10. CF és az eldönthetőség. Idő és tár.

1. Algoritmikusan eldönthető-e a következő feladat:

   (a) Adott G környezetfüggetlen nyelvtan és adott R reguláris nyelvtan esetén kérdés, hogy $L(G) = L(R)$ igaz-e.

   (b) Adott G környezetfüggetlen nyelvtan és adott R reguláris nyelvtan esetén kérdés, hogy $L(G)$ tartalmazza-e az $L(R)$ nyelvet.

   (c) Az adott G környezetfüggetlen nyelvtan generál-e 10-nél kevesebb karakterből álló szót.

   (d) Adott két környezetfüggetlen nyelvtan G_1 és G_2 a Σ felett. Igaz-e, hogy $L(G_1) \cup L(G_2) = \Sigma^*$?

   Megoldás:

   (a) Nem. Indirekt bizonyítás: tegyük fel, hogy van rá algoritmus, azaz mindig megálló M TG. Megmutatjuk, hogy akkor olyan M' is van, ami egy adott G CF nyelvtanról eldönti, hogy $L(G) = \Sigma^*$ teljesül-e, ami a tanultak szerint ellentmondás.

   M' egy adott G bemenet esetén futtassa az M gépet a (G,R) páron, ahol R egy reguláris nyelvtan, ami a Σ^* nyelvet generálja, és akkor fogadjon el, ha M elfogad. Így M' pontosan akkor fogadja el a G nyelvtant, ha $L(G) = L(R) = \Sigma^*$, ami csak akkor lehet, ha $L(G) = \Sigma^*$.

   (b) Nem. Az (a)-ban megadott konstrukció most is működik, hiszen $L(G) \supseteq L(R) = \Sigma^*$ csak akkor lehet, ha $L(G) = \Sigma^*$.

   (c) Igen. Ehhez csak ellenőrizni kell az összes 10-nél rövidebb szót. Egy szóra annak eldöntése, hogy benne van-e egy CF nyelvtan által generált nyelvben történhet pl. úgy, hogy Chomsky-normálformájúvá alakítjuk és utána alkalmazzuk a CYK algoritmust. (Vagy az elágazás és korlátozás módszerével a levezetésre kipróbáljuk az összes lehetőséget, azzal a megszorítással, hogy amikor a generálásnál egy legalább 10 hosszú szimbólumsorozatot kapunk, akkor azon az ágon nem megyünk tovább.)

   (d) Nem. Indirekt bizonyítás: tegyük fel, hogy van rá algoritmus, azaz mindig megálló M TG. Megmutatjuk, hogy akkor olyan M' is van, ami egy adott G CF nyelvtanról eldönti, hogy $L(G) = \Sigma^*$ teljesül-e, ami a tanultak szerint ellentmondás.

   M' egy adott G bemenet esetén futtassa az M gépet a $G_1 = G_2 = G$ választással és akkor fogadjon el, ha M elfogad. Így M' pontosan akkor fogadja el a G nyelvtant, ha $L(G) \cup L(G) = L(G) = \Sigma^*$.

2. Legyen $L = \{xxx : x \in \Sigma^*\}$. Igazolja, hogy $L \in \mathrm{SPACE}(\log n)$!

   Megoldás: A logaritmikus tár nem teszi lehetővé azt az egyszerű megoldást, amikor lemásoljuk vagy felülírjuk a bemenetet. Mivel a tár kevesebb, mint n, ezért biztos kell a bemenetin kívül még szalag, és ennek a használatát korlátozzuk. Legyen mondjuk 2 további szalag. A 2-on a bemenetet olvasva számláljuk a karaktereket, azaz, amikor a bemenet végére érünk, akkor a 2. szalagon a bemenet hossza (n) álljon bináris alakban. Sőt még egyszerűbb ha ennek harmadát írjuk fel (minden 3. lépésnél adunk hozzá egyet). Ha a bemenet hossza nem osztható 3-mal, akkor el kell utasítani. Utána a 3. szalagra egymás után az $i = 1, 2, \ldots, n/3$ számokat írjuk (bináris alakban!), és mindegyiknél ellenőrizzük, hogy a bemenet i-edik, $(i+n/3)$-adik és $(i+2n/3)$-adik karaktere megegyezik-e. Ha valamelyik hármas nem azonos, akkor a TG álljon meg elutasítva, különben, ha a végére ért, akkor fogadjon el.

3. Bizonyítsa be, hogy az $L = \{(G,k)$: a G gráfban nincs k független pont $\}$ nyelvre $L \in \mathrm{PSPACE}$ és $L \in \mathrm{EXPTIME}$ is teljesül!

   Megoldás: Gondolkozzunk algoritmusban. Az kell, hogy ez polinomiális méretű tárat, illetve exponenciális időt használjon. Vegyük észre, hogy a triviális algoritmus jó: felsoroljuk a részhalmazokat, és mindegyikre ellenőrizzük, hogy független-e. Egy m csúcsú gráf esetén, ha ez pl. a szomszédossági mátrixával adott, akkor a bemenet hossza $n = m^2$. A lehetséges részhalmazok száma $< 2^m$ és minden részhalmaz ellenőrzése $\leq m^2$ lépés. Egy részhalmaz előállítása az előzőből megy a pontszámban lineáris, $O(m)$ időben, ezért az eljárás $O(m^2 \cdot 2^m)$ időkorlátos, ami miatt $L \in \mathrm{TIME}(m^2 \cdot 2^m) \subset \mathrm{TIME}(2^n) \subset \mathrm{EXPTIME}$.

   A tárigénynél csak arra kell figyelni, hogy ne akarjuk a lehetséges részhalmazok mindegyikét egyszerre eltárolni, hanem amikor előállítunk egyet, akkor történjen meg ennek ellenőrzése, és azután a részhalmazt írjuk felül a következővel. Így akkor csak az aktuális részhalmazt és azt kell tárolni, éppen melyik pontpárjánál tartunk az ellenőrzésében, ami csak lineáris tárat igényel, ezért $L \in$ PSPACE is teljesül.

pont } nyelvre $L \in$ SPACE(n) és $L \in$ TIME(2^n) is teljesül (a gráf a szomszédossségi mátrixával adott, n a bemenet hossza) !

Megoldás: Gondolkozzunk algoritmusban. Vegyük észre, hogy a triviális algoritmus jó: felsoroljuk a részhalmazokat, és mindegyikre ellenőrizzük, hogy független-e. Egy m csúcsú gráf esetén, ha ez a szomszédossági mátrixával adott, akkor a bemenet hossza $n = m^2$. A lehetséges részhalmazok száma $< 2^m$ és minden részhalmaz ellenőrzése $\leq m^2$ lépés. Egy részhalmaz előállítása az előzőből megy a pontszámban lineáris, $O(m)$ időben, ezért az eljárás $O(m^2 \cdot 2^m)$ időkorlátos, ami miatt $L \in$ TIME$(m^2 \cdot 2^m) \subset$ TIME(2^n).

A tárigénynél csak arra kell figyelni, hogy ne akarjuk a lehetséges részhalmazok mindegyikét egyszerre eltárolni, hanem amikor előállítunk egyet, akkor történjen meg ennek ellenőrzése, és azután a részhalmazt írjuk felül a következővel. Így akkor csak az aktuális részhalmazt és azt kell tárolni, éppen melyik pontpárjánál tartunk az ellenőrzésében, ami csak lineáris tárat igényel, ezért $L \in^(n)$.

4. Legyen L az $S \to \mathtt{a}S\mathtt{a} \mid \mathtt{b}S\mathtt{b} \mid \mathtt{a} \mid \mathtt{b}$ nyelvtan által generált nyelv. Igaz-e, hogy az L nyelv
   (a) rekurzívan felsorolható?
   (b) rekurzív?
   (c) reguláris?
   (d) SPACE(n)-ben van?
   (e) TIME(n)-ben van?

Megoldás: A generált nyelv a palindromok nyelve. amiről tudjuk, hogy van hozzá veremautomata. Akkor Turing-gép is van, tehát az (a) igaz. (Valójában láttunk is rá determinisztikus Turing-gépet az előadáson.)

A veremautomata olyan, hogy megáll a szó végénél, ezért megálló Turing-gép is van, azaz a (b) is igaz.

(c) az anyagban szerepelt, hogy ez a nyelv nem reguláris (pumpálás!). teját a válasz: nem.

(d) Az előadáson vázolt 1 szalagos és 2 szalagos Turing-gép is csak lineáris hosszú szalagot használ (az egyik a bemenetet, a másik ezt lemásolja), ezért igaz.

(e) Az említett két Turin-gép közül az 1 szalagos lépésszáma $O(n^2)$, de a 2 szalagosé $O(n)$, és ez utóbbi igazolja, hogy ez az állítás is igaz.

Megjegyzés: elég lett volna a (c) mellett az (e) állítást indokolni, mert ha egy nyelv benn van a TIME(n) osztályban, akkor a SPACE(n) osztályban is benne van, sőt rekurzív, és így rekurzívan feésorolható is.