Számítási modellek

9: Redukció, kiszámíthatóság, bonyolultság

Redukálhatóság / Visszavezethetőség

- Ha tudjuk, hogy egy probléma eldönthetetlen
  (pl. $L_u = \{<M,w> \mid M$ egy TG és $w \in L(M)\}$),
  akkor ezt fel tudjuk használni, hogy megmutassuk más problémákról,
  hogy eldönthetetlenek.
- $HALT = \{<M,w> \mid M$ megáll w inputtal$\}$.

Tétel: *HALT* eldönthetetlen.

Biz. Megmutatjuk, hogy L_u redukálható *HALT*-ra.

- Ellentmondáshoz tfh. *HALT* eldönthető.
- Legyen H egy TG, ami eldönti *HALT*-ot.
- Konstruálunk egy S TG-t, ami eldönti L_u-t.
- S = “$<M,w>$ inputtal
    1. Futtassuk H-t $<M,w>$ inputtal, ami eldönti, hogy M megáll-e w-vel.
       Ha nem, elutasítjuk.
    2. Szimuláljuk M-t w inputtal, amíg meg nem áll (megállás garantált H miatt).
    3. Ha M elfogadta, akkor elfogadjuk.
       Ha M elutasította, akkor elutasítjuk.
- S eldönti L_u-t, ami ellentmondás. Tehát, H nem létezhet. Így *HALT* eldönthetetlen. $\square$

Redukálhatóság / visszavezethetőség

- Ha van két nyelvünk (vagy problémánk) *A* és *B*, akkor *A* **redukálható / visszavezethető** *B*-re azt jelenti, hogy *B*-t használhatjuk *A* megoldására.
- Ha *A* redukálható *B*-re, akkor *B* megoldásával megoldást kapunk *A*-ra.
  - *B* könnyű → *A* könnyű.
  - *A* nehéz → *B* nehéz.
    (ezt a formát fogjuk használni)

E_{TM} eldönthetetlen

- Legyen E_{TM} = {<*M*> | *M* egy TG és *L*(*M*) = Ø}

Tétel: E_{TM} eldönthetetlen.

Biz.: Ellentmondással. Megmutatjuk, hogy L_u redukálhaó E_{TM}-re.

- Tfh. E_{TM} eldönthető. Legyen *R* egy TG, ami eldönti E_{TM}-t.
- Konstruálunk egy *S* TG-t, ami eldönti L_u-t.
- *S* = “<*M*,*w*> inputtal
- 1. Transzformáljuk *M*-t egy új M_w TG-re:
  M_w = “*x* inputtal
    1. Ha $x \neq w$, elutasítjuk.
    2. Másképpen, futtassuk *M*-t *w* inputtal
    3. Elfogadjuk, ha *M* elfogadja.”

  2. Használjuk *R*-t, ami eldönti, hogy $L(M_w)=\emptyset$.

  3. Ha igen [ekkor *M* elutasítja *w*-t] akkor elutasítjuk.

  Ha nem [ekkor *M* elfogadja *w*-t] akkor elfogadjuk.

$$L(M_w) = \begin{cases} \{w\} & \text{if } x = w \\ \emptyset & \text{if } x \neq w. \end{cases}$$

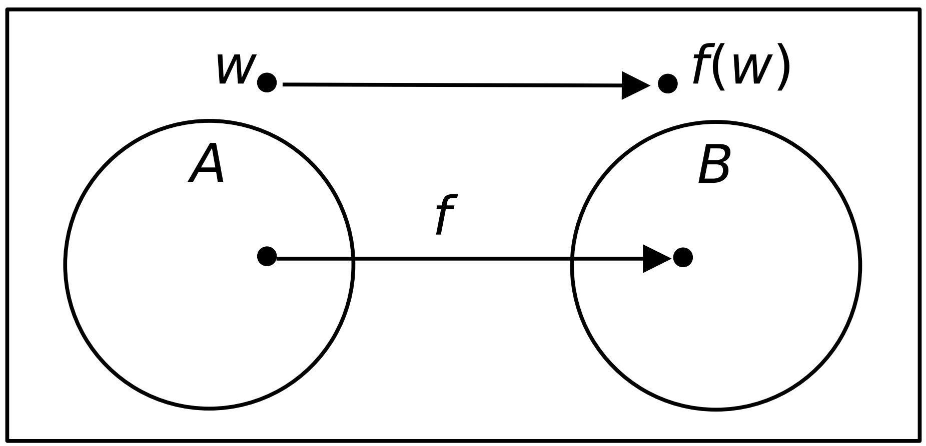
- *S* eldönti L_u-t, ami ellentmondás. Tehát, E_{TM} eldönthetetlen. □

Számítási feladatok

- Az eldöntési problémák általánosításai a (ki)számítási problémák.
- Az $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ TG **kiszámítja** az $f : \Sigma^* \rightarrow \Gamma^*$ függvényt, ha
  - minden $u \in \Sigma^*$-beli szóra megáll, és ekkor
  - $f(u) \in \Gamma^*$ olvasható az utolsó szalagján.
- Az f: $\Sigma^* \rightarrow \Gamma^*$ függvény **kiszámítható**, ha létezik TG M, amely kiszámítja az f függvényt.

Redukálhatóság / visszavezethetőség

- $A \subseteq \Sigma^*$ **redukálható / visszavezethető** $B \subseteq \Gamma^*$-re,
  ha van olyan $f : \Sigma^* \to \Gamma^*$ kiszámítható függvény,
  hogy $w \in A \Leftrightarrow f(w) \in B$. Jelölés: $A \leq B$.



- Példa: $L_u \leq \overline{E}_{TM}$
  A redukcióhoz a kiszámítható függvény:
  $f(<M,w>) = <M_w>$.
  Mivel: $<M,w> \in L_u \;\Leftrightarrow\; <M_w> \in \overline{E}_{TM}$
  (M elfogadja w-t $\Leftrightarrow L(M_w) \neq \emptyset$)

Emlékeztető: M_w = “x inputtal

1. Ha $x \neq w$, elutasítjuk.
2. Máskülönben, futtassuk M-t w inputtal
3. Elfogadjuk, ha M elfogadja.”

Redukció - tulajdonságok

Tétel: HA $A \leq B$ és B eldönthető, akkor A is eldönthető.

Biz.: TF. TG R eldönti B-t.

- Konstruálunk egy S TG-t, ami eldönti A-t:
- S = "w inputtal
  1. Kiszámítjuk $f(w)$-t
  2. Futtassuk R-t az $f(w)$ inputtal, hogy eldöntsük $f(w) \in B$
  3. Mikor R megáll, írjuk ki ugyanazt az eredményt mint R."

Korollár: Ha $A \leq B$ és A eldönthetetlen, akkor B is eldönthetetlen.

Tétel: Ha $A \leq B$ és B T-felismerhető, akkor A is T-felismerhető.

Biz.: Mint fent.

Korollár: Ha $A \leq B$ és A nem T-felismerhető, akkor B sem T-felismerhető.

E_{TM} nem T-felismerhető

- Legyen $E_{TM} = \{<M> \mid M$ egy TG és $L(M) = \emptyset\}$

Tétel: E_{TM} nem T-felismerhető ($E_{TM} \notin RE$).

Biz.: Megmutatjuk, hogy $\overline{L_u} \leq E_{TM}$.

- A redukciós függvény $f(<M,w>)=<M_w>$
- Emlékeztető: M_w = “x inputtal
    1. Ha $x \neq w$, elutasítjuk
    2. egyébként futtassuk M-t w inputtal
    3. elfogadjuk, ha M elfogadja w-t.”

$$L(M_w) = \begin{cases} \{w\} & \text{if } x = w \\ \emptyset & \text{if } x \neq w. \end{cases}$$

- $<M,w> \in \overline{L_u} \iff <M_w> \in E_{TM}$
- Ha E_{TM} T-felismerhető lenne, akkor $\overline{L_u}$ is az lenne. Ellentmondás. $\square$

EQ_{TM} és $\overline{EQ}_{TM}$ nem T-felismerhető

- Legyen $EQ_{TM} = \{<M_1,M_2> \mid M_1,M_2$ TG-k és $L(M_1) = L(M_2)\}$.

Tétel: EQ_{TM} és $\overline{EQ}_{TM}$ nem T-felismerhető (EQ_{TM}, $\overline{EQ}_{TM} \notin RE$).

Biz.:

1.) $\overline{L}_u \leq EQ_{TM}$
2.) $\overline{L}_u \leq \overline{EQ}_{TM}$

- Minden w szóhoz legyen T_w a következő TG:
  T_w = “x inputtal
    1. ignoráljuk x-t
    2. szimuláljuk M-t w inputtal
- 1.) Megadjuk az f függvényt, ami $\overline{L}_u$ problémát ($<M,w>$ formában adott) EQ_{TM} problémára ($<T_1,T_2>$ formában adott) képez:
  - $f(<M,w>) = <T_w,T_{reject}>$, ahol T_{reject} egy TG, ami mindent elutasít.
  - $<M,w> \in \overline{L}_u \Longleftrightarrow <T_w,T_{reject}> \in EQ_{TM}$.
  - Ha EQ_{TM} T-felismerhető lenne, akkor $\overline{L}_u$ is az lenne. Ellentmondás.
- 2.) Hasonlóan, $f(<M,w>) = <T_w,T_{accept}>$, ahol T_{accept} egy TG, ami mindent elfogad.
  - $<M,w> \in \overline{L}_u \Longleftrightarrow <T_w,T_{accept}> \in \overline{EQ}_{TM}$.
  - Ha $\overline{EQ}_{TM}$ T-felismerhető lenne, akkor $\overline{L}_u$ is az lenne. Ellentmondás. □

Bonyolultságelmélet – időbonyolultsági osztályok

- Milyen hatékonyan dönthető el az adott probléma?
- TIME $(f(n)) = \{ L \mid L$ eldönthető $O(f(n))$ időigényű determinisztikus TG-pel $\}$
- NTIME $(f(n)) = \{ L \mid L$ eldönthető $O(f(n))$ időigényű nemdeterminisztikus TG-pel $\}$
- $\mathrm{P} = \bigcup_{k \geq 0} \mathrm{TIME}(n^k)$
- $\mathrm{NP} = \bigcup_{k \geq 0} \mathrm{NTIME}(n^k)$
- Nyilvánvaló: $\mathrm{P} \subseteq \mathrm{NP}$
- Sejtés: $\mathrm{P} \neq \mathrm{NP}$
- A P vs. NP probléma a “7 Milleneumi Probléma” egyike.
    - Igazolásáért vagy cáfolatáért az Intézet 1M $-t fizet a Clay Institute.

Bonyolultságelmélet – NP-beli problémák

- NP: bonyolultsági osztály, amely azon *L* nyelveket tartalmazza, melyekhez létezik *L*-t **polinom időben eldöntő nemdeterminisztikus** TG.

- Másik definíció: Az NP a determinisztikus TG-pel **polinom időben ellenőrizhető** döntési problémák halmaza.

- Precíz tétellé tehető:
  Egy döntési probléma (nyelv) akkor és csak akkor NP-beli, ha minden igen-inputhoz megadható polinom méretű és polinom időben ellenőrizhető **tanú** *c* (ami igazolja, hogy *c* valóban igen-input).

- coNP: bonyolultsági osztály, amely azon nyelveket tartalmazza, melyeknek komplementere NP-beli.
  - Nem tudjuk, hogy coNP különbözik-e NP-től.

Polinom idejű redukció

- Az $f : \Sigma^* \rightarrow \Gamma^*$ függvény **polinom időben kiszámítható**, ha van olyan TG, ami polinom időben kiszámítja.

- $L' \subseteq \Sigma^*$ **polinom időben redukálható / visszavezethető** $L'' \subseteq \Gamma^*$-ra, ha van olyan $f : \Sigma^* \rightarrow \Gamma^*$ polinom időben kiszámítható függvény, hogy $w \in L' \Leftrightarrow f(w) \in L''$.
  Jelölés: $L' \leq_p L''$.

- A polinom idejű redukciót (vagy visszavezetést) **Karp-redukció**nak (vagy **Karp-visszavezetés**nek) is nevezik.

C-teljesség

- Informálisan: ha egy *A* problémát tudunk redukálni egy *B* problémára, akkor *B* legalább olyan nehéz, mint *A*. A legnehezebb problémák (egy bonyolultsági osztályban) azok, melyekre minden probléma redukálható.

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

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

NP-teljesség

- Ha speciálisan C = NP:

- Egy L nyelv NP-teljes, ha
  - $L \in$ NP és
  - L NP-nehéz, azaz minden $L' \in$ NP esetén $L' \leq_p L$
    - $\leq_p$: polinom idejű redukció

- Megjegyzés: Úgy is fogalmazhatunk, hogy az L eldöntési probléma NP-teljes.

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

P vs. NP

- Az előző tétel szerint, ha valaki találna egy NP-teljes problémára polinom idejű determinisztikus algoritmust, azzal bizonyítaná, hogy P = NP.
- Egyetlen NP-teljes problémára sem ismert polinom idejű determinisztikus algoritmus és ha P ≠ NP, ilyet nem is fogunk találni.
- Az NP-teljes problémákra úgy tekinthetünk, mint eldönthető, de hatékonyan nem eldönthető problémákra.

Cook-Levin tétel

- Kielégíthetőségi (satisfiability) probléma:
  SAT = $\{w \mid w$ kielégíthető nulladrendű KNF $\}$

- Példa: KNF:
  $(\neg X_1 \vee X_3) \wedge (\neg X_1 \vee \neg X_3 \vee X_4 \vee X_5) \wedge (X_1 \vee X_2 \vee \neg X_4)$.

- **Tétel** (Cook-Levin): SAT NP-teljes.

- **Tétel**: Ha L NP-teljes, $L \leq_p L'$ és $L' \in$ NP,
  akkor L' NP-teljes.

- A következő problémák NP-teljessége ezen tétel alapján
  bizonyítható.

k-SAT

- A minden tagban pontosan k különböző literált tartalmazó KNF-eket k-KNF-nek nevezzük ($k \geq 1$).

- Példa: 3-KNF:
  $(\neg X_1 \vee X_3 \vee X_5) \wedge (\neg X_1 \vee \neg X_3 \vee X_4) \wedge (X_1 \vee X_2 \vee \neg X_4)$.

- k-SAT = $\{w \mid w$ kielégíthető k-KNF $\}$

- **Tétel**: 3-SAT NP-teljes.
- **Biz.**: Ötlet: A SAT $\leq_p$ 3-SAT redukcióhoz alakítsunk minden $X_1 \vee \cdots \vee X_n$ tagot n-2 tag konjunkciójára:
  $(X_1 \vee X_2 \vee Y_2) \wedge$
  $(\neg Y_2 \vee X_3 \vee Y_3) \wedge$
  $(\neg Y_3 \vee X_4 \vee Y_4) \wedge \cdots \wedge$
  $(\neg Y_{n-3} \vee X_{n-2} \vee Y_{n-2}) \wedge$
  $(\neg Y_{n-2} \vee X_{n-1} \vee X_n)$, ahol $Y_2,\ldots, Y_{n-2}$ új változók.
  Ez a 3-KNF pontosan akkor lesz kielégíthető, amikor az eredeti KNF.
  Az összes tag átalakítása után kapott formula legfeljebb háromszor olyan hosszú, mint az eredeti, azaz a hossznövekedés polinomiális.

- Megjegyzés: 2-SAT $\in$ P.

3-Színezhetőség

- Legyen $k \geq 1$. 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*-Színezés = {<*G*> | *G* gráf *k*-színezhető } .
- **Tétel**: 3-Színezés NP-teljes.
- Megjegyzés: 2-Színezés $\in$ P.

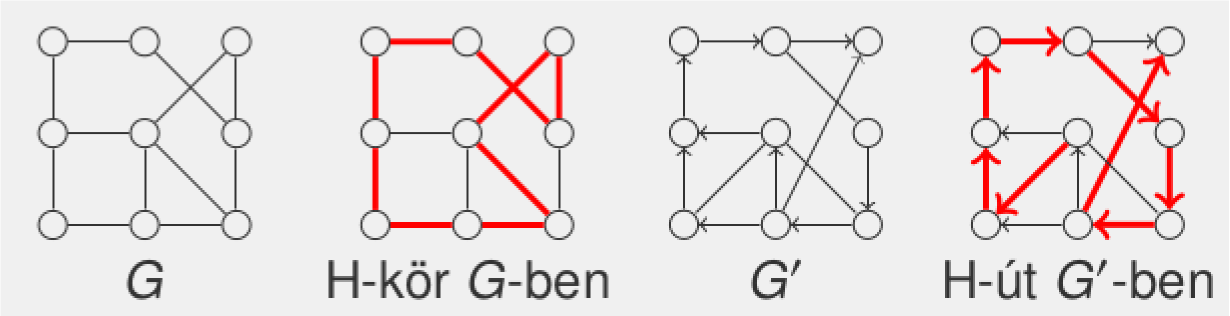
Klikk, független csúcshalmaz

- A G egyszerű, irányítatlan gráf egy teljes részgráfját **klikk**nek, egy üres részgráfját **független csúcshalmaz**nak nevezzük.
- Legyen $S \subseteq V(G)$ és $e \in E(G)$. Ha $S \cap e \neq \emptyset$, akkor a S csúcshalmaz **lefogja** e-t. Ha S minden $e \in E(G)$ élt lefog, akkor S egy **lefogó csúcshalmaz**.
- Klikk = $\{<G,k> \mid G$-nek van k méretű klikkje $\}$
- Független csúcshalmaz = $\{<G,k> \mid G$-nek van k méretű független csúcshalmaza $\}$
- Lefogó csúcshalmaz = $\{<G,k> \mid G$-nek van k méretű lefogó csúcshalmaza $\}$

- **Tétel**: Független csúcshalmaz, Klikk, Lefogó csúcshalmaz NP-teljes.

Irányítatlan/irányított Hamilton út/kör

- Adott egy *G* gráf. Egy a *G* összes csúcsát pontosan egyszer tartalmazó utat **Hamilton út**nak, egy a *G* összes csúcsát pontosan egyszer tartalmazó kört **Hamilton kör**nek nevezünk. Ha a gráf irányított, a Hamilton útnak/körnek irányítottnak kell lennie. Jelölés: H-út/H-kör.



- HÚ= {<*G*,*s*,*t*> | van a *G* irányított gráfban *s*-ből *t*-be H-út } .
- IHÚ= {<*G*,*s*,*t*> | van a *G* irányítatlan gráfban *s*-ből *t*-be H-út } .
- IHK= {<*G*> | van a *G* irányítatlan gráfban H-kör } .

- **Tétel**: HÚ, IHÚ, IHK NP-teljes.

Utazóügynök probléma (TSP)

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

- Eldöntési verzió:
  TSP = {<*G*,*k*> | *G*-ben van ≤ *k* súlyú H-kör } .

- **Tétel**: TSP NP-teljes.

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 P ≠ NP, akkor létezik NP-köztes nyelv.

- Az alábbi problémáknak se a P-belisége, se NP-nehézsége nem ismeretes (így NP-köztes jelöltek):
  - Gráfizomorfizmus= {<*G*,*G'*> | *G* és *G'* irányítatlan izomorf gráfok }.
  - Prímfaktorizáció: adjuk meg egy egész szám prímtényezős felbontását [számítási feladat].
  - Kapuminimalizálás: adott digitális áramkört minimális számú 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.
  Probléma: az input hossza így mindig alsó korlát lesz a tárigényre.

- Az **offline Turing gép (OTG)** egy olyan TG, melynek
  - az első szalagja csak olvasható, a többbi 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

- 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.

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

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

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

Determinisztikus és nemdeterminisztikus tárbonyolultság

- Az offline TG-pel **szublineáris** (lineáris alatti) tárbonyolultság is lehetséges.

- SPACE($f(n)$) := { L | L eldönthető $O(f(n))$ többlet tárkorlátos determinisztikus offline TG-pel }
- NSPACE($f(n)$) := { L | L eldönthető $O(f(n))$ többlet tárkorlátos nemdeterminisztikus offline TG-pel }
- PSPACE := $\cup_{k\geq 0}$ SPACE (n^k)
- NPSPACE := $\cup_{k\geq 0}$ NSPACE (n^k)
- L := SPACE (log n)
  más néven: LSPACE vagy LOGSPACE
- NL :=NSPACE (log n)
  más néven: NLSPACE vagy NLOGSPACE

Savitch tétele

- **Tétel** (Savitch): Ha $s(n) \in \Omega(\log n)$, akkor
  $\mathrm{NSPACE}(s(n)) \subseteq \mathrm{SPACE}(s^2(n))$.
- **Korollár**: PSPACE = NPSPACE.

Hierarchia tétel

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

- **Hierarchia tétel**:
  - NL $\subset$ PSPACE és P $\subset$ EXPTIME.
  - L $\subseteq$ NL=coNL $\subseteq$ P $\subseteq$ NP $\subseteq$ NPSPACE=PSPACE $\subseteq$ EXPTIME

- Sejtés: A fenti tartalmazási lánc minden tartalmazása valódi.

Kiszámítási problémák közelítő megoldásai

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

- Egy algoritmust **α-közelítő**nek 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 /α -szorosa,

- 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.)

NP-nehéz kiszámítási problémák közelítő megoldásai

- Különösen érdekesek a közelítő megoldások az NP-nehéz számítási problémákhoz. A közelítő algoritmustól ilyenkor elvárhatjuk, hogy az hatékony (polinomiális) legyen.

- Példa: Minimális méretű lefogó csúcshalmaz keresése egy *G* irányítatlan gráfban. A probléma NP-nehéz.

  Jelölje $\tau(G) = \min \{|S| \mid S$ lefogó csúcshalmaz G-ben $\}$.

  Megengedett válasz: egy lefogó csúcshalmaz.

  - Mohón vegyük sorban minden, az adott pillanatig fedetlen él mindkét végpontját *S*-hez, amíg van fedetlen él.
  - Az algoritmus során talált fedetlen élek párosítást alkotnak, tehát $|S|/2$ csúcsra szükség van már csak az ő lefogásukhoz is. Így $\tau(G) \geq |S|/2$, tehát találtunk egy legfeljebb $2\,\tau(G)$ méretű lefogó csúcshalmazt.
  - Az algoritmus tehát 2-közelítő és futási ideje $O(|V(G)| + |E(G)|)$.

NP-nehéz kiszámítási problémák közelítő megoldásai

- **Állítás**: Ha P ≠ NP, akkor TSP-re semmilyen $g(n)$ függvényre nincs polinom idejű $g(n)$-approximáció. (Megengedett válaszok: az ügynök egy körútja.)
- **Biz.**: Ha lenne, akkor polinom időben megoldhatnánk a Hamilton kör problémát.
  - A Hamilton kör probléma egy G bemenetéhez konstruálunk a TSP-hez egy G' gráfot:
  - G' éleinek hossza legyen 1 (azaz, ha $(u,v) \in G$), a nem-éleinek hossza pedig $n\, g(n)$ (azaz, ha $(u,v) \notin G$).
  - Ha G-ben volt Hamilton-kör, akkor G'-ben van n hosszú körút, ha nem volt, akkor legalább $ng(n)$ hosszúságú lesz minden körút.
  - Így az approximációs algoritmus válasza alapján polinom időben eldönthető, hogy az eredeti gráfban volt-e Hamilton-kör.

NP-nehéz kiszámítási problémák közelítő megoldásai

- TSP egy rosszul közelíthető probléma. Az alábbi változat viszont jól közelíthető.
- **Metrikus utazó ügynök**: Az élsúlyokra teljesül a háromszög egyenlőtlenség.
- 2-közelítés:
    - Készítsünk egy T minimális feszítőfát, ennél nyilván nem lehet kisebb semmilyen összefüggő részgráf élsúlyainak összege (így az ügynök körútjáé sem).
    - Legfeljebb kétszer ilyen hosszúságú Hamilton-kört gyárthatunk az alábbi algoritmussal:
    - Először járjuk be a T fát, minden élet kétszer érintve ("rajzoljuk körbe"). A kapott körséta hossza legfeljebb 2OPT.
    - Ebben a körsétában lecserélhetünk bármely utat a két végpontja közötti élre, a háromszög egyenlőtlenség miatt ettől nem nő a körséta hossza. Ilyen cserékkel könnyen Hamilton-körré alakíthatjuk a körsétát, melynek összsúlya legfeljebb 2OPT.
- Az algoritmus 2-közelítő és futási ideje $O\,(|V(G)| + |E(G)|)$.
- Megjegyzés: Ismeretes (3/2)-közelítő algoritmus is (Christofides).