Algoritmer, datastrukturer och komplexitet

Övning 10

Anton Grensjö
grensjo@csc.kth.se

18 november 2015

Kursplanering

Ö9: NP-fullständighetsbevis

F27: NP-fullständighetsreduktioner

F28: NP-fullständighetsreduktioner

Ö10: NP-fullständighetsbevis

Bonusdatum labb 4 måndag 23/11

F29: Approximationsalgoritmer

F30: Approximationsalgoritmer

Ö11: Approximationsalgoritmer

Idag

- Fler bevis av NP-fullständighet

Repetition

Definitioner

- Ett problem tillhör NP om det kan verifieras på polynomisk tid.
- Ett problem A är NP-svårt om varje problem i NP kan reduceras till A.
- Ett problem är NP-fullständigt om båda dessa villkor är uppfyllda.

Bevis av NP-fullständighet

Vad bör ingå i ett bevis?

Säg att vi vill visa att problemet A är NP-fullständigt. Då bör vi utföra följande steg:

1. Visa att $A \in NP$.
   a) Föreslå vad en lösning kan vara.
   b) Visa att om svaret är ja så kan lösningen verifieras.
   c) Visa att verifikationen tar polynomisk tid.
2. Visa att A är NP-svårt.
   a) Hitta ett känt NP-fullständigt problem B att reducera.
   b) Hitta och beskriv en karp-reduktion av B till A.
   c) Bevisa att reduktionen är polynomisk.
   d) Bevisa att reduktionen är korrekt.
      i) Visa att ja-instanser mappas till ja-instanser och nej-instanser till nej-instanser.

   Vi vet sedan att $B \leq_p A$.
3. Nu har vi visat att A är NP-fullständigt!

Uppgift 6 (övning 9): Är en Eulergraf k-färgbar?

Betrakta beslutsproblemet att givet en Eulergraf G och ett heltal $k \geq 3$ avgöra om G är k-färgbar. Är detta NP-fullständigt?
Ja. Ligger i NP, då det är enkelt att verifiera om en lösning är en korrekt färgning. För att visa NP-svårt så reducerar vi k-färgningsproblemet.

- Säg att vi har en graf G. Vi gör om den till en Eulergraf G' enligt följande:
  - Handskakningslemmat $\implies$ det finns ett jämnt antal noder med udda valens.
  - Para ihop dessa noder två och två.
  - För varje par, inför ett nytt hörn, med en kant till vardera hörn i paret.
- Varför är G' en Eulergraf?
- Modifikationerna påverkar inte antalet färger, varför inte?

Uppgift 6: Är en Eulergraf k-färgbar?

Visa G k-färgbar $\iff$ G' k-färgbar.

$\Longrightarrow$: Antag f en färgning av G (varje hörn x får en färg $f(x)$).
Definiera färgningen f' av G':

- Om $x \in G, \ \ f'(x) = f(x)$
- Om $x \notin G$ så finns det två grannhörn $y, z \in G$. Låt $f(x)$ vara en godtycklig färg som är skild från $f(y)$ och $f(z)$.

$\Longleftarrow$: Trivialt. Varför?

Problem 1: Konstruera kappsäckslösning

Kappsäcksproblemet:

- Beslutsproblem.
- Indata:
  - Mängd P av föremål med vikt w_i och värde v_i.
  - Kappsäcksstorlek S.
  - Ett mål K.
- Fråga: Går det att välja ut föremål av sammanlagt värde minst K så att deras sammanlagda vikt är högst S?
- Detta är ett känt NP-fullständigt problem.

Uppgift:

- Antag att vi har en algoritm A som löser beslutsproblemet.
- Konstruera en algoritm som löser det konstruktiva kappsäcksproblemet:
  - Om svaret på A är ja, vilka föremål ska packas ned i kappsäcken?
- Vi får anropa A $\mathcal{O}(|P|)$ gånger, men den ska i övrigt gå på polynomisk tid.

Problem 1: Konstruera kappsäckslösning

- Idé: Använd beslutsproblemet för att avgöra om ett element behövs. Om det inte behövs, släng bort det.
-
```
function KAPPSÄCK(P, S, K)
    if ¬A(P, S, K) then return "Ingen lösning"
    for varje föremål p ∈ P do
        if A(P − {p}, S, K) then P ← P − {p}
    return P
```
- Korrekthet
  - P innehåller inga överflödiga element, ty for-slingan löper över alla element i P och tar bort de som är överflödiga.
  - P innehåller en lösning, ty detta är dels uppfyllt innan slingan, och dels efter varje iteration.

Problem 2: Tillförlitlighet hos internet

Bakgrund:

- Datorer och förbindelser mellan datorer går ofta sönder.
- Därför vill man att det ska finnas flera alternativa vägar mellan varje par av datorer på internet.
- T.ex: Om det finns tre olika disjunkta vägar (med avseende på hörn) mellan dator A och B så kan två datorer (vilka som helst) försvinna, utan att förbindelsen mellan A och B försvinner.

Problem 2: Tillförlitlighet hos internet

Problemet:

- Se internet som en oriktad graf. Hörn: datorer. Kant från X till Y: en möjlig direktförbindelse mellan X och Y.
- För varje par av datorer (X, Y) har vi en önskad tillförlitlighet $T(X, Y)$, som säger hur många disjunkta vägar det ska finnas i internet mellan X och Y.
- Vi har en budget B, som säger hur många direktförbindelser vi har råd att skapa.
- Frågan: Kan vi plocka ut en delgraf bestående av B kanter, så att det för alla par av hörn (X, Y) finns minst $T(X, Y)$ disjunkta stigar i delgrafen mellan X och Y?

Visa att detta problem är NP-fullständigt!

Problem 2: Tillförlitlighet hos internet

- Beslutsproblem: Givet grafen G, går det att plocka ut en delgraf G' bestående av exakt B kanter så att det för alla par av hörn (X, Y) finns minst $T(X, Y)$ disjunkta stigar i G' mellan X och Y?
- Visa att problemet ligger i NP.
  - Lösning? En delgraf G' enligt ovan. Tillsammans med, för varje par (X, Y) av noder, en lista över minst $T(X, Y)$ disjunkta stigar.
  - Kan på polynomisk tid verifiera genom att
    a) Kontrollera att G' är en delgraf till grafen.
    b) Kontrollera att G' har exakt B kanter.
    c) Kontrollera för varje par av hörn att det krävta antalet disjunkta stigar finns specifierade, är korrekta och disjunkta.

Problem 2: Tillförlitlighet hos Internet

- Visa att problemet är NP-svårt.
  - Vilket problem ska vi reducera?
    Hamiltonsk cykel fungerar fint.
  - 1: **function** HAMCYCLE($G = (V, E)$)
    2: $B \leftarrow |V|$
    3: $T(X, Y) \leftarrow 2, \quad \forall X, Y \in V$
    4: **return** INTERNETBESLUT(G, T, B)
  - Reduktionens tidskomplexitet $\in \mathcal{O}(|V|^2)$, polynomiskt i indatastorlek.
  - Korrekthet?

Problem 2: Tillförlitlighet hos internet

Korrekthet:

- Antag att vi har en ja-instans s av hamiltonsk cykel, dvs det finns en hamiltoncykel i G. Vi ska då visa att vår konstruerade instans s' av INTERNETBESLUT också är en ja-instans.

  Notera att om vi väljer ut delgrafen $G' = (V, E')$, där E precis innehåller kanterna i hamiltoncykeln, så är det en lösning till INTERNETBESLUT. Varför?

- Antag att vår konstruerade instans s' av INTERNETBESLUT är en ja-instans. Vi vill visa att det existerar en hamiltoncykel i G.

  Välj två godtyckliga noder $X, Y \in V$. Det existerar två disjunkta stigar mellan dem. Unionen av dessa stigar är en hamiltoncykel. Varför?

Problemet är således NP-fullständigt.

Problem 3: Håstads leksaksaffär

- Håstads leksaksaffär säljer ett pussel som består av en låda och ett antal kort (se figur).
- Kort kan placeras i lådan på två sätt, med framsidan upp eller med baksidan upp, eftersom det finns urgröpningar i korten som måste passa in i lister som sitter i lådan.
- På varje kort finns två kolumner med hål, men vissa av hålen är igenfyllda.
- Målet med pusslet är att lägga korten i lådan på ett sånt sätt att hela botten täcks; för varje hålposition måste alltså åtminstone ett av korten ha ett igenfyllt hål.

Bevisa att följande språk är NP-fullständigt:

$$\textsc{Toys} = \{\langle c_1, c_2, \ldots, c_n \rangle : (c_i \text{ beskriver kort}) \land (\text{det går att lösa pusslet})\}$$

Håstads leksaksaffär

- Visa ligger i NP.
- Visa NP-svårt.
  - Vilket problem ska vi reducera?
    Vi kan försöka med CNF-SAT.
    - Givet en CNF-formel

$$\varphi = (l_1 \vee l_2 \vee \cdots \vee l_i) \wedge (l_j \vee \cdots \vee l_k) \wedge \cdots \wedge (\cdots)$$

      med n variabler och m klausuler, vill vi konstruera en en instans γ av LEKSAKER, som är satisfierbar om och endast om γ är lösbar.

Håstads leksaksaffär

Reduktionen

Vi reducerar problemet enligt följande:

1) Varje variabel x_i motsvarar ett kort c_i.
2) Det finns två kolumner med m positioner på varje kort, dvs en rad för varje klausul.
3) Om x_i finns med i klausul j, så är det vänstra hålet på rad j i c_i förtäckt.
4) Om $\bar{x}_i$ finns med i klausul j, så är det högra hålet på rad j i c_i förtäckt.
5) Alla andra positioner har hål.
6) Det finns ett extra “magiskt” kort c_{n+1} med hål endast i den vänstra kolumnen.

Antag att φ är satisfierbar. Då löser vi vårt specialfall γ av LEKSAKER genom att låta extrakortet ligga rätt väg, låta c_i ligga rätt väg om $x_i = 1$, men vända på c_i om $\bar{x}_i$. Alla hål täcks eftersom varje klausul är satisfierad.
Omvänt, om γ är löst, välj $x_i = 1$ om c_i ligger rätt väg, $x_i = 0$ om c_i ligger fel väg. OBS: Vad gör vi om extrakortet är vänt?

Uppgift 4: Processorschemaläggning

Problemet:

- Vi har n stycken jobb, samt tillräckligt mänga processorer.
- Varje jobb tar en tidsenhet att göra.
- Det finns villkor på formen “jobb i kan inte exekveras samtidigt som jobb j”.
- Går det att schemalägga jobben på ett sådant sätt att alla jobb är klara efter 3 tidsenheter?

Visa att problemet är NP-fullständigt!

- Visa ligger i NP.
- Visa NP-svårt.
  - Vilket problem?
  - 3-färgning!

Uppgift 4: Processorschemaläggning

- Vi vill reducera 3-färgning till schemaläggningsproblemet.
- Låt hörnen i grafen motsvara jobb, samt låt en kant mellan två noter betyda att jobben ej får utföras samtidigt.
- Vi tolkar de tre färgerna som de tre tidsenheterna.
- Anropa schemaläggningsproblemet med detta indata och returnera svaret!

Korrekthet:

- Om det finns en trefärgning, så finns det en tillåten schemaläggning.
- Om det för vår reducerade instans finns en tillåten schemaläggning så motsarar denna en tillåten trefärgning.

Nästa gång

- Hur kan vi hantera NP-fullständiga problem i praktiken?
  Approximationsalgoritmer

Mina slides ligger på kurshemsidan under Kursinnehåll/Övningsanteckningar.