

1. Adjon időkorlátot az alábbi feladatokban megkonstruált Turing-gépekhez!

- Adjon 1-szalagos determinisztikus Turing-gépet a $\{0^{2^m} \mid m \geq 0\}$ nyelvhez!

Megoldás: Az első 0-t átírjuk 1-re, ez jelöli majd az input elején az első 0-t. Jobbra haladva minden második 0-t átírjuk x-re. Ha az utolsót nem írtuk át, akkor páratlan hosszú volt az input, elutasítjuk. Ha átírtuk, akkor páros hosszú volt és a megmaradt 0-k száma feleződött. Visszamegyünk az elejére és megint minden második 0-t átírjuk x-re. Az útközben talált x-eken csak jobbra lépünk. Ezt addig ismételjük, amíg lehet. Ha végül nem marad 0, csak az elején az 1-es, akkor elfogadjuk. *Időkorlát:* A bemenet mérete $n = 2^m$. Az első fázisban az input végéig megyünk, majd vissza az elejére, ez $O(n)$ lépés. Ezt annyiszor kell ismételni, ahányszor 2^m -et el kell osztani 2-vel, hogy 1-et kapjunk. Ez $\log_2(2^m) = m = \log_2(n)$. Tehát az időkorlát $O(n \log_2(n))$.

- Adjon 2-szalagos nemdeterminisztikus Turing-gépet a $\{ww \mid w \in \{0,1\}^*\}$ nyelvhez!

Megoldás: Először megjelöljük a 2. szalag elejét, majd lemásoljuk az input első néhány karakterét a 2. szalagra. Minden karakter másolása után lehetőség van a következő karakter másolására vagy a másolás befejezésére és átlépésre egy új állapotba nemdeterminisztikus módon. Az új állapotban a 2. szalag fejét az elejére visszük. Ezután mindkét fejjel jobbra haladva a szó első felét a 2. szalagon karakterenként össze tudjuk hasonlítani az 1. szalagon a szó második felével. *Időkorlát:* az első fej minden lépésben jobbra lép vagy helyben marad és akkor a második fej balra haladva legfeljebb n lépést tesz, ezért a NTG bármely számítási úton $O(n)$ lépést végez.

- Adjon 2-szalagos determinisztikus Turing-gépet a $\{ww \mid w \in \{0,1\}^*\}$ nyelvhez!

Megoldás: 2 szalagos gépet készítünk. Először megjelöljük a 2. szalag elejét, majd lemásoljuk az egész inputot a 2. szalagra. Visszafelé haladva az 1. szalagon csak minden második lépésben mozdítjuk a fejet balra, így amikor a 2. szalagon visszaérünk az elejére, akkor az 1. szalagon a fej középen lesz. Ezután jobbra haladva a 2. szalagon a szó első felét karakterenként össze tudjuk hasonlítani az 1. szalagon a szó második felével. *Időkorlát:* Mindhárom fázisban legfeljebb $n + 1$ lépést végez a TG, tehát az időkorlát $O(n)$.

- Adjon 1-szalagos determinisztikus Turing-gépet a $\{ww \mid w \in \{0,1\}^*\}$ nyelvhez!

Megoldás: Meg kell keresnünk valahogy az input közepét: ehhez a 0-kat x -re, az 1-eket y -ra változtatjuk a következő módon. Átírjuk az első karaktert x -re ill. y -ra, majd az utolsó karaktert is. Mindig visszamegyünk az első 0 vagy 1 karakterre, azt is átírjuk, majd az utolsó ilyen is. A legelső karaktert meg akarjuk különböztetni a többitől, hogy fel tudjuk ismerni, hogy a szalag elején vagyunk, ezért a legelső karaktert x helyett a -ra, y helyett b -re változtatjuk. Ha nem sikerül mindent átírni, akkor az input hossza páratlan volt, elutasítjuk. Ha sikerült, akkor az utolsó átírt karakter épp a második w első karaktere, ezt írjuk át z -re, de menjünk különböző állapotba, ha ott 0 ill. 1 volt. Menjünk a szalag elejére, ott el tudjuk dönteni, hogy a olvasott karakter megegyezik-e a megjegyzett karakterrel, ami a második rész első karaktere. Ha nem, elutasítjuk az inputot. Ha igen, akkor írjuk át ezt is z -re, majd oda vissza ingázva a két rész között, összehasonlítjuk a két részt. Ha minden stimmel, akkor elfogadjuk az inputot. *Időkorlát:* Az input közepének megkereséséhez $n/2$ alkalommal megyünk jobbra, majd balra, mindig legfeljebb n távolságra, ez $O(n^2)$ lépés. Az összehasonlítás során pedig $n/2$ alkalommal megyünk jobbra, majd balra, mindig legfeljebb $n/2$ távolságra, ez is $O(n^2)$ lépés. Tehát az időkorlát $O(n^2)$.

- Adjon 2-szalagos nemdeterminisztikus Turing-gépet az $\{1^m \mid m \text{ összetett szám}\}$ nyelvhez!

Megoldás: Először megjelöljük a 2. szalag elejét, majd néhány x -et írunk a 2. szalagra. Minden karakter írása után lehetőség van még egy x írására vagy az írás befejezésére és átlépésre egy új állapotba nemdeterminisztikus módon. Az új állapotban a 2. szalag fejét az elejére visszük. Ezután mindkét fejjel jobbra haladva, amíg x -et olvasunk a 2. szalagon és 1-et az 1. szalagon, addig jobbra lépünk mindkét szalagon. Ha második szalagon egy $*$ -hoz érünk és az 1. fej 1-et olvas, akkor a 2. fejet az elejére moztatjuk, és ismételjük az előzőeket. Ha második szalagon

egy $*$ -hoz érünk és ekkor az 1. fej is $*$ -ot olvas, akkor az egyesek száma osztható volt az x -ek számával, ezért elfogadjuk az inputot. *Időkorlát:* Vigyázat, ez a NTG így nem időkorlátos, mert az x -ek írása az elején bármennyig eltarthat. Ha azonban úgy módosítjuk, a fenti NTG-t, hogy legfeljebb annyi x -et írjon nemdeterminisztikusan, mint az input hossza, akkor már időkorlátos lesz. Az x -ek írása és az oszthatóság ellenőrzése is minden számítási úton $O(n)$ lépés.

- Álljon az L nyelv azokból az (n, m) párokból, amelyekben n és m egy-egy pozitív egész szám bináris alakja, és ez a két szám relatív prím. Igaz-e, hogy $L \in P$?

Megoldás: Itt a bemenet hossza $\Theta(\log n + \log m)$, de az euklideszi algoritmus, amivel két szám legnagyobb közös osztója meghatározható, ebben is polinom idejű algoritmus.

- Bizonyítsa be, hogy nincs olyan polinom időkorlátos Turing-gép, amelynek bemenete egy pozitív egész szám k bináris alakban, kimenete pedig 2^k bináris alakban!

Megoldás: A bemenet mérete $n = \log_2(k)$, a kimenet mérete viszont $\log_2(2^k) = k = 2^n$, vagyis exponenciális. Mivel tudjuk, hogy $2^n \notin O(n^k)$, ez tehát nem polinom. A kimenetről könnyen látható, hogy egy 1 után k darab 0 következik. Ennek a kimeneti szalagra írásához a Turing-gép mindenképp kénytelen k lépést végezni, ami viszont nem polinomiális az input méretében.

- Adjon polinom időkorlátos Turing-gépet, amelynek bemenete két pozitív egész szám k és m bináris alakban, kimenete pedig $2^k \bmod m$ (vagyis 2^k maradéka m -el osztva) szintén bináris alakban!

Megoldás: Először is vegyük észre, hogy itt nincs probléma a kimenet méretével, hiszen a kimenet egy m -nél kisebb szám, ami tehát nem hosszabb a bemenetnél. Az algoritmus pedig az ismételt négyzetre emelések módszere, ami volt BSZ-ből.

- Álljon az L nyelv azokból az irányítatlan gráfokból, melyekben nincs kör. Igazolja, hogy $L \in P$.

Megoldás: Elegendő azt megmutatni, hogy erre az eldöntési feladatra van polinom idejű algoritmus. A szélességi (vagy mélységi) bejárás alkalmas erre: egy tetszőleges pontból indítjuk és akkor hagyjuk abba, hogy ha vagy minden pontot bejárt vagy talált egy élet, ami nem faél. Az utóbbi esetben van kör a gráfban. Ha viszont minden él faél, akkor a gráf egy erdő, tehát nincs benne kör.

Az algoritmus lépésszáma v csúcsú gráfokon mátrixos megadás esetén $O(v^2)$, ami lineáris a bemenet $N = v^2$ hosszában.

Megjegyzés: Ha ezt Turing-géppel akarjuk megvalósítani, akkor több lépés kell. Hiszen míg a mátrixból két pont közötti él létezését eldönteni $O(1)$ lépés, addig a Turing-gép szalagján oda kell menjünk a megfelelő helyre, ami v^2 -tel arányos lépést is jelenthet. De az ebből adódó $O(v^4)$ összes lépésszám is polinomiális v -ben (és így N -ben is).

- Az L nyelv álljon az olyan (G, s, t) hármasokból, ahol G egy irányított gráf, s és t a gráfnak két csúcsa és G -ben van út s -ből t -be. Igazolja, hogy $L \in P$.

Megoldás: Egy s -ből induló szélességi bejárással megválaszolható a kérdés. (A bejárás leállhat, ha elérünk t -be vagy ha bejártuk s komponensének összes csúcsát.) Mivel a BFS polinom idejű, ezért $L \in P$.

- Legyen $L_1 \in P$ és $L_2 \in P$. Bizonyítsa be, hogy $L_1 \cup L_2 \in P$ is teljesül!

Megoldás: Legyen M_1 egy $O(n^k)$ időkorlátos Turing-gép, ami az L_1 nyelvet ismeri fel, M_2 pedig egy $O(n^\ell)$ időkorlátos Turing-gép, ami az L_2 nyelvet. Konstruálunk egy új, M_3 TG-et: A w bemeneten M_3 először futtatja M_1 -et, majd ha megállt (biztos megáll, hiszen időkorlátos), akkor szintén a w bemeneten futtatja M_2 -t (ez is meg fog állni). Ha bármelyik TG elfogadta az inputot, akkor M_3 is elfogadja, különben elutasítja. M_3 időkorlátja nyilván $O(n^k + n^\ell) = O(n^{\max(k, \ell)})$, vagyis polinomiális.

8. Bizonyítsa be az alábbi két nyelvről, hogy NP-beliek! Melyikről tudja belátni, hogy P-ben van?

(a) G irányítatlan gráfok nyelve, amelyekben van legfeljebb 100 élből álló kör.

(b) (G, k) párokból álló nyelv, ahol a G irányítatlan gráfban van legfeljebb k élből álló kör.

Megoldás: (a) A tanú tétel alapján NP-ben van, hiszen legyen az L_1 nyelv az olyan (G, y) párok nyelve, ahol G egy irányítatlan gráf és y egy legfeljebb 100 élű kör csúcsai a kör mentén sorban felsorolva. Világos, hogy G pontosan akkor van benne a feladatbeli L nyelvben, ha van hozzá olyan y , hogy $(G, y) \in L_1$. Ennek az y -nak a hossza, ha a körbeli csúcsok sorszámát binárisan felírjuk, $100 \log v \in O(\log v)$, ahol v a G csúcsainak száma, tehát $|y|$ kisebb, mint a G megadásának hossza, ezért belefér a polinom korlátba. Még az kell, hogy $L_1 \in P$. Ez azért igaz, mert az L_1 -be tartozáshoz azt kell ellenőrizni, hogy a bemenet y részében legfeljebb 100 csúcs szerepel, ezek mind különbözőek és hogy a szomszédos csúcsok, valamint az első és utolsó csúcs között megy él. (Igazából azt is megengedhetnénk, hogy a felsorolásban legyenek ismétlődő csúcsok, azaz egy zárt élsorozatot adjanak, mert ha ilyen van, akkor legfeljebb 100 élű kör is van.)

(b) Ahhoz, hogy ez NP-ben van, az előző bizonyítást csak kicsit kell módosítani: az y tanú most egy kört alkotó legfeljebb k darab csúcs felsorolása. A bemenetben k binárisan van leírva, azaz a bemenet hossza $\Theta(v^2 + \log k)$. Formálisan nézve egy $k \log v$ hosszú tanú ebben nem polinomiális. De vegyük észre, hogy csak $k \leq v$ esetben lehet megoldás, és ekkor $k \log v \leq v \log v < v^2$, tehát a tanú valóban polinom hosszú (sőt, lineáris) a bemenet hosszához viszonyítva. Mivel az L_1 -be tartozáshoz legfeljebb $k \leq v$ csúcs különbözőségét és ennyi él meglétét kell ellenőrizni, ez megoldható polinom időben, azaz $L_1 \in P$.

Az (a)-ban megadott nyelv világos, hogy P-ben is benne van, hiszen egy v csúcsú gráf esetén a lehetséges 100 élű körök száma nem több, mint v^{100} , ami polinomiális. Adott 100 pontra annak ellenőrzése, hogy az adott sorrendben kört alkotnak konstans sok lépés, tehát minden lehetőség ellenőrzése összesen $O(v^{100})$ és így polinom idő.

A (b)-nél nem alkalmazhatjuk az előző eljárást, mert a lehetőségek száma most v^k nagyságrendű, ami nem polinomiális v -ben és $\log k$ -ban, és itt még az a feltevés sem segít, hogy $k \leq v$. Ennek ellenére ez a nyelv is P-ben van. Ehhez azt kell észrevenni, hogy egy gráfban egy legrövidebb kör megtalálható polinom sok lépésben.

Vegyünk egy tetszőleges uv élet a gráfban és a BFS segítségével keressük meg a legrövidebb utat a $G - uv$ gráfban u és v között. Világos, hogy ez az út az uv éllel kiegészítve a legrövidebb uv élet tartalmazó kör G -ben. Ha ugyanezt elvégezzük a gráf minden élére, akkor az így megtalált körök közül a legrövidebb a gráfban egy legkevesebb élű kör.

A e darab BFS lépéseinek száma $O(v^4)$, a keletkező körök hosszának meghatározása összesen megoldható $O(v^2)$ lépésben, tehát az egész polinomiális.

9. Igazolja, hogy a

(a) $\text{MAXKLIKK} = \{(G, k) : G \text{ irányítatlan gráfban van } k \text{ pontú klikk}\}$ nyelv NP-ben van.

(b) $5\text{KLIKK} = \{G : G \text{ irányítatlan gráfban van } 5 \text{ pontú klikk}\}$ nyelv

(i) NP-ben van, (ii) P-ben van.

Megoldás: (a) A tanú tételt használjuk: az L_1 nyelv álljon az olyan (G, k, y) hármasokból, ahol G egy irányítatlan gráf, k egy pozitív egész és y olyan k darab csúcs felsorolása, melyek G -ben egy teljes gráfot alkotnak. Ha $(G, k) \in \text{MAXKLIKK}$, akkor $k \leq v$, tehát a megfelelő y hossza polinomiális. Azt kell ellenőrizni, hogy y valóban k darab különböző csúcsot tartalmaz, és hogy bármely kettő között megy él (ami kevesebb, mint $k^2 \leq v^2$ él ellenőrzését jelenti), és ez megoldható polinom időben. Ha $(G, k) \notin \text{MAXKLIKK}$, akkor nincs olyan y , ami megfelel ennek az ellenőrzésnek.

(b) Az előző bizonyítás a $k = 5$ esetre alkalmazva jó az NP-beliség bizonyításához.

Az, hogy $5\text{KLIKK} \in P$ abból következik, hogy összesen $\binom{v}{5}$ lehetséges 5 pontú klikk van, ezek ellenőrizhetők összesen $O(v^5)$ lépésben. Ebből, mivel $P \subseteq \text{coNP}$, következik, hogy $5\text{KLIKK} \in \text{coNP}$.

10. Bizonyítsa be, hogy az alábbi nyelvek co NP-beliek!

(a) Az olyan páros gráfok nyelve, amelyekben van teljes párosítás.

(b) Az olyan gráfok nyelve, amelyekben van teljes párosítás.

(c) Az olyan gráfok nyelve, amelyekben akárhogyan színezzük ki az éleket 2 színnel, mindig keletkezik egyszínű háromszög.

Megoldás: Mindegyik esetben azt kell megmutatni, hogy a nyelv komplementere NP-ben van. Ehhez a tanú tétel értelmében elegendő egy hatékony tanúsítványt mutatnunk a komplementer nyelvre, ami lényegében azt jelenti, hogy a nyelvbe nem tartozáshoz kell rövid tanú és ezt ellenőrző polinom idejű eljárás.

(a) A nyelv polinom időben felismerhető, ennek bizonyítása is jó megoldás, de itt most, a gyakorlás kedvéért, egy tanút mutatunk a komplementerre.

Arra, hogy egy páros gráfban nincs teljes párosítás tanú egy olyan X ponthalmaz, ami megsérti a Hall-feltételt. Ha megadjuk a pontoknak egy ilyen részalmazát, akkor ellenőrizni kell, hogy ezek a pontok mind a páros gráf egyik pontosztályából kerültek ki. A szomszédai meghatározásának lépésszáma $O(n^2)$, és ennyi időben azt is ellenőrizni tudjuk, hogy a számuk valóban kisebb mint $|X|$.

Fontos, hogy ha nincs teljes párosítás a gráfban, akkor a Hall-tétel értelmében mindig létezik ilyen X halmaz. Amennyiben van teljes párosítás a gráfban, akkor világos, hogy ilyen X nem létezik.

Szigorúan véve a nyelv komplementerébe azok a bemenetek is beletartoznak, amelyek nem páros gráfot írnak le. Ilyenkor az üres szó is megfelel tanúnak, hiszen polinom időben meg tudunk győződni róla, ha egy gráf nem páros gráf (BFS), és arról is, ha a bemenet nem gráfot ír le (mert nem négyzetszám hosszú 0/1 sorozat vagy a megfelelő mátrix nem lenne szimmetrikus – nem irányítatlan a gráf).

A későbbiekben ilyen esetekben a nem megfelelő alakú bemenet (tehát nem gráf vagy ha irányítatlan gráf kellene, akkor nem ilyet leíró) esetére általában nem fogunk kitérni. Ezek az előbb vázolt módon egyszerűen kezelhetők.

(b) A Hall-tétel helyett itt a Tutte-tételt kell alkalmazni: arra, hogy nincs teljes párosítás tanú egy olyan X ponthalmaz, hogy ezt elhagyva a gráfból, a megmaradt részben a páratlan pontszámú komponensek száma több, mint $|X|$. **Fontos**, hogy ha nincs teljes párosítás a gráfban, akkor a Tutte-tétel értelmében mindig létezik ilyen X halmaz.

Tehát L_1 állhat a (G, x) párokból, ahol x az X halmaz pontjainak felsorolása, aminek n csúcsú gráf esetén a hossza $O(n \log n)$. Az L_1 -be tartozáshoz ellenőrizni kell, hogy x a gráf k különböző csúcsának a felsorolása. Az elhagyásukkal keletkező gráf (mátrixa) polinom időben megkonstruálható. Ennek a maradék gráfnak a komponensei (és azok mérete) egy szélességi bejárással polinom időben meghatározhatók, és innen a Tutte-tételbeli feltétel ellenőrizhető.

Azt is tudjuk, hogy ha van teljes párosítás akkor nincs megfelelő x .

(c) Nézzük a komplementer tulajdonságot: a gráf éleihez lehet úgy két színt rendelni, hogy minden háromszögben mindkét szín előforduljon. Erre tanú egy megfelelő színezés, amit $O(n^2)$ bittel leírhatunk (n a gráf csúcsainak száma). A gráfbeli összes háromszög ellenőrzése $O(n^3)$ lépés, ezért $L_1 \in P$. (A nyelv komplementere az előbb megadott tulajdonságú gráfokból és a nem gráfot leíró bemenetekből áll, de ez utóbbi eset a korábbiak szerint kezelhető, hiszen felismerhető polinom időben.)

11. Álljon a nyelv az olyan (G, t) párokból, ahol G egy nemnegatív élsúlyokkal rendelkező irányítatlan gráf, $t > 0$ egész, és G -ben minden, t darab élből álló párosítás súlya legalább t^2 . Igazolja, hogy ez a nyelv co NP-ben van!

Megoldás: A komplementer tulajdonság: a gráfban van olyan t élű párosítás, melynek súlya kisebb, mint t^2 . Ehhez tanú egy ilyen párosítás, amihez $2t$ csúcsot kell megadni. Ennek hossza $\Theta(t \log v) \subset O(v^2)$, mert $t < v$ kell legyen, ha v a gráf csúcsainak száma. Amit az L_1 -be tartozáshoz ellenőrizni

kell: ez valóban $2t$ különböző csúcsa-e a gráfnak és a megfelelő élek a gráfban szerepelnek-e. Ezek mindegyike megvalósítható $O(t) \subseteq O(v)$ lépésben. Ki kell még számolni a súlyaik összegét, ami kisebb kell legyen, mint t^2 . Ha legnagyobb súly M , akkor minden súlyban a számjegyek száma legfeljebb $\log M$. Mivel az összeadást lineáris időben el lehet végezni, az összeg kiszámolása $O(t \log M)$ lépést igényel. Így a teljes lépésszám $O(v \log M)$. Az input akár éllistával, akár szomszédossági mátrixszal van megadva, a mérete $n \geq v + \log M$, hiszen szerepelnie kell minden csúcsnak és a legnagyobb súlynak is. Mivel $O(v \log M) \subseteq O(n^2)$, az eljárás polinomiális.

12. Legyen $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ olyan polinom időben kiszámolható, bijektív függvény, aminél minden $x \in \{0, 1\}^*$ szóra teljesül, hogy $|f(x)| = |x|$.

Legyen $L = \{y \mid \text{van olyan 1-gyel kezdődő } x, \text{ amire } f(x) = y\}$. Igaz-e, hogy $L \in \text{NP} \cap \text{co NP}$?

Megoldás: A függvény bijektív, tehát van inverze, de ez nem feltétlenül jelenti azt, hogy ezt az inverzet polinom időben ki is tudjuk számolni.

Azt viszont, hogy $L \in \text{NP}$ elég egy megfelelő tanúsítvánnyal igazolni. Legyen $L_1 = \{(y, 1z) \mid f(1z) = y\}$. Ekkor az $x = 1z$ tanú hossza a feltétel szerint $|y|$, az ellenőrzése pedig az $f(x)$ kiszámolásából áll, ami polinomiális.

A co NP -beliséghez az $L_0 = \{(y, 0z) \mid f(0z) = y\}$ nyelv használható, mert a feltétel miatt, ha $f(0z) = y$, akkor nincs másik t szó, amire $f(t) = y$.

13. Igazolja, hogy az a nyelv, ami az összes olyan M determinisztikus véges automata leírásából áll, melyre $L(M) \neq \emptyset$ teljesül, NP-ben van.

Megoldás: Arra, hogy $L(M) \neq \emptyset$ egy tanú lehet egy $x \in L(M)$. Az ellenőrzés lépésszáma $O(|x|)$. Azt kell még meggondolni, hogy van olyan $x \in L(M)$, amire $|x|$ polinomiális az automata leírásának hosszában, azaz, hogy ha a nyelv nem üres, akkor van rövid eleme. Ehhez vegyük észre, hogy ha $x \in L(M)$ egy legrövidebb szó, akkor a neki megfelelő állapotátmenetek sorozatában nem ismétlődhet állapot (különben az ismétlődések közötti rész kihagyható lenne a szóból). Ez viszont azt jelenti, hogy $|x|$ kisebb, mint az állapotok száma, tehát polinomiális (valójában lineáris).

Megjegyzés: igazából ez a nyelv P-ben is benne van: azt kell eldönteni, hogy az automatában a kezdő állapotból elérhető egy elfogadó állapot, ami egy, a gráfon végrehajtott bejárással megoldható.

14. Igazolja, hogy $3\text{SZÍN} \prec 100\text{SZÍN}$!

Megoldás: Módosíthatjuk az előadáson látott $3\text{SZÍN} \prec 4\text{SZÍN}$ bizonyítást. Tetszőleges G gráfból úgy készül $f(G) = G'$, hogy a csúcshalmazt kiegészítjük 97 további csúccsal, ezek legyenek összekötve G minden csúcsával és egymással is. Ezzel egy polinom időben számolható f függvényt definiáltunk, G szomszédossági mátrixát kiegészítjük 97 további sorral, oszloppal, az új, nem diagonális elemek mind 1-ek.

Ha $G \in 3\text{SZÍN}$, akkor G' -re egy ilyen színezés kiterjeszthető: az új csúcsok az eddigiektől és egymástól is különböző színt kapnak, amihez összesen 100 szín elég. Másrészt, ha $G' \in 100\text{SZÍN}$, akkor egy jó színezésében a G csúcsai legfeljebb 3 színnel vannak színezve, tehát ilyenkor valóban $G \in 3\text{SZÍN}$.

Ha precíznek akarunk lenni, a Karp-redukció értékét minden bemenetre, a nem gráfokra is meg kell adni. Könnyű látni, hogy ha f nem változtat az ilyen bemeneten, az jó lesz mivel ilyenkor biztos, hogy a bemenet nem gráf, így se a 3SZÍN , se a 100SZÍN nyelvben nincs benne. Az meg, hogy ez az eset áll fenn, polinom időben eldönthető. Tehát az f számolása azzal kezdődik, hogy ellenőrzi, hogy a kapott x bemenet gráfot ad meg vagy sem, ha nem, akkor $f(x) = x$, különben meg a fent leírt módon módosított gráf lesz f értéke.

Megjegyzés: Mivel általában annak ellenőrzése, hogy a bemenet egy gráf (vagy általában az, hogy formailag jó) általában meg polinom időben, ezért f -et sokszor csak a formailag jó bemeneteken adjuk meg.

15. Mutassa meg, hogy ha $X \prec Y$, akkor $\overline{X} \prec \overline{Y}$ is igaz.

Megoldás: Ugyanaz az f függvény jó lesz, hiszen $x \in X \Leftrightarrow f(x) \in Y$ pontosan akkor teljesül, amikor $x \notin X \Leftrightarrow f(x) \notin Y$.