Logika és számításelmélet GY
Második Zárthelyi, 2009-12-7 hétfő

Instrukciók

- Minden feladat 1 pontot ér, részpontokat lehet kapni. A zárthelyi érdemjegye maga a pontszám kerekítve.
- **Akinek csak számításelméletből kell írni, annak az 5B az 5. feladata, a többieknek (akiknek logika is kell) pedig az 5A.**
- Eredmények szerdán infosheeten/kurzusfórumon.

1. Feladat: Legyen $\Sigma = \{a,b\}$ és $L = \{aba^n | n \geq 0\}$. Ekkor

a. Adjon meg egy egyszalagos determinisztikus Turing-gépet, ami felismeri az L nyelvet!
b. Adja meg a Turing-gép végrehajtási sorozatát az *abaa* szóra!

2. Feladat: Legyen $\Sigma = \{a,b\}$ és $f{:}\Sigma^* \to \Sigma^*$ olyan, hogy $\forall\, w \in \Sigma^*{:} f(w) = w \sqcup a^m, \text{ahol } m = l(w)+1$. Adjon meg egy determinisztikus Turing-gépet, ami kiszámítja ezt a függvényt!

3. Feladat: Legyen $\Sigma = \{a,b,c\}$ és legyen adott az M Turing-gép (lsd ábra).

a. Adja meg a gép végrehajtási fáját az *aca* szóra!
b. Melyik nyelvet ismeri fel M?
c. Adjon időkorlátot M-re, ha ez lehetséges!

4. Feladat: Eldönthető-e az a probélma, hogy egy adott M Turing-géphez létezik-e olyan szó, amin megáll?

5A. Feladat: Legyen adott a következő formulahalmaz:
$H = \{\forall\, x \,\exists\, yP(x,y) \supset \forall\, y(P(y,y) \wedge \exists\, zQ(z,y)),\ \forall\, xP(x,a),\ \forall\, x \,\forall\, y(Q(x,y) \supset P(x,x))\}$

a. Állítsa elő a *H*-nak megfelelő *S* klózhalmazt. (*a* konstanst jelöl)
b. Mi lesz az *S*-hez tartozó Herbrand-univerzum?
c. Mutassa meg alaprezolúcióval, hogy a halmaz kielégíthetetlen.

5B. Feladat: Legyen adott a következő *SAT**-gal jelölt probléma:

SAT = { dnf | dnf diszjunkt normálforma kielégíthető }*

Mutassa meg, hogy az *SAT** probléma NP-teljes!

(Emlékeztető:

- dnf = $(l_1 \wedge l_2 \wedge ...) \vee (l_1' \wedge l_2' \wedge ...) \vee ...$, ahol minden l egy ítéletváltozó vagy annak negáltja
- knf = $(l_1 \vee l_2 \vee ...) \wedge (l_1' \vee l_2' \vee ...) \wedge ...$, ahol minden l egy ítéletváltozó vagy annak negáltja)

6. Feladat: Legyen $\Sigma = \{a,b,c\}$ és legyen $L = \left\{uws | u,w,s \in \Sigma^* \wedge l(s) = l(u) \cdot l(w)\right\}$
Adjon olyan (tetszőleges) Turing-gépet, ami eldönti ezt a nyelvet!

Megoldás

M1.

a. Könnyű, minden lépésben olvassunk be egy betűt a szalagról, írni ugyanazt írjuk vissza és léptessük a fejet jobbra, az állapot pedig változzon a következők szerint:

- q0 -> q1, ha a
- q1 -> q2, ha b
- q2 -> q2, ha a
- q2 -> qi, ha ü (= üres szalag szimbólum)
- minden egyéb esetben pedig -> qn

b. Az így kapott Turing-gép végrehajtási sorozata abaa-ra a következő:
q0, $\underline{a}$baa -> q1, a$\underline{b}$aa -> q2, ab$\underline{a}$a -> q2, aba$\underline{a}$ -> q2, abaa$\underline{ü}$ -> qi

M2.

Adjunk meg egy kétszalagos gépet, ami először egy ü-t ír a segédszalagra, majd átmásolja w-t. Ezután lép egyet jobbra az inputszalagon (ü az inputra), majd a segédszalagon visszafele olvasva l(w) számú a-t kiír az inputszalagra. Végül mikor qi-be lép, még egy a-t kiír oda.

M3.

a. aca végrehajtási fája
b. mit ismer fel: Egyszerűen meg kell nézni mit csinál a gép. Először kiír egy ü-t a segédszalagra [q0], majd valahány (m>=0) betűt átmásol az inputról a segédre [q1]. Mikor elér egy c-ig akkor áll meg, majd a segéden visszalépteti a fejet az átmásolt rész végére. Ezt követően c-ket olvas be párhuzamosan az átmásolt rész olvasásával [q2]. Ez addig tart (n>=0 lépés), amíg el nem érjük a segédszalag elejére írt ü-t (=> n = m). Ezt követően megint vissza tesszük a segéd fejét a kimásolt részre és párhuzamosan ugyanazt olvassuk a két szalagról [q3]. Ha mindkettő egyszerre elfogyott, akkor elfogadjuk a szót. Következésképpen az elfogadott szó "u c^m u", alakú, ahol u tetszőleges m-hosszú szó az abc felett. Továbbá m>0, ugyanis az üres szót nem fogadja el a gép.
c. Időkorlát: Ez sem nehéz feladat. A gép átmenetgráfja a hurkokon kívül más kört nem tartalmaz és a hurkokon mindig haladunk előre az inputszalagon, továbbá az inputon nem lépünk vissza. Így mindenféleképpen véges lesz a lépésszám minden inputra. Valamint a hurokéleken összesen legfeljebb n lépést tehetünk meg (az input szó hossza), a további inputon nem előre-haladó élek száma az egész gráfban 4. Így az n+4 jó időkorlátnak.

M4.

Könnyű, az utolsó gyakorlaton vett "Adott Tg megáll-e minden szón?" probléma megoldása itt is egy-az-egyben alkalmazható. Vagyis ugyanannak az M'-konstrukció felhasználásával meg kell mutatni, hogy L_{halt} visszavezethető erre a problémára. Innen pedig tétellel következik az eldönthetetlenség.

M5a.

Jelölje most az egyszerűség kedvéért v a vagyot és n a negációt. A feladatot persze elírtam az utolsó implikációs formula utolsó tagja negálva kéne, hogy legyen. Sebaj, a feladat így is értelmes, legfeljebb alaprezolúcióval nem jön ki az üresklóz. Amiből következik, hogy S kielégíthető lenne.

a. Csak a klózok magjait felsorolva a klózhalmaz: S = {(1) nP(b,y) v P(w,w), (2) nP(b,y) v Q(f(y,w), w), (3) P(x,a), (4) nQ(x,y) v **n**P(x,x)}, ahol b új Skolemkonstans, f(,) pedig új Skolemfüggvény.
b. Herbrand-uni = {a,b,f(a,a), f(a,b), f(b,a), f(b,b), f(f(a,a), a), ...}
c. Alaprezolúcióval: (ha jól adtam volna fel...)

1. P(b,a) /(3)
2. nP(b,a) v P(f(a,a), f(a,a)) /(1)
3. P(f(a,a), f(a,a)) /rez(1,2)
4. nP(b,a) v Q(f(a,a),a) /(2)
5. Q(f(a,a),a) /rez(1,4)
6. nQ(f(a,a),a) v nP(f(a,a), f(a,a)) /(4)
7. nP(f(a,a), f(a,a)) /rez(5,6)
8. üresklóz /rez(3,7)

M5b.

Nyilvánvalóan egy dnf-ből könnyen elő lehet állítani egy neki megfelelő knf-et (minden tagból egy elemet kiválasztunk, ezeket össze-vagyolva kapjuk a knf egy tagját, és ezt az összes lehetséges módon elvégezve kaphatjuk a knf-et), és fordítva is egy knf-ből a megfelelő dnf-et. Ez az átalakítás nyilván egy Turing-kiszámítható függvény lesz, és elvégezhető polinomiális időben. Így a két feladat egymásra visszavezethető, azaz $SAT \leq_p SAT^*$és $SAT^* \leq_p SAT$ is teljesül.
Ekkor tétel szerint $SAT^* \leq_p SAT \wedge SAT \in NP \Rightarrow SAT^* \in NP$, továbbá $SAT \leq_p SAT^* \wedge SAT^* \in NP \wedge SAT\ \mathrm{NP-teljes} \Rightarrow SAT^*\ \mathrm{NP-teljes}$.

M6.

Mivel u és w határát csak próbálgatással lehet eldönteni, lehetett sejteni, hogy nemdeterminisztikus megoldás az elvárt (és ez megengedett is volt). 3 szalagos géppel a következőt lehetett vázlatosan elvégezni: Először valameddig lemásoljuk a szó elejét az első segédszalagra, majd egy újabb valamekkora részét a másik segédszalagra. Ekkor a szorzatot lehet úgy kezelni, hogy az első segédszalagon annyiszor megyek végig, ahány betű van a másikon. Ekkor már csak ki kell olvasni maradék részét az inputnak, miközben számoljuk a szorzatot. Esetleg ki lehetett számolni előbb a szorzatot egy negyedik szalagra (értsd: ráírunk szorzat-számú a betűt a 4. szalagra), majd a 4 újraolvasását lehetett volna párhuzamosan végezni a maradék rész kiolvasásával.