Egy S TG ugyanis egy M LKA bemenetére menjen q_i-be, ha $\langle M\rangle \notin L(M)$ illetve menjen q_n-be, ha $\langle M\rangle \in L(M)$. Mivel $L(M)$ eldönthető ezért S mindig terminál.
- $L_{d,\mathrm{LKA}}$ felismerhetetlen LKA-val ($\Rightarrow \notin \mathcal{L}_1$)

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M\rangle \mid M \text{ LKA és } \langle M\rangle \notin L(M)\}$.

- $L_{d,\mathrm{LKA}}$ eldönthető.

  Egy S TG ugyanis egy M LKA bemenetére menjen q_i-be, ha $\langle M\rangle \notin L(M)$ illetve menjen q_n-be, ha $\langle M\rangle \in L(M)$. Mivel $L(M)$ eldönthető ezért S mindig terminál.
- $L_{d,\mathrm{LKA}}$ felismerhetetlen LKA-val ($\Rightarrow \notin \mathcal{L}_1$)

  (Cantor féle átlós módszerrel)

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M\rangle \mid M \text{ LKA és } \langle M\rangle \notin L(M)\}$.

- $L_{d,\mathrm{LKA}}$ eldönthető.

  Egy S TG ugyanis egy M LKA bemenetére menjen q_i-be, ha $\langle M\rangle \notin L(M)$ illetve menjen q_n-be, ha $\langle M\rangle \in L(M)$. Mivel $L(M)$ eldönthető ezért S mindig terminál.
- $L_{d,\mathrm{LKA}}$ felismerhetetlen LKA-val ($\Rightarrow \notin \mathcal{L}_1$)

  (Cantor féle átlós módszerrel)

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

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

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

- $L_{d,\text{LKA}}$ eldönthető.

  Egy S TG ugyanis egy M LKA bemenetére menjen q_i-be, ha $\langle M\rangle \notin L(M)$ illetve menjen q_n-be, ha $\langle M\rangle \in L(M)$. Mivel $L(M)$ eldönthető ezért S mindig terminál.
- $L_{d,\text{LKA}}$ felismerhetetlen LKA-val ($\Rightarrow \notin \mathcal{L}_1$)

  (Cantor féle átlós módszerrel)

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

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

R és $\mathcal{L}_1$ viszonya

Tétel

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

Bizonyítás: Az előző tételek miatt $\mathcal{L}_1 \subseteq \mathrm{R}$.

Legyen $L_{d,\mathrm{LKA}} = \{\langle M\rangle \mid M \text{ LKA és } \langle M\rangle \notin L(M)\}$.

- $L_{d,\mathrm{LKA}}$ eldönthető.

  Egy S TG ugyanis egy M LKA bemenetére menjen q_i-be, ha $\langle M\rangle \notin L(M)$ illetve menjen q_n-be, ha $\langle M\rangle \in L(M)$. Mivel $L(M)$ eldönthető ezért S mindig terminál.
- $L_{d,\mathrm{LKA}}$ felismerhetetlen LKA-val ($\Rightarrow \notin \mathcal{L}_1$)

  (Cantor féle átlós módszerrel)

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

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

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

Számítási feladatok megoldása TG-pel

Az eldöntési problémák általánosításai a (ki)számítási problémák. Ilyenkor a kiszámítandó f függvény igen/nem helyett más értékeket is adhat eredményül. Ezúttal is algoritmikus megoldást keresünk.

Számítási feladatok megoldása TG-pel

Az eldöntési problémák általánosításai a (ki)számítási problémák. Ilyenkor a kiszámítandó f függvény igen/nem helyett más értékeket is adhat eredményül. Ezúttal is algoritmikus megoldást keresünk.

Legyen $\text{Dom}(f) = \Sigma^*$, $\text{Ran}\ (f) \subseteq \Delta^*$ valamely Σ, Δ ábécékre.

Számítási feladatok megoldása TG-pel

Az eldöntési problémák általánosításai a (ki)számítási problémák. Ilyenkor a kiszámítandó f függvény igen/nem helyett más értékeket is adhat eredményül. Ezúttal is algoritmikus megoldást keresünk.

Legyen $\mathrm{Dom}(f) = \Sigma^*$, $\mathrm{Ran}\ (f) \subseteq \Delta^*$ 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.

Számítási feladatok megoldása TG-pel

Az eldöntési problémák általánosításai a (ki)számítási problémák. Ilyenkor a kiszámítandó f függvény igen/nem helyett más értékeket is adhat eredményül. Ezúttal is algoritmikus megoldást keresünk.

Legyen $\mathrm{Dom}(f) = \Sigma^*$, $\mathrm{Ran}\ (f) \subseteq \Delta^*$ 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.] Az ilyen TG-eket **számító Turing gépnek** nevezzük.

Számítási feladatok megoldása TG-pel

Az eldöntési problémák általánosításai a (ki)számítási problémák. Ilyenkor a kiszámítandó f függvény igen/nem helyett más értékeket is adhat eredményül. Ezúttal is algoritmikus megoldást keresünk.

Legyen $\mathrm{Dom}(f) = \Sigma^*$, $\mathrm{Ran}\ (f) \subseteq \Delta^*$ 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.] Az ilyen TG-eket **számító Turing gépnek** nevezzük.

Példa:

$f(u) = ub$

$(u \in \{a, b\}^*)$.

Számítási feladatok megoldása TG-pel

Az eldöntési problémák általánosításai a (ki)számítási problémák. Ilyenkor a kiszámítandó f függvény igen/nem helyett más értékeket is adhat eredményül. Ezúttal is algoritmikus megoldást keresünk.

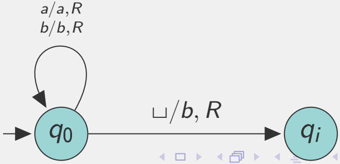
Legyen $\mathrm{Dom}(f) = \Sigma^*$, $\mathrm{Ran}\ (f) \subseteq \Delta^*$ valamely Σ, Δ ábécékre.

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.] Az ilyen TG-eket **számító Turing gépnek** nevezzük.

Példa:
$f(u) = ub$
$(u \in \{a, b\}^*)$.



Visszavezetés

Definíció

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

Visszavezetés

Definíció

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

Definíció

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

Visszavezetés

Definíció

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

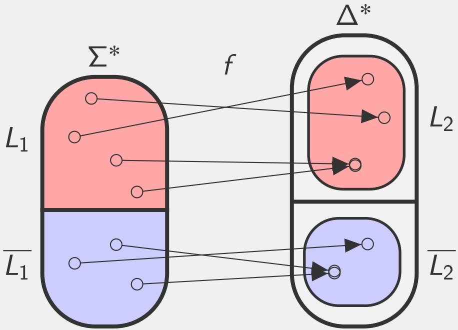
Definíció

$L_1 \subseteq \Sigma^*$ **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 szakirodalomban: many-one reducibility.

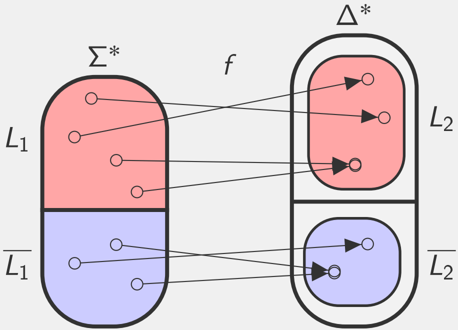
Visszavezetés

$L_1 \leqslant L_2$



Visszavezetés

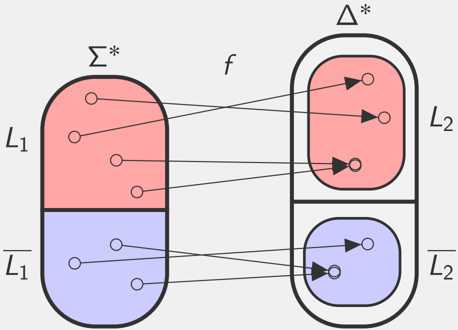
$L_1 \leqslant L_2$



(1) f az egész Σ^*-on értelmezett, (2) f kiszámítható, (3) $f(L_1) \subseteq L_2$ valamint (4) $f(\overline{L_1}) \subseteq \overline{L_2}$.

Visszavezetés

$L_1 \leqslant L_2$



(1) f az egész Σ^*-on értelmezett, (2) f kiszámítható, (3) $f(L_1) \subseteq L_2$ valamint (4) $f(\overline{L_1}) \subseteq \overline{L_2}$.

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

Visszavezetés

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

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 $L_1 \in R$.

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 $L_2 \notin R$.

R és RE

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

R és RE

Univerzális nyelv: $L_{\mathrm{u}} = \{\langle M, w\rangle \,|\, w \in L(M)\}$.

Tétel

$L_u \in \mathrm{RE} \backslash \mathrm{R}$

R és RE

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

Tétel

$L_u \in \mathrm{RE} \backslash \mathrm{R}$

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

R és RE

Univerzális nyelv: $L_\mathrm{u} = \{\langle M, w\rangle \,|\, w \in L(M)\}$.

Tétel

$L_u \in$ RE$\setminus$R

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

Tétel

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

R és RE

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

Tétel

$L_u \in \mathrm{RE} \setminus \mathrm{R}$

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

Tétel

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

Következmény: RE nem zárt a komplementer-képzésre.

R és RE

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

Tétel

$L_u \in$ RE\R

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

Tétel

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

Következmény: RE nem zárt a komplementer-képzésre.

Tétel

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

R és RE

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

Tétel

$L_u \in$ RE$\setminus$R

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

Tétel

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

Következmény: RE nem zárt a komplementer-képzésre.

Tétel

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

Megállási probléma: $L_\mathrm{halt} = \{\langle M, w\rangle \mid M \text{ megáll } w\text{-n}\}$.

R és RE

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

Tétel

$L_u \in$ RE$\setminus$R

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

Tétel

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

Következmény: RE nem zárt a komplementer-képzésre.

Tétel

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

Megállási probléma: $L_{\mathrm{halt}} = \{\langle M, w\rangle \mid M$ megáll w-n$\}$.

Tétel

$L_{\mathrm{halt}} \in$ RE$\setminus$R.

Eldönthetetlen problémák

Grammatikák:

Tétel

Eldönthetetlenek az alábbi CF nyelvtanokkal kapcsolatos kérdések. Legyen G_1 és G_2 két CF nyelvtan.

- $L(G_1) \cap L(G_2) \neq \emptyset$
- $L(G_1) = \Gamma^*$ valamely Γ-ra
- $L(G_1) = L(G_2)$
- $L(G_1) \subseteq L(G_2)$

Eldönthetetlen problémák

Grammatikák:

Tétel

Eldönthetetlenek az alábbi CF nyelvtanokkal kapcsolatos kérdések. Legyen G_1 és G_2 két CF nyelvtan.

- $L(G_1) \cap L(G_2) \neq \emptyset$
- $L(G_1) = \Gamma^*$ valamely Γ-ra
- $L(G_1) = L(G_2)$
- $L(G_1) \subseteq L(G_2)$

Megjegyzés: Reguláris nyelvekre ezek a kérdések eldönthetők.

Eldönthetetlen problémák

Grammatikák:

Tétel

Eldönthetetlenek az alábbi CF nyelvtanokkal kapcsolatos kérdések. Legyen G_1 és G_2 két CF nyelvtan.

- $L(G_1) \cap L(G_2) \neq \emptyset$
- $L(G_1) = \Gamma^*$ valamely Γ-ra
- $L(G_1) = L(G_2)$
- $L(G_1) \subseteq L(G_2)$

Megjegyzés: Reguláris nyelvekre ezek a kérdések eldönthetők.

Megjegyzés: $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ esetén is eldönthető a szóprobléma. Input: G és $u \in T^*$. Output: $u \stackrel{?}{\in} L(G)$.

Eldönthetetlen problémák

Logikai kérdések:

Tétel

Eldönthetetlen, hogy φ elsőrendű logikai formulára és Φ formulahalmazra

(1) $\models \varphi$ teljesül-e (φ logikailag igaz-e).

(2) φ kielégíthetetlen-e

(3) φ kielégíthető-e

(4) $\Phi \models \varphi$ teljesül-e

Eldönthetetlen problémák

Logikai kérdések:

Tétel

Eldönthetetlen, hogy φ elsőrendű logikai formulára és Φ formulahalmazra

(1) $\models \varphi$ teljesül-e (φ logikailag igaz-e).
(2) φ kielégíthetetlen-e
(3) φ kielégíthető-e
(4) $\Phi \models \varphi$ teljesül-e

Eldönthető, hogy φ nulladrendű logikai formulára és Φ formulahalmazra

(1) $\models \varphi$ teljesül-e (φ tautologikusan igaz-e).
(2) φ kielégíthetetlen-e
(3) φ kielégíthető-e
(4) $\Phi \models \varphi$ teljesül-e

Bonyolultságelmélet – időbonyolultsági osztályok

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

Bonyolultságelmélet – időbonyolultsági osztályok

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

- $\mathrm{TIME}\,(f(n)) =$
  $\{L \mid L \text{ eldönthető } O(f(n)) \text{ időigényű determinisztikus TG-pel}\}$

Bonyolultságelmélet – időbonyolultsági osztályok

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

- $\mathrm{TIME}\,(f(n)) =$
  $\{L \mid L$ eldönthető $O(f(n))$ időigényű determinisztikus TG-pel$\}$
- $\mathrm{NTIME}\,(f(n)) =$
  $\{L \mid L$ eldönthető $O(f(n))$ időigényű nemdeterminisztikus TG-pel$\}$

Bonyolultságelmélet – időbonyolultsági osztályok

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

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

Bonyolultságelmélet – időbonyolultsági osztályok

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

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

Bonyolultságelmélet – időbonyolultsági osztályok

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

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

Bonyolultságelmélet – időbonyolultsági osztályok

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

- $\mathrm{TIME}(f(n)) =$
  $\{L \mid L$ eldönthető $O(f(n))$ időigényű determinisztikus TG-pel$\}$
- $\mathrm{NTIME}(f(n)) =$
  $\{L \mid L$ eldönthető $O(f(n))$ időigényű nemdeterminisztikus TG-pel$\}$
- $\mathrm{P}=\bigcup_{k\geqslant 1}\mathrm{TIME}(n^k)$.
- $\mathrm{NP}=\bigcup_{k\geqslant 1}\mathrm{NTIME}(n^k)$.
- Észrevétel: $\mathrm{P}\subseteq\mathrm{NP}$.
- Sejtés: $\mathrm{P}\neq\mathrm{NP}$ (sejtjük, hogy igaz, de bizonyítani nem tudjuk).

Bonyolultságelmélet – időbonyolultsági osztályok

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

- $\mathrm{TIME}(f(n)) =$
  $\{L \mid L$ eldönthető $O(f(n))$ időigényű determinisztikus TG-pel$\}$
- $\mathrm{NTIME}(f(n)) =$
  $\{L \mid L$ eldönthető $O(f(n))$ időigényű nemdeterminisztikus TG-pel$\}$
- $\mathrm{P}=\bigcup_{k\geqslant 1}\mathrm{TIME}(n^k)$.
- $\mathrm{NP}=\bigcup_{k\geqslant 1}\mathrm{NTIME}(n^k)$.
- Észrevétel: $\mathrm{P} \subseteq \mathrm{NP}$.
- Sejtés: $\mathrm{P} \neq \mathrm{NP}$ (sejtjük, hogy igaz, de bizonyítani nem tudjuk).
- A $\mathrm{P} \neq \mathrm{NP}$ sejtés a Clay Matematikai Intézet által 2000-ben nyilvánosságra hozott 7 Milleneumi Probléma egyike. Igazolásáért vagy cáfolatáért az Intézet 1M$-t fizet.

Milyenek az NP-beli problémák?

P-re úgy gondolunk, hogy ez az osztály tartalmazza a hatékonyan megoldható problémákat. (Ami nem teljesen igaz.)

Milyenek az NP-beli problémák?

P-re úgy gondolunk, hogy ez az osztály tartalmazza a hatékonyan megoldható problémákat. (Ami nem teljesen igaz.)

Milyen problémákat tartalmaz NP?

Milyenek az NP-beli problémák?

P-re úgy gondolunk, hogy ez az osztály tartalmazza a hatékonyan megoldható problémákat. (Ami nem teljesen igaz.)

Milyen problémákat tartalmaz NP?

Egy L NP-beli problémához definíció szerint létezik őt polinom időben eldöntő NTG ami gyakorlatilag a következőképpen működik:

a probléma minden bemenetére próbál polinom időben **„megsejteni”** (értsd: nemdeterminisztikusan előllítani) egy kis méretű „tanút”, ami azt bizonyítja, hogy a bemenet egy igen-példány.

Milyenek az NP-beli problémák?

P-re úgy gondolunk, hogy ez az osztály tartalmazza a hatékonyan megoldható problémákat. (Ami nem teljesen igaz.)

Milyen problémákat tartalmaz NP?

Egy L NP-beli problémához definíció szerint létezik őt polinom időben eldöntő NTG ami gyakorlatilag a következőképpen működik:

a probléma minden bemenetére próbál polinom időben **„megsejteni”** (értsd: nemdeterminisztikusan előllítani) egy kis méretű „tanút”, ami azt bizonyítja, hogy a bemenet egy igen-példány.

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).

Milyenek az NP-beli problémák?

P-re úgy gondolunk, hogy ez az osztály tartalmazza a hatékonyan megoldható problémákat. (Ami nem teljesen igaz.)

Milyen problémákat tartalmaz NP?

Egy L NP-beli problémához definíció szerint létezik őt polinom időben eldöntő NTG ami gyakorlatilag a következőképpen működik:

a probléma minden bemenetére próbál polinom időben **„megsejteni”** (értsd: nemdeterminisztikusan előllítani) egy kis méretű „tanút”, ami azt bizonyítja, hogy a bemenet egy igen-példány.

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).

Polinom idejű visszavezetés

Definíció

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

Polinom idejű visszavezetés

Definíció

Az $f : \Sigma^* \to \Delta^*$ szófüggvény **polinom időben kiszámítható**, ha van olyan Turing-gép, ami polinom időben 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: $L_1 \leqslant_p L_2$.

Polinom idejű visszavezetés

Definíció

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

$\mathcal{C}$-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.

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

$\mathcal{C}$-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.

$\mathcal{C}$-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.

NP-teljesség

Ha speciálisan $\mathcal{C}$=NP:

Definíció

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

- $L \in \mathrm{NP}$
- L NP-nehéz, azaz minden $L' \in \mathrm{NP}$ esetén $L' \leqslant_p L$.

NP-teljesség

Ha speciálisan $\mathcal{C}$=NP:

Definíció

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

- $L \in \mathrm{NP}$
- L NP-nehéz, azaz minden $L' \in \mathrm{NP}$ esetén $L' \leqslant_p L$.

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

NP-teljesség

Ha speciálisan $\mathcal{C}$=NP:

Definíció

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

- $L \in \mathrm{NP}$
- L NP-nehéz, azaz minden $L' \in \mathrm{NP}$ esetén $L' \leqslant_p L$.

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

Tétel

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

NP-teljesség

Ha speciálisan $\mathcal{C}$=NP:

Definíció

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

- $L \in \mathrm{NP}$
- L NP-nehéz, azaz minden $L' \in \mathrm{NP}$ esetén $L' \leqslant_p L$.

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

Tétel

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

NP-teljesség és a P$\stackrel{?}{=}$NP probléma

Intuitive az NP-teljes problémák az NP-beli problémák legnehezebbjei.

NP-teljesség és a $P \stackrel{?}{=} NP$ probléma

Intuitive az NP-teljes problémák 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.

NP-teljesség és a P$\stackrel{?}{=}$NP probléma

Intuitive az NP-teljes problémák 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.

Mivel nem tudjuk, hogy P$\stackrel{?}{=}$NP (azt sejtjük, hogy nem igaz!), ezért nyilván egyetlen NP-teljes problémára sem ismeretes polinomiális idejű determinisztikus algoritmus és amennyiben a sejtésünk igaz, ilyet nem is fogunk találni.

NP-teljesség és a P$\stackrel{?}{=}$NP probléma

Intuitive az NP-teljes problémák 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.

Mivel nem tudjuk, hogy P$\stackrel{?}{=}$NP (azt sejtjük, hogy nem igaz!), ezért nyilván egyetlen NP-teljes problémára sem ismeretes polinomiális idejű determinisztikus algoritmus és amennyiben a sejtésünk igaz, ilyet nem is fogunk találni.

Így az NP-teljes problémákra úgy tekinthetünk, mint eldönthető, de **hatékonyan nem eldönthető** problémákra.

NP-teljesség és a P$\stackrel{?}{=}$NP probléma

Intuitive az NP-teljes problémák 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.

Mivel nem tudjuk, hogy P$\stackrel{?}{=}$NP (azt sejtjük, hogy nem igaz!), ezért nyilván egyetlen NP-teljes problémára sem ismeretes polinomiális idejű determinisztikus algoritmus és amennyiben a sejtésünk igaz, ilyet nem is fogunk találni.

Így az NP-teljes problémákra úgy tekinthetünk, mint eldönthető, de **hatékonyan nem eldönthető** problémákra.

Cook-Levin tétel

SAT$= \{\langle\varphi\rangle \,|\, \varphi$ kielégíthető nulladrendű KNF$\}$

Cook-Levin tétel

SAT$= \{\langle\varphi\rangle \,|\, \varphi$ kielégíthető nulladrendű KNF$\}$

Tétel (Cook-Levin)

SAT NP-teljes.

Cook-Levin tétel

SAT$= \{\langle\varphi\rangle \,|\, \varphi$ kielégíthető nulladrendű KNF$\}$

Tétel (Cook-Levin)

SAT NP-teljes.

Tétel

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

Cook-Levin tétel

SAT$= \{\langle\varphi\rangle \,|\, \varphi$ kielégíthető nulladrendű KNF$\}$

Tétel (Cook-Levin)

SAT NP-teljes.

Tétel

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

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

Cook-Levin tétel

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

Tétel (Cook-Levin)

SAT NP-teljes.

Tétel

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

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

A következő problémák NP-teljesség ezen tétel alapján bizonyítható.

kSAT

Egy minden tagjában pontosan k különböző literált tartalmazó konjunktív normál formát (KNF-et) **kKNF-nek** nevezünk ($k \geqslant 1$).

kSAT

Egy minden tagjában pontosan k különböző literált tartalmazó konjunktív normál formát (KNF-et) **kKNF-nek** nevezünk ($k \geqslant 1$).

Példák: 4KNF:

$(\neg X_1 \vee X_3 \vee X_5 \vee \neg X_6) \wedge (\neg X_1 \vee \neg X_3 \vee X_4 \vee \neg X_6) \wedge (X_1 \vee X_2 \vee \neg X_4 \vee \neg X_6)$.

kSAT

Egy minden tagjában pontosan k különböző literált tartalmazó konjunktív normál formát (KNF-et) **kKNF-nek** nevezünk ($k \geqslant 1$).

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: $(\neg X_1 \vee X_3) \wedge (\neg X_1 \vee \neg X_3) \wedge (X_1 \vee X_2) \wedge (\neg X_2 \vee X_3)$.

kSAT

Egy minden tagjában pontosan k különböző literált tartalmazó konjunktív normál formát (KNF-et) **kKNF-nek** nevezünk ($k \geqslant 1$).

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: $(\neg X_1 \vee X_3) \wedge (\neg X_1 \vee \neg X_3) \wedge (X_1 \vee X_2) \wedge (\neg X_2 \vee X_3)$.

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

kSAT

Egy minden tagjában pontosan k különböző literált tartalmazó konjunktív normál formát (KNF-et) ***k*KNF-nek** nevezünk ($k \geqslant 1$).

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: $(\neg X_1 \vee X_3) \wedge (\neg X_1 \vee \neg X_3) \wedge (X_1 \vee X_2) \wedge (\neg X_2 \vee X_3)$.

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

Tétel

3SAT NP-teljes.

Megjegyzés: 2SAT $\in$ P.

3 színezhetőség

Definíció

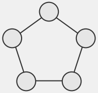
Legyen $k \geqslant 1$ egész szám. Egy gráf **k-színezhető**, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

3 színezhetőség

Definíció

Legyen $k \geqslant 1$ egész szám. Egy gráf *k-**színezhető***, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

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



3 színezhetőség

Definíció

Legyen $k \geqslant 1$ egész szám. Egy gráf ***k*-színezhető**, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

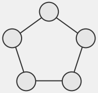
Példa: Egy 5 csúcsból álló kör 3-színezhető, de nem 2-színezhető.

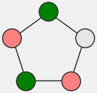
3 színezhetőség

Definíció

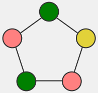
Legyen $k \geqslant 1$ egész szám. Egy gráf ***k*-színezhető**, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

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





???



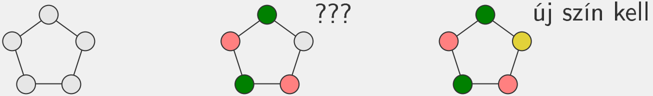
új szín kell

3 színezhetőség

Definíció

Legyen $k \geqslant 1$ egész szám. Egy gráf ***k*-színezhető**, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

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



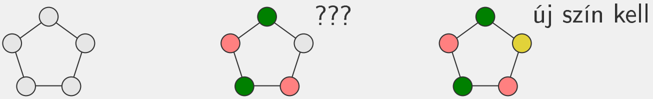
$k\text{SZÍNEZÉS} = \{\langle G\rangle \mid G\ k\text{-színezhető}\}$.

3 színezhetőség

Definíció

Legyen $k \geqslant 1$ egész szám. Egy gráf *k*-**színezhető**, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

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



$k\textsc{Színezés}=\{\langle G\rangle \mid G \ k\text{-színezhető}\}$.

Tétel

$3\textsc{Színezés}$ NP-teljes.

Megjegyzés: $2\textsc{Színezés} \in \mathrm{P}$.

Klikk, független ponthalmaz

Definíció

A G egyszerű, irányítatlan gráf egy teljes részgráfját **klikknek**, egy üres részgráfját **független ponthalmaznak** mondjuk.

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**.

Klikk, független ponthalmaz

Definíció

A G egyszerű, irányítatlan gráf egy teljes részgráfját **klikknek**, egy üres részgráfját **független ponthalmaznak** mondjuk.

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**.

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

Klikk, független ponthalmaz

Definíció

A G egyszerű, irányítatlan gráf egy teljes részgráfját **klikknek**, egy üres részgráfját **független ponthalmaznak** mondjuk.

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**.

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

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

Klikk, független ponthalmaz

Definíció

A G egyszerű, irányítatlan gráf egy teljes részgráfját **klikknek**, egy üres részgráfját **független ponthalmaznak** mondjuk.

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**.

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

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

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

Klikk, független ponthalmaz

Definíció

A G egyszerű, irányítatlan gráf egy teljes részgráfját **klikknek**, egy üres részgráfját **független ponthalmaznak** mondjuk.

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**.

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

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

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

Tétel

Független ponthalmaz,Klikk, Lefogó ponthalmaz NP-teljes.

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.

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.

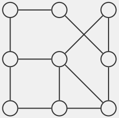
Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.

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.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



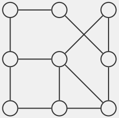
G

Irányítatlan/irányított Hamilton út/kör

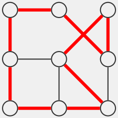
Definíció

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.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



G



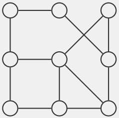
H-kör G-ben

Irányítatlan/irányított Hamilton út/kör

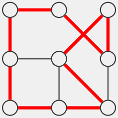
Definíció

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.

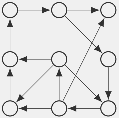
Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



G



H-kör G-ben



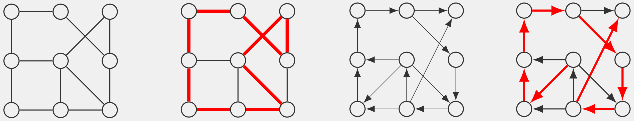
G'

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.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



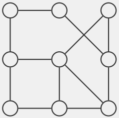
G H-kör G-ben G' H-út G'-ben

Irányítatlan/irányított Hamilton út/kör

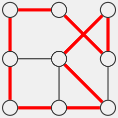
Definíció

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.

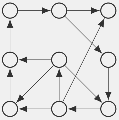
Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



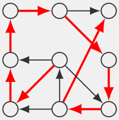
G



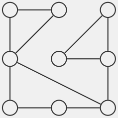
H-kör G-ben



G'



H-út G'-ben



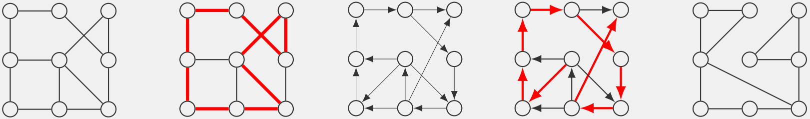
nincs H-kör

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.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



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

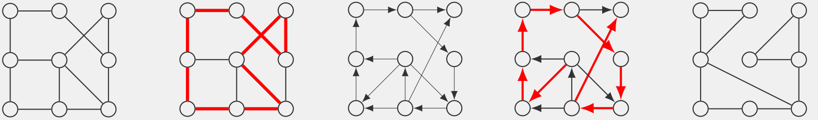
HÚ=$\{\langle G, s, t\rangle \mid$ van a G irányított gráfban s-ből t-be H-út$\}$.

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.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



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$\}$.

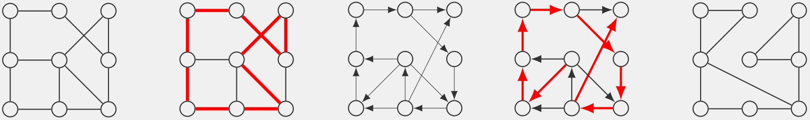
IHÚ=$\{\langle G,s,t\rangle\,|\,$van a G irányítatlan gráfban s-ből t-be H-út$\}$.

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.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.



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

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

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

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

Hamilton út problémák NP teljessége

Tétel

HÚ, IHÚ, IHK NP-teljes

Hamilton út problémák NP teljessége

Tétel

HÚ, IHÚ, IHK NP-teljes

Az utazóügynök probléma:

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

Hamilton út problémák NP teljessége

Tétel

HÚ, IHÚ, IHK NP-teljes

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$\}$.

Hamilton út problémák NP teljessége

Tétel

HÚ, IHÚ, IHK NP-teljes

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

Hamilton út problémák NP teljessége

Tétel

HÚ, IHÚ, IHK NP-teljes

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

További NP-teljes problémák

LINEÁRIS DIOPHANTOSZI EGYENLŐTLENSÉGRENDSZER $=$

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

További NP-teljes problémák

LINEÁRIS DIOPHANTOSZI EGYENLŐTLENSÉGRENDSZER= $\{\langle \mathbf{A}, \mathbf{b} \rangle \mid \mathbf{Ax} \leqslant \mathbf{b}$ egészegyütthatós egyenlőtlenségrendszernek van egész megoldása$\}$.

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 $K\}$

További NP-teljes problémák

Lineáris diophantoszi egyenlőtlenségrendszer$=$
$\{\langle \mathbf{A}, \mathbf{b} \rangle \mid \mathbf{Ax} \leqslant \mathbf{b}$ egészegyütthatós egyenlőtlenségrendszernek van egész megoldása$\}$.

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 $K\}$

Hátizsák$:= \{\langle a_1, \ldots, a_n, b, p_1, \ldots p_n, k \rangle \in (\mathbb{R}^+)^{2n+2} \mid$
$\exists\ I \subseteq \{1, \ldots n\}$, amelyre $\sum_{i \in I} a_i \leqslant b$ és $\sum_{i \in I} p_i \geqslant k\}$.

További NP-teljes problémák

Lineáris diophantoszi egyenlőtlenségrendszer=
$\{\langle \mathbf{A}, \mathbf{b}\rangle \mid \mathbf{Ax} \leqslant \mathbf{b}$ egészegyütthatós egyenlőtlenségrendszernek van egész megoldása$\}$.

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 $K\}$

Hátizsák:= $\{\langle a_1, \ldots, a_n, b, p_1, \ldots p_n, k\rangle \in (\mathbb{R}^+)^{2n+2} \mid \exists\ I \subseteq \{1, \ldots n\}$, amelyre $\sum_{i \in I} a_i \leqslant b$ és $\sum_{i \in I} p_i \geqslant k\}$.

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

További NP-teljes problémák

LINEÁRIS DIOPHANTOSZI EGYENLŐTLENSÉGRENDSZER= $\{\langle \mathbf{A}, \mathbf{b} \rangle \mid \mathbf{Ax} \leqslant \mathbf{b}$ egészegyütthatós egyenlőtlenségrendszernek van egész megoldása$\}$.

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 $K\}$

HÁTIZSÁK:= $\{\langle a_1, \ldots, a_n, b, p_1, \ldots p_n, k \rangle \in (\mathbb{R}^+)^{2n+2} \mid \exists\ I \subseteq \{1, \ldots n\}$, amelyre $\sum_{i \in I} a_i \leqslant b$ és $\sum_{i \in I} p_i \geqslant k\}$.

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

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 partícióban a súlyok összege $\leqslant 1\}$.

NP lehetséges szerkezete

NP-köztes nyelv

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.

NP lehetséges szerkezete

NP-köztes nyelv

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.

Az alábbi problémáknak se a P-belisége, se NP-nehézsége nem ismeretes (így NP-köztes jelöltek):

NP lehetséges szerkezete

NP-köztes nyelv

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.

Az alábbi problémáknak se a P-belisége, se NP-nehézsége nem ismeretes (így NP-köztes jelöltek):

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

NP lehetséges szerkezete

NP-köztes nyelv

L NP-köztes, ha $L \in$ NP, $L \notin$ P és L nem NP-teljes.

Ladner tétele

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

Az alábbi problémáknak se a P-belisége, se NP-nehézsége nem ismeretes (így NP-köztes jelöltek):

- Gráfizomorfizmus$=\{\langle G_1, G_2\rangle \mid G_1$ és G_2 irányítatlan izomorf gráfok$\}$.
- Prímfaktorizáció: adjuk meg egy egész szám prímtényezős felbontását [számítási feladat],
- Kapuminimalizálás: adott digitális áramkört minél kevesebb logikai kapuval megvalósítani [számítási feladat].

Tárbonyolultság – Az offline Turing gép

A tárbonyolultság mérésének problémája:

Első megközelítésben a tárigény a működés során felhasznált (vagyis a fejek által meglátogatott) cellák száma.

Tárbonyolultság – Az offline Turing gép

A tárbonyolultság mérésének problémája:

Első megközelítésben a tárigény a működés során felhasznált (vagyis a fejek által meglátogatott) cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

Tárbonyolultság – Az offline Turing gép

A tárbonyolultság mérésének problémája:

Első megközelítésben a tárigény a működés során felhasznált (vagyis a fejek által meglátogatott) cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

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.
A **nemdeterminisztikus offline Turing gép** (NOTG) ugyanilyen, csak a gép nemdeterminisztikus.

Tárbonyolultság – Az offline Turing gép

A tárbonyolultság mérésének problémája:

Első megközelítésben a tárigény a működés során felhasznált (vagyis a fejek által meglátogatott) cellák száma.

Probléma: Hiába "takarékoskodik" a felhasznált cellákkal a gép, az input hossza így mindig alsó korlát lesz a tárigényre.

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.
A **nemdeterminisztikus offline Turing gép** (NOTG) ugyanilyen, csak a gép nemdeterminisztikus.

Állítás

Minden TG-hez megadható vele ekvivalens offline TG.

Az offline Turing gépek tárigénye

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító TG, melynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Első szalagját bemeneti, utolsó szalagját kimeneti, többi szalagját munkaszalagnak nevezzük.

Definíció

Egy OTG **többlet tárigénye** egy adott inputra azon celláknak a száma, melyeken a működés során valamelyik munkaszalag feje járt.

Az offline Turing gépek tárigénye

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító TG, melynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Első szalagját bemeneti, utolsó szalagját kimeneti, többi szalagját munkaszalagnak nevezzük.

Definíció

Egy OTG **többlet tárigénye** egy adott inputra azon celláknak a száma, melyeken a működés során valamelyik munkaszalag feje járt.

Definíció

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ az többlet tárigénye.

Az offline Turing gépek tárigénye

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító TG, melynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Első szalagját bemeneti, utolsó szalagját kimeneti, többi szalagját munkaszalagnak nevezzük.

Definíció

Egy OTG **többlet tárigénye** egy adott inputra azon celláknak a száma, melyeken a működés során valamelyik munkaszalag feje járt.

Definíció

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ az többlet tárigénye.

Számító OTG-re hasonlóan. Nemdeterminisztikus OTG-re (NOTG) értelmszerűen módosítva.

Az offline Turing gépek tárigénye

Definíció

A **számító offline Turing gép** olyan legalább 2 szalagos számító TG, melynek az első szalagja csak olvasható, az utolsó szalagja csak írható. Első szalagját bemeneti, utolsó szalagját kimeneti, többi szalagját munkaszalagnak nevezzük.

Definíció

Egy OTG **többlet tárigénye** egy adott inputra azon celláknak a száma, melyeken a működés során valamelyik munkaszalag feje járt.

Definíció

Egy offline TG $f(n)$ **többlet tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ az többlet tárigénye.

Számító OTG-re hasonlóan. Nemdeterminisztikus OTG-re (NOTG) értelmszerűen módosítva.

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

- $\mathrm{SPACE}(f(n)) := \{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos determinisztikus offline TG-pel$\}$

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

- $\mathrm{SPACE}(f(n)) := \{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos determinisztikus offline TG-pel$\}$
- $\mathrm{NSPACE}(f(n)) :=$
  $\{L \mid L$ eldönthető $O(f(n))$ többlet tárkorlátos nemdeterminisztikus offline TG-pel$\}$

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

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

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

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

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

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

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

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

Determinisztikus és nemdeterminisztikus tárbonyolultsági osztályok

Így az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultságot is mérhetünk.

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

Savitch tétele és a hierarchia tétel

Tétel (Savitch)

Ha $f(n) \geqslant \log n$, akkor $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

Következmény

$\mathrm{PSPACE} = \mathrm{NPSPACE}$

Savitch tétele és a hierarchia tétel

Tétel (Savitch)

Ha $f(n) \geqslant \log n$, akkor $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

Következmény

$\mathrm{PSPACE} = \mathrm{NPSPACE}$

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

Savitch tétele és a hierarchia tétel

Tétel (Savitch)

Ha $f(n) \geqslant \log n$, akkor $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

Következmény

$\mathrm{PSPACE} = \mathrm{NPSPACE}$

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

Hierarchia tétel

$\mathrm{NL} \subset \mathrm{PSPACE}$ és $\mathrm{P} \subset \mathrm{EXPTIME}$.

Savitch tétele és a hierarchia tétel

Tétel (Savitch)

Ha $f(n) \geqslant \log n$, akkor $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

Következmény

PSPACE = NPSPACE

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

Hierarchia tétel

$\mathrm{NL} \subset \mathrm{PSPACE}$ és $\mathrm{P} \subset \mathrm{EXPTIME}$.

$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{NPSPACE}=\mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$

Savitch tétele és a hierarchia tétel

Tétel (Savitch)

Ha $f(n) \geqslant \log n$, akkor $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

Következmény

PSPACE = NPSPACE

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

Hierarchia tétel

NL $\subset$ PSPACE és P $\subset$ EXPTIME.

L $\subseteq$ NL $\subseteq$ P $\subseteq$ NP $\subseteq$ NPSPACE=PSPACE $\subseteq$ EXPTIME

Sejtés: A fenti tartalmazási lánc minden tartalmazása valódi.

Számítási problémák közelítő megoldásai

Jelölje egy kiszámítási (optimalizálási) problémában OPT az optimális értéket (minimumot/maximumot).

Számítási problémák közelítő megoldásai

Jelölje egy kiszámítási (optimalizálási) problémában OPT az optimális értéket (minimumot/maximumot).

Definíció

Egy algoritmust **α-közelítőnek** hívunk, ha minden inputra az algoritmus kimenete megengedett és a visszaadott érték OPT-nak

- minimumkeresési feladat esetén legfeljebb α-szorosa,
- maximumkeresési feladat esetén legalább $1/\alpha$-szorosa,

Számítási problémák közelítő megoldásai

Jelölje egy kiszámítási (optimalizálási) problémában OPT az optimális értéket (minimumot/maximumot).

Definíció

Egy algoritmust α-**közelítőnek** hívunk, ha minden inputra az algoritmus kimenete megengedett és a visszaadott érték OPT-nak

- minimumkeresési feladat esetén legfeljebb α-szorosa,
- maximumkeresési feladat esetén legalább $1/\alpha$-szorosa,

Megjegyzés: α nem feltétlenül konstans, lehet az input *n* hosszának egy függvénye is.

Példa: Irányított gráfban maximális aciklikus részgráf keresése.

Számítási problémák közelítő megoldásai

Jelölje egy kiszámítási (optimalizálási) problémában OPT az optimális értéket (minimumot/maximumot).

Definíció

Egy algoritmust α-**közelítőnek** hívunk, ha minden inputra az algoritmus kimenete megengedett és a visszaadott érték OPT-nak

- minimumkeresési feladat esetén legfeljebb α-szorosa,
- maximumkeresési feladat esetén legalább $1/\alpha$-szorosa,

Megjegyzés: α nem feltétlenül konstans, lehet az input n hosszának egy függvénye is.

Példa: Irányított gráfban maximális aciklikus részgráf keresése.

Rendezzük sorba a csúcsokat.

Számítási problémák közelítő megoldásai

Jelölje egy kiszámítási (optimalizálási) problémában OPT az optimális értéket (minimumot/maximumot).

Definíció

Egy algoritmust α-**közelítőnek** hívunk, ha minden inputra az algoritmus kimenete megengedett és a visszaadott érték OPT-nak

- minimumkeresési feladat esetén legfeljebb α-szorosa,
- maximumkeresési feladat esetén legalább $1/\alpha$-szorosa,

Megjegyzés: α nem feltétlenül konstans, lehet az input n hosszának egy függvénye is.

Példa: Irányított gráfban maximális aciklikus részgráf keresése.

Rendezzük sorba a csúcsokat. A sorrend szerint haladva, minden csúcsra vizsgáljuk meg, hogy előre-élből vagy hátra-élből van-e több, a kisebbséghez tartozó éleket dobjuk el.

Számítási problémák közelítő megoldásai

Jelölje egy kiszámítási (optimalizálási) problémában OPT az optimális értéket (minimumot/maximumot).

Definíció

Egy algoritmust α-**közelítőnek** hívunk, ha minden inputra az algoritmus kimenete megengedett és a visszaadott érték OPT-nak

- minimumkeresési feladat esetén legfeljebb α-szorosa,
- maximumkeresési feladat esetén legalább $1/\alpha$-szorosa,

Megjegyzés: α nem feltétlenül konstans, lehet az input n hosszának egy függvénye is.

Példa: Irányított gráfban maximális aciklikus részgráf keresése.

Rendezzük sorba a csúcsokat. A sorrend szerint haladva, minden csúcsra vizsgáljuk meg, hogy előre-élből vagy hátra-élből van-e több, a kisebbséghez tartozó éleket dobjuk el. A kapott gráf aciklikus és az élek legalább felét tartalmazza, így az algoritmus 2-közelítő. (Itt a megengedett kimenetek az aciklikus részgráfok.)

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Példa: Minimális méretű lefogó ponthalmaz keresése egy G irányítatlan gráfban. A probléma NP-nehéz. Jelölje $\tau(G) = \min\{|S| \mid S \text{ lefogó ponthalmaz } G\text{-ben}\}$.

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Példa: Minimális méretű lefogó ponthalmaz keresése egy G irányítatlan gráfban. A probléma NP-nehéz. Jelölje $\tau(G) = \min\{|S| \mid S \text{ lefogó ponthalmaz } G\text{-ben}\}$.

Megengedett válasz: egy lefogó ponthalmaz.

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Példa: Minimális méretű lefogó ponthalmaz keresése egy G irányítatlan gráfban. A probléma NP-nehéz. Jelölje $\tau(G) = \min\{|S| \mid S \text{ lefogó ponthalmaz } G\text{-ben}\}$.

Megengedett válasz: egy lefogó ponthalmaz.

Mohón vegyük sorban minden, az adott pillanatig fedetlen él mindkét végpontját S-hez, amíg van fedetlen él.

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Példa: Minimális méretű lefogó ponthalmaz keresése egy G irányítatlan gráfban. A probléma NP-nehéz. Jelölje $\tau(G) = \min\{|S| \mid S$ lefogó ponthalmaz G-ben$\}$.

Megengedett válasz: egy lefogó ponthalmaz.

Mohón vegyük sorban minden, az adott pillanatig fedetlen él mindkét végpontját S-hez, amíg van fedetlen él. Az algoritmus során talált fedetlen élek diszjunktak, tehát $|S|/2$ csúcsra szükség van már csak az ő lefogásukhoz is.

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Példa: Minimális méretű lefogó ponthalmaz keresése egy G irányítatlan gráfban. A probléma NP-nehéz. Jelölje $\tau(G) = \min\{|S| \mid S \text{ lefogó ponthalmaz } G\text{-ben}\}$.

Megengedett válasz: egy lefogó ponthalmaz.

Mohón vegyük sorban minden, az adott pillanatig fedetlen él mindkét végpontját S-hez, amíg van fedetlen él. Az algoritmus során talált fedetlen élek diszjunktak, tehát $|S|/2$ csúcsra szükség van már csak az ő lefogásukhoz is. Így $\tau(G) \geqslant |S|/2$, tehát találtunk egy legfeljebb $2\tau(G)$ méretű lefogó ponthalmazt.

Számítási problémák közelítő megoldásai

Különösen érdekesek az NP-nehéz kiszámítási problémák (eldöntési verziójuk NP-nehéz), ilyenkor ugyanis nem ismeretes hatékony egzakt megoldás. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az viszont hatékony (polinomiális) legyen.

Példa: Minimális méretű lefogó ponthalmaz keresése egy G irányítatlan gráfban. A probléma NP-nehéz. Jelölje $\tau(G) = \min\{|S| \mid S \text{ lefogó ponthalmaz } G\text{-ben}\}$.

Megengedett válasz: egy lefogó ponthalmaz.

Mohón vegyük sorban minden, az adott pillanatig fedetlen él mindkét végpontját S-hez, amíg van fedetlen él. Az algoritmus során talált fedetlen élek diszjunktak, tehát $|S|/2$ csúcsra szükség van már csak az ő lefogásukhoz is. Így $\tau(G) \geqslant |S|/2$, tehát találtunk egy legfeljebb $2\tau(G)$ méretű lefogó ponthalmazt.

Az algoritmus tehát 2-közelítő és hatékonysága $O(|V(G)| + |E(G)|)$.

Számítási problémák közelítő megoldásai

Állítás

Ha P$\neq$ NP, akkor TSP-re semmilyen $g(n)$ függvény esetén se létezik polinom idejű $g(n)$-approximáció. (Megengedett válaszok: az ügynök egy körútja.)

Számítási problémák közelítő megoldásai

Állítás

Ha P$\neq$ NP, akkor TSP-re semmilyen $g(n)$ függvény esetén se létezik polinom idejű $g(n)$-approximáció. (Megengedett válaszok: az ügynök egy körútja.)

Bizonyítás: Ha létezne ilyen, akkor polinom időben megoldhatnánk a Hamilton kör problémát a következőképpen. A Hamilton kör probléma egy tetszőleges G bemeneti gráfjához elkészítjük a TSP egy G' bemeneti gráfját a következőképpen. Legyen $n = |V(G)|$. G' egy teljes, élsúlyozott gráf szintén n csúcson. A G éleinek megfelelő G'-beli élek súlya legyen 1, míg a G nem-éleinek megfelelő G'-beli élek súlya legyen $ng(n)$. G'-t G-ből polinom időben elkészíthetjük. Ha G-ben volt Hamilton-kör, akkor G'-ben van n összsúlyú körút. Ha nem volt, akkor viszont minden körút legalább $ng(n) + n - 1$ súlyú.

Számítási problémák közelítő megoldásai

Állítás

Ha P$\neq$ NP, akkor TSP-re semmilyen $g(n)$ függvény esetén se létezik polinom idejű $g(n)$-approximáció. (Megengedett válaszok: az ügynök egy körútja.)

Bizonyítás: Ha létezne ilyen, akkor polinom időben megoldhatnánk a Hamilton kör problémát a következőképpen. A Hamilton kör probléma egy tetszőleges G bemeneti gráfjához elkészítjük a TSP egy G' bemeneti gráfját a következőképpen. Legyen $n = |V(G)|$. G' egy teljes, élsúlyozott gráf szintén n csúcson. A G éleinek megfelelő G'-beli élek súlya legyen 1, míg a G nem-éleinek megfelelő G'-beli élek súlya legyen $ng(n)$. G'-t G-ből polinom időben elkészíthetjük. Ha G-ben volt Hamilton-kör, akkor G'-ben van n összsúlyú körút. Ha nem volt, akkor viszont minden körút legalább $ng(n) + n - 1$ súlyú.

Hívjuk meg most a polinom idejű approximációs algoritmust.

Számítási problémák közelítő megoldásai

Ha a válasz $\leqslant ng(n)$, akkor a minimáls összsúlyú körút csupa 1 súlyú élekből áll, így G-ben van Hamilton kör. Ha ennél nagyobb értéket kapunk, akkor a $g(n)$-approximáció miatt n-nél hosszabb az optimális körút, így G-ben nincs Hamilton kör. Így ez az algoritmus polinom időben eldönti IHK-t, amiből IHK NP-teljessége miatt P=NP következik, ami ellentmond a feltételeknek. $\square$

Számítási problémák közelítő megoldásai

Ha a válasz $\leqslant ng(n)$, akkor a minimáls összsúlyú körút csupa 1 súlyú élekből áll, így G-ben van Hamilton kör. Ha ennél nagyobb értéket kapunk, akkor a $g(n)$-approximáció miatt n-nél hosszabb az optimális körút, így G-ben nincs Hamilton kör. Így ez az algoritmus polinom időben eldönti IHK-t, amiből IHK NP-teljessége miatt P=NP következik, ami ellentmond a feltételeknek. $\square$

TSP tehát egy rosszul közelíthető probléma. Az alábbi változat viszont jól közelíthető.

Számítási problémák közelítő megoldásai

Ha a válasz $\leqslant ng(n)$, akkor a minimáls összsúlyú körút csupa 1 súlyú élekből áll, így G-ben van Hamilton kör. Ha ennél nagyobb értéket kapunk, akkor a $g(n)$-approximáció miatt n-nél hosszabb az optimális körút, így G-ben nincs Hamilton kör. Így ez az algoritmus polinom időben eldönti IHK-t, amiből IHK NP-teljessége miatt P=NP következik, ami ellentmond a feltételeknek. $\square$

TSP tehát egy rosszul közelíthető probléma. Az alábbi változat viszont jól közelíthető.

Metrikus utazó ügynök probléma: Ugyanaz, mint a TSP, de az élsúlyokra teljesül a háromszög egyenlőtlenség, azaz

$$\forall \{a,b\},\{b,c\},\{a,c\} \in E(G)\text{-re} \quad w(a,c) \leqslant w(a,b) + w(b,c),$$

ahol w a $G = (V,E)$ irányítatlan gráf élsúlyozása.

Számítási problémák közelítő megoldásai

Ha a válasz $\leqslant ng(n)$, akkor a minimáls összsúlyú körút csupa 1 súlyú élekből áll, így G-ben van Hamilton kör. Ha ennél nagyobb értéket kapunk, akkor a $g(n)$-approximáció miatt n-nél hosszabb az optimális körút, így G-ben nincs Hamilton kör. Így ez az algoritmus polinom időben eldönti IHK-t, amiből IHK NP-teljessége miatt P=NP következik, ami ellentmond a feltételeknek. □

TSP tehát egy rosszul közelíthető probléma. Az alábbi változat viszont jól közelíthető.

Metrikus utazó ügynök probléma: Ugyanaz, mint a TSP, de az élsúlyokra teljesül a háromszög egyenlőtlenség, azaz

$$\forall \{a,b\},\{b,c\},\{a,c\} \in E(G)\text{-re} \quad w(a,c) \leqslant w(a,b) + w(b,c),$$

ahol w a $G = (V,E)$ irányítatlan gráf élsúlyozása.

Állítás

A metrikus utazóügynök probléma polinom időben 2-approximálható.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

T összsúlyánál nyilván nem $\exists$ kisebb összsúlyú minden csúcsot lefedő összefüggő részgráf, így az ügynök minden körútja is legalább ilyen hosszú.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

T összsúlyánál nyilván nem $\exists$ kisebb összsúlyú minden csúcsot lefedő összefüggő részgráf, így az ügynök minden körútja is legalább ilyen hosszú. Tehát a kapott körséta hossza legfeljebb 2OPT.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

T összsúlyánál nyilván nem $\exists$ kisebb összsúlyú minden csúcsot lefedő összefüggő részgráf, így az ügynök minden körútja is legalább ilyen hosszú. Tehát a kapott körséta hossza legfeljebb 2OPT.

A háromszög egyenlőtlenség miatt nem nő a körséta hossza, ha a körséta egy csúcsát kihagyjuk, azaz a két szomszédja között éllel helyettesítjük a csúcs meglátogatásához szükséges 2 élt.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

T összsúlyánál nyilván nem $\exists$ kisebb összsúlyú minden csúcsot lefedő összefüggő részgráf, így az ügynök minden körútja is legalább ilyen hosszú. Tehát a kapott körséta hossza legfeljebb 2OPT.

A háromszög egyenlőtlenség miatt nem nő a körséta hossza, ha a körséta egy csúcsát kihagyjuk, azaz a két szomszédja között éllel helyettesítjük a csúcs meglátogatásához szükséges 2 élt.

Alkalmazzuk ezt a rövidítést a körséta minden ismétlődő csúcsára, így végül egy Hamilton kört kapunk, melynek összsúlya legfeljebb 2OPT.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

T összsúlyánál nyilván nem $\exists$ kisebb összsúlyú minden csúcsot lefedő összefüggő részgráf, így az ügynök minden körútja is legalább ilyen hosszú. Tehát a kapott körséta hossza legfeljebb 2OPT.

A háromszög egyenlőtlenség miatt nem nő a körséta hossza, ha a körséta egy csúcsát kihagyjuk, azaz a két szomszédja között éllel helyettesítjük a csúcs meglátogatásához szükséges 2 élt.

Alkalmazzuk ezt a rövidítést a körséta minden ismétlődő csúcsára, így végül egy Hamilton kört kapunk, melynek összsúlya legfeljebb 2OPT.

Az algoritmus műveletigénye $O(|E(G)|)$.

Számítási problémák közelítő megoldásai

Bizonyítás: Legyen G egy bemenet. Készítsünk el G egy T minimális összsúlyú feszítőfáját. Járjuk be T-t mélységi bejárással. Az így kapott körséta a fa minden élét kétszer látogatja meg és egy minden csúcsot legalább egyszer meglátogat.

T összsúlyánál nyilván nem $\exists$ kisebb összsúlyú minden csúcsot lefedő összefüggő részgráf, így az ügynök minden körútja is legalább ilyen hosszú. Tehát a kapott körséta hossza legfeljebb 2OPT.

A háromszög egyenlőtlenség miatt nem nő a körséta hossza, ha a körséta egy csúcsát kihagyjuk, azaz a két szomszédja között éllel helyettesítjük a csúcs meglátogatásához szükséges 2 élt.

Alkalmazzuk ezt a rövidítést a körséta minden ismétlődő csúcsára, így végül egy Hamilton kört kapunk, melynek összsúlya legfeljebb 2OPT.

Az algoritmus műveletigénye $O(|E(G)|)$.

Megjegyzés: Ismeretes (3/2)-közelítő algoritmus is (Christofides).