F. Drewes 2005-04-26

# Datavetenskapens grunder, VT06

## – Gruppövning 3 –

Fokusera på uppgift 1 och 2. Den tredje är mindre viktigt och diskuteras i mån av tid.

1. Är mängderna av (1) alla avgörbara språk och (2) alla igenkännbara (alltså semi-avgörbara, eng. *Turing recognizable*) språk slutna under (a) union, (b) snitt och (c) komplement? Argumentera!

2. Den här uppgiften är ganska lik inlämningsuppgiften från i måndags. Den handlar om följande problem.

   *Input:* Ett reguljärt uttryck $R$ och en DFA $M$.
   *Fråga:* Gäller $L(R) \cap L(M) \neq \emptyset$, dvs accepterar $M$ minst en sträng i $L(R)$?

   Formulera problemet som ett språk och visa sedan att det är avgörbart genom att använda dig av resultat som bevisades tidigare.

3. Ett för teorin väldigt viktigt algoritmiskt problem är *Posts korrespondensproblem* (se avsnitt 5.2 i boken). Dess input är ett ändligt antal "dominostenar" $\left[\frac{u_1}{v_1}\right], \ldots, \left[\frac{u_k}{v_k}\right]$ där $u_i$ och $v_i$ är strängar. Frågan är om det finns en ändlig indexsekvens $i_1, \ldots, i_n$ ($n > 0$, $i_j \in \{1, \ldots, k\}$) där

   $$u_{i_1} \cdots u_{i_n} = v_{i_1} \cdots v_{i_n}.$$

   Med andra ord, om man lägger lämpliga (kopior av) dominostenarna i en rad, får man då samma sträng i övre och undre halvan? Detta kallas en *matchning*.

   Ett exempel är $\left[\frac{b}{a}\right]$, $\left[\frac{a}{abb}\right]$, $\left[\frac{aab}{b}\right]$, där bl a sekvensen $2, 1, 1, 3$ producerar lika strängar:

   $$\left[\frac{a}{abb}\right]\left[\frac{b}{a}\right]\left[\frac{b}{a}\right]\left[\frac{aab}{b}\right].$$

   Diskutera först i små grupper, sedan allihop, om ni tror att problemet är avgörbart, semi-avgörbart, eller inte ens det. Försök hitta algoritmer respektive förklara varför ni misslyckas. Blir situationen annorlunda om alfabetet för strängarna består av bara en symbol?

Kommentarer…

1. Svaret är 'ja' utom i fall 2c. För avgörbara språk: Låt $M, M'$ avgöra språk $L, L'$. För att avgöra $L \cup L'$ och $L \cap L'$ kombineras $M$ och $M'$ till nya maskiner som först räknar ut $M(u)$ och $M'(u)$ och sedan svarar i enlighet med resultaten. Att avgöra $L$s komplement $\overline{L}$ är ännu enklare – maskinen räknar precis som $M$ men negerar till slut svaret (dvs byter ut 0 mot 1 och tvärtom).

   För att visa slutenhet i fallen 2a och (särskilt) 2b är det bra att komma ihåg att uppräkningsbar och semiavgörbar är detsamma. Låt alltså nu $M$ och $M'$ vara TM:er som semiavgör $L$ respective $L'$. Fall 2b fungerar precis som 1b medan 2a är lite mer komplicerat. $M(u)$ och $M'(u)$ måste beräknas "parallellt" eftersom bara en av dem behöver stanna. Kort sagt, de två initiala konfigurationerna sparas på bandet genom att inputsträngen kopieras. Sedan körs $M$ och $M'$ stegvis omväxlande tills en av dem stannar.

   Mängden av alla uppräkningsbara språk är inte sluten under komplement. Det visades på föreläsningen att $L$ är avgörbart om och endast om både $L$ och $\overline{L}$ är semiavgörbara (=uppräkningsbara), dvs om $L$ är semiavgörbart men inte avgörbart kan $\overline{L}$ inte vara semiavgörbart.

2. Om vi skriver problemet som ett språk får vi

   $$\{\langle R, M\rangle \mid R \text{ är ett reguljärt uttryck och } M \text{ en DFA där } L(R) \cap L(M) \neq \emptyset\}.$$

   Enligt resultat som bevisats tidigare (mha konstruktiva bevis!) kan $R$ skrivas om till en ekvivalent DFA. Dessutom har vi övertygat oss om att man, om man har två DFA:er $M, M'$, kan skapa en som accepterar $L(M) \cap L(M')$ (dvs klassen av reguljära språk är sluten under snitt) och problemet $E_{\text{DFA}}$ (dvs om en DFA inte accepterar någon sträng alls) är avgörbart. Vi får alltså följande algoritm för att avgöra problemet:

   (a) Konstruera en DFA $M'$ som är ekvivalent med $R$.
   (b) Utifrån $M$ och $M'$, konstruera $M''$ sådan att $L(M'') = L(M') \cap L(M)$.
   (c) Avgör om $\langle M''\rangle \in E_{\text{DFA}}$ och returnera negationen av resultatet.

3. Problemet är inte avgörbart men däremot är det semi-avgörbart. Att visa semi-avgörbarhet är enkelt: Man prövar helt enkelt alla indexsekvenser tills man har hittat en lösning. Oavgörbarheten (vars bevis hittas i boken) beror intuitivt på att index-sekvensen som representerar en lösning kan bli hur lång som helst. Så länge man inte har hittat en lösning vet man alltså (utom in undantagsfall) aldrig när man kan sluta leta.

   Om alfabetet innehåller bara en symbol är problemet däremot avgörbart. Svaret är i så fall 'ja' om och endast om det finns $i, j$ sådana att $|u_i| \leq |v_i|$ och $|u_j| \geq |v_j|$.