

## Om NP-hårde problemer

30 marts 2007

## Om NP-hårde problemer

- 1 Ja/Nej-algoritmer og -problemer
- 2 At løse vs. at afprøve
- 3 P vs. NP
- 4 Reduktioner
- 5  $\mathbb{NP}$ -fuldstændighed
- 6 Sudoku
- 7 Litteratur

- Jeg antager at
  - I ved hvad en algoritme er
  - I ved hvordan man beregner køretiden af en algoritme
- Jeg antager *ikke* at I ved noget om Turing-maskiner (Og jeg vil ikke bruge noget om Turing-maskiner her)
- **Definition:** En algoritme er en **Ja/Nej-algoritme** hvis den som output kun har "Ja" hhv. "Nej"
- Eksempler:
  - Input: heltal  $x$       Output: "Ja," hvis  $x$  kan divideres med 7, "Nej" ellers
  - Input: graf  $G$ , heltal  $k$       Output: "Ja," hvis  $G$  kan farves med  $k$  farver, "Nej" ellers
- Ikke-eksempler:
  - Input: graf  $G$       Output: antal farver som skal bruges for at farve  $G$
  - Input: ufuldstændig Sudoku      Output: udfyldt Sudoku

- **Definition:** Et **Ja/Nej-problem** er en afbildning fra en eller anden input-mængde til mængden  $\{\text{Ja}, \text{Nej}\}$
- Eksempel: afbildningen fra (mængden af grafer)  $\times \mathbb{N}$  til  $\{\text{Ja}, \text{Nej}\}$  givet ved foreskriften: "Ja," hvis  $G$  kan farves med  $k$  farver, "Nej" ellers
- (der er nogen detaljer her mht. *kodningen* af input som vi springer over)
- **Definition:** En Ja/Nej-algoritme  $A$  **løser** et Ja/Nej-problem  $Q : I \rightarrow \{\text{Ja}, \text{Nej}\}$  hvis  $A(x) = Q(x)$  for alle  $x \in I$ .
- **Definition:** Et Ja/Nej-problem  $Q$  kaldes **løsbar** hvis der findes en Ja/Nej-algoritme der løser det.
- **Vigtig sætning:** Der findes **uløsbare** Ja/Nej-problemer.
- F.eks. Posts korrespondence-problem, se [http://en.wikipedia.org/wiki/Post\\_correspondence\\_problem](http://en.wikipedia.org/wiki/Post_correspondence_problem)

- **Definition (igen):** En Ja/Nej-algoritme  $A$  **løser** et Ja/Nej-problem  $Q : I \rightarrow \{\text{Ja}, \text{Nej}\}$  hvis  $A(x) = Q(x)$  for alle  $x \in I$ .

- **Definition:** Givet et Ja/Nej-problem  $Q : I \rightarrow \{\text{Ja}, \text{Nej}\}$  og en mængde (af *certifikater*)  $J$ . En Ja/Nej-algoritme  $A$  **afprøver** ("verifies")  $Q$  hvis der

- til ethvert  $x \in I$  med  $Q(x) = \text{"Ja"}$  findes et  $y \in J$  så  $A(x, y) = \text{"Ja"}$ ,
- til ethvert  $x \in I$  med  $Q(x) = \text{"Nej"}$  ikke findes noget  $y \in J$  med  $A(x, y) = \text{"Ja"}$ .

- Eksempel:  $I = \text{Grafer} \times \mathbb{N}$ ,  $Q(G, k) = \text{"Kan } G \text{ farves med } k \text{ farver?"}$ ,  $J = \text{farvninger af grafer}$ ,  $A = \text{"Givet graf } G, \text{ tal } k \text{ og certifikat } H, \text{ se efter om } H \text{ er en } k\text{-farvning af } G"$

5/10

- **Definition:**
  - $\mathbb{P}$  er mængden af alle Ja/Nej-problemer hvortil der findes **polynomiske** Ja/Nej-algoritmer der **løser** dem.
  - $\text{NP}$  er mængden af alle Ja/Nej-problemer hvortil der findes **polynomiske** Ja/Nej-algoritmer der **afprøver** dem.
- $\mathbb{P} \subseteq \text{NP}$ , men er  $\mathbb{P} \neq \text{NP}$ ? **Ved ikke ...**

6/10

- **Definition:** Givet to Ja/Nej-problemer  $Q_1 : I_1 \rightarrow \{\text{Ja}, \text{Nej}\}$  og  $Q_2 : I_2 \rightarrow \{\text{Ja}, \text{Nej}\}$ . En algoritme  $A$  med input  $I_1$  og output  $I_2$  kaldes en **reduktion fra  $Q_1$  til  $Q_2$**  hvis

$$Q_1(x) = \text{"Ja"} \Leftrightarrow Q_2(A(x)) = \text{"Ja"} \quad \text{for alle } x \in I_1$$

- En **polynomisk reduktion** er en reduktion der kører i polynomisk tid.

7/10

- **Definition:** Et Ja/Nej-problem  $Q$  er **NP-hårdt** hvis der til ethvert problem  $Q' \in \text{NP}$  findes en polynomisk reduktion fra  $Q'$  til  $Q$ .
- $\text{NP}^{\text{C}}$  (mængden af **NP-fuldstændige problemer**) er mængden af alle problemer som
  - ligger i  $\text{NP}$  og
  - er  $\text{NP}$ -hårde.
- Eksempler på  $\text{NP}^{\text{C}}$ -problemer: graffarvning, subset-sum, **partielle latinske kvadrater** etc.
- **at vise** at et givet Ja/Nej-problem  $Q$  er  $\text{NP}$ -hårdt: Opskriv en **reduktion** fra et andet Ja/Nej-problem  $Q'$  som er  $\text{NP}$ -hårdt til  $Q$ .

8/10

- Lad  $I$  være mængden af alle  $n^2 \times n^2$ -matricer hvor nogen indgange er fyldt ud med heltal mellem 1 og  $n^2$ . Lad  $Q : I \rightarrow \{\text{Ja}, \text{Nej}\}$  være problemet

$Q(x) = \text{"Ja"}$   $\Leftrightarrow$  matricen  $x$  kan fyldes ud til en Sudoku

- **Sætning:**  $Q \in \text{NP}$  (nemt at vise)
- **Sætning:**  $Q$  er  $\text{NP}$ -hårdt.  
Ikke så nemt at vise; f.eks. ved at reducere partielle-latinske-kvadrater-problemet til  $Q$ .

9 / 10

- om  $\text{P}$ ,  $\text{NP}$  og  $\text{NPC}$ : Cormen, Leiserson, Rivest.  
*Introduction to Algorithms*. Kapitel 36
- om reduktion af LATIN til Sudoku:
  - <http://web.archive.org/web/20060521153500/http://www.dcs.warwick.ac.uk/~pwg/cs301/sudoku.html>
  - <http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/MasterThesis.pdf>