Komplexitätstheorie

NP-Vollständigkeit: Reduktionen (2)
Der Satz von Cook/Levin

Helmut Veith
Technische Universität München

Organisatorisches

Anmeldung zur Lehrveranstaltung:
complexity@tiki.informatik.tu-muenchen.de
Subject: Teilnehmer
Subject: Schein

Übungen
3 Abgabetermine: 1.12., 8.1., 10.2.
Maximal 100 Punkte
Gruppenarbeit erlaubt

Zusatzpunkte fuer Vorlesungsausarbeitungen.
Muendliche Pruefung.

Betrachtete NP Probleme

Vertex Cover
Independent Set
3Colorability
Clique
Planar 3Colorability
SAT
Rucksack
Circuit SAT
Bin Packing
Hamiltonian Cycle
TSP

Boolesche Bedingungen

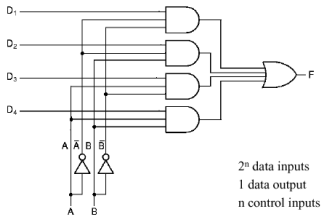
INSTANCE
C Source Code der Form

```
if (f) p;
```

wobei f aus Booleschen Variablen, &&, ||, ! besteht.

QUERY
Ist f eine erfüllbare Bedingung ?
d.h. gibt es eine Belegung der Variablen, sodass p ausgeführt wird ?

Schaltkreise



Uniforme Schaltkreise

Theorem

Zu jedem polynomialen Algorithmus (PTIME Algorithmus) A mit binären Eingaben und Ausgabe 0 oder 1 gibt es eine Familie von Schaltkreisen C_1, C_2, …, sodass

(1) C_i in Zeit polynomiell in i konstruiert werden kann.

(2) Für alle Eingaben der Länge n, der Schaltkreis C_i dasselbe Ergebnis wie A liefert.

SAT-Probleme

CIRCUIT SAT Ist ein Schaltkreis erfüllbar ?

SAT Ist eine Formel erfüllbar ?

CNF SAT Ist eine CNF Formel erfüllbar ?

3SAT Ist eine CNF Formel mit maximal 3 Literalen pro Klause erfüllbar ?

SAT, CNFSAT, ...

INSTANCE: Boolesche Funktion

CIRCUIT SAT: Schaltkreis F

SAT: Boolesche (aussagenlogische) Formel F

CNFSAT: Boolesche Formel in konjunktiver Normalform

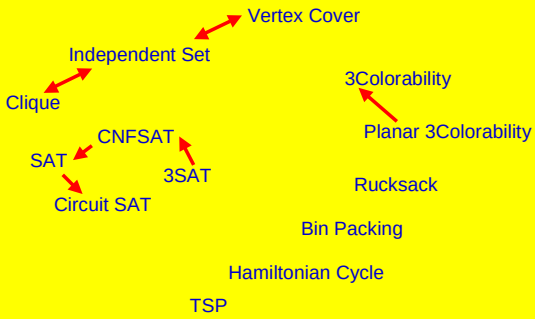
3SAT: Boolesche Formel in konjunktiver Normalform mit maximal 3 Literalen pro Klausel

$$(z_{11} \vee z_{12} \vee z_{13}) \wedge$$
$$(z_{21} \vee z_{22} \vee z_{23}) \wedge$$
$$(z_{31} \vee z_{32} \vee z_{33}) \wedge$$
$$(z_{41} \vee z_{42} \vee z_{43}) \wedge$$
$$\ldots$$

QUERY

Gibt es eine Belegung der Variablen, sodass f wahr wird ?

Betrachtete NP Probleme



Zum Reduktionsbegriff

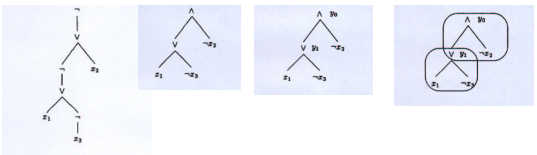
Problem A
Instance: Eingabe von A
Query: Gesuchte Eigenschaft A
Query bestimmt, ob **YES-Instance** oder **NO-Instance**

Problem B
Instance: Eingabe von B
Query: Gesuchte Eigenschaft von B

Reduktion A→B
Polynomieller Algorithmus R, sodass
W ist YES-Instance von A gdw
R(W) ist YES-Instance von B.

"B ist nicht einfacher als A".

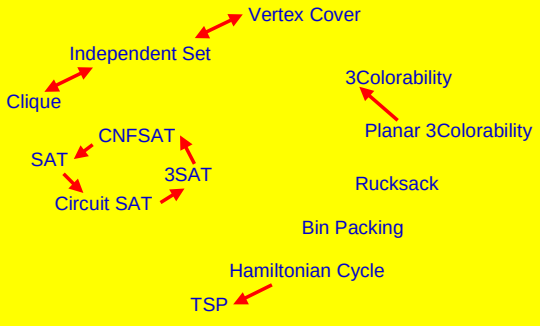
CIRCUIT SAT → 3SAT



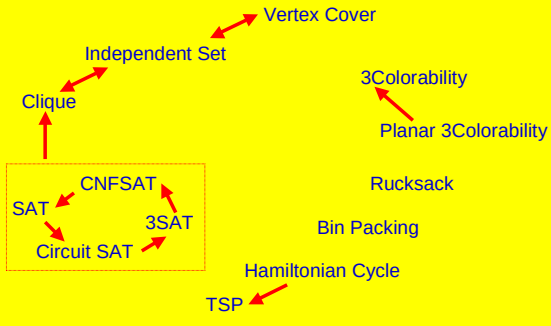
$y_0 \wedge y_0 \equiv (y_1 \wedge x_2) \wedge y_1 \equiv (x_1 \vee \neg x_3)$

$y_0 \wedge (\neg y_0 \vee y_1) \wedge (\neg y_0 \vee \neg x_2) \wedge (y_0 \vee \neg y_1 \vee x_2) \wedge$
$(y_1 \vee \neg x_1) \wedge (\neg y_1 \vee x_1 \vee \neg x_3) \wedge (y_1 \vee x_3)$

Betrachtete NP Probleme



Betrachtete NP Probleme



CLIQUE

Gibt es K Studenten unter den N Anwesenden, die einander kennen ?

INSTANCE

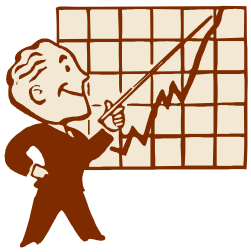
Graph G=(V,E); Zahl K.

QUERY

Hat G eine Clique C mit | C | >= K ?

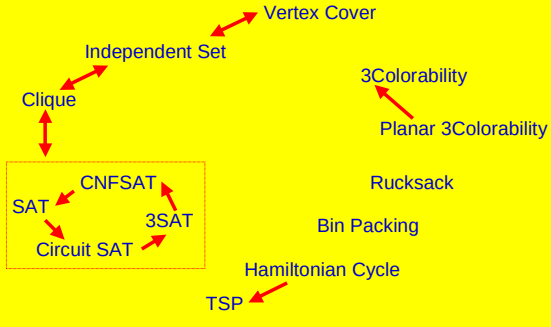
d.h. ein C, |C| >= K, sodass $\forall$ i,j $\in$ C . (i,j) $\in$ E

3SAT → CLIQUE



CLIQUE → CIRCUIT SAT

Betrachtete NP Probleme



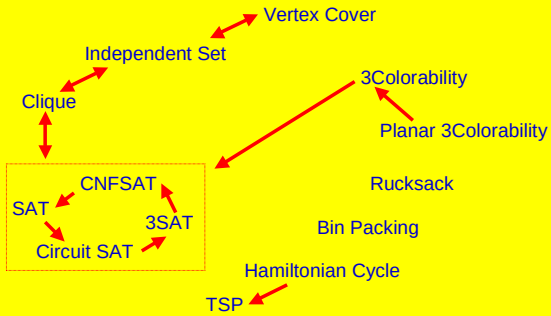
3COLORABILITY

Reichen 3 Übungsgruppen, damit Studenten, die einander kennen, verschiedene Gruppen erhalten ?

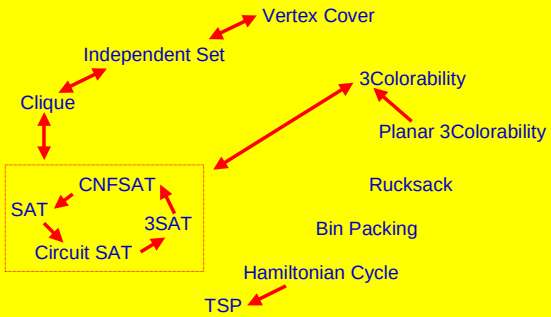
INSTANCE
Graph G=(V,E).

QUERY
Gibt es eine Funktion
F : V → {red,green,blue}
sodass benachbarte Knoten unterschiedlich gefaerbt werden, d.h.
∀ (i,j) ∈ E . F(i) ≠ F(j)

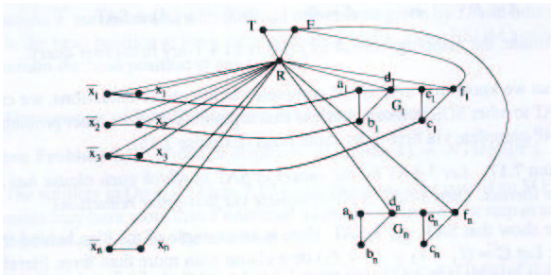
Betrachtete NP Probleme



Betrachtete NP Probleme



3SAT → 3COL



Bemerkung

Ein Problem X in NP kann meist einfacher auf SAT reduziert werden als umgekehrt !

Cook / Levin 1971

Alle Probleme in NP können auf SAT reduziert werden !

Uniforme Schaltkreise

Theorem
Zu jedem polynomialen Algorithmus (PTIME Algorithmus) A mit binären Eingaben und Ausgabe 0 oder 1 gibt es eine Familie von Schaltkreisen $C_1, C_2, \ldots$, sodass
(1) C_i in Zeit polynomiell in i konstruiert werden kann.
(2) Für alle Eingaben der Länge n, der Schaltkreis C_n dasselbe Ergebnis wie A liefert.

Idee: Verwende den Uniformitätssatz zur Konstuktion einer CIRCUIT SAT Instanz.

Cook / Levin 1971

Theorem Alle Probleme in NP können auf SAT reduziert werden.

Definition A ist NP-vollständig, wenn A in NP ist, und alle NP-Probleme auf A reduziert werden können.

Korollar (Satz von Cook/Levin)
SAT ist NP-vollständig.

Strategie: NP-Vollständigkeit

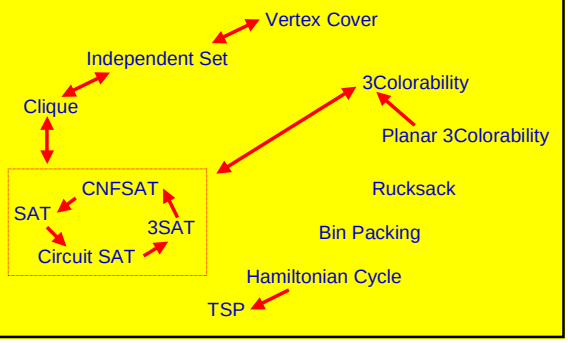
Lemma
Wenn A, B in NP sind, A NP-vollständig und
A→ B, dann ist B NP-vollständig.

Beweisidee: Polynomieller Algorithmus für B kann auch mittels der Reduktion für A verwendet werden.

Analysestrategie für neues Problem X:

1) Zeige, dass X in NP ist: Algorithmus.
2) Finde ein bekanntes NP-vollständiges Problem Y, und zeige, dass Y → X gilt.

Betrachtete NP Probleme



Folgen von NP-Vollständigkeit

Alle vollständigen Probleme sind wechselseitig reduzierbar.

Ein polynomieller Algorithmus für ein vollständiges Problem impliziert einen Algorithmus für alle Probleme in NP.

Zu zeigen: Uniformitätssatz.

Uniforme Schaltkreise

Theorem

Zu jedem polynomialen Algorithmus (PTIME Algorithmus) A mit binären Eingaben und Ausgabe 0 oder 1 gibt es eine Familie von Schaltkreisen C_1, C_2, …, sodass

(1) C_i in Zeit polynomiell in i konstruiert werden kann.

(2) Für alle Eingaben der Länge n, der Schaltkreis C_i dasselbe Ergebnis wie A liefert.

Danke für Ihre Aufmerksamkeit