Reduktion $3SAT \leqslant_p IS$

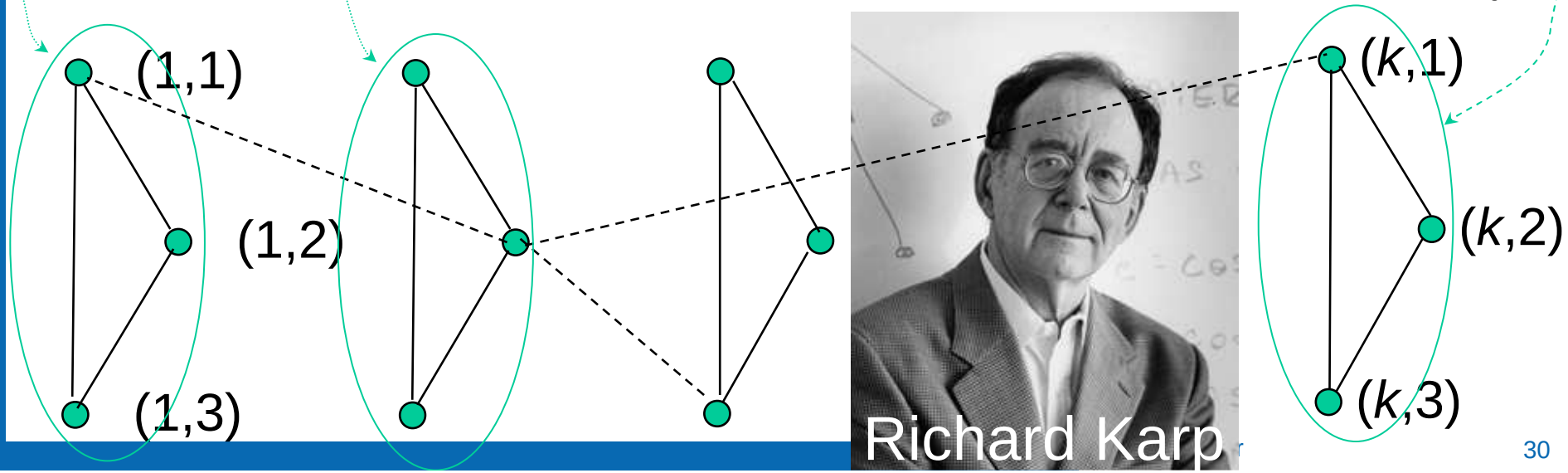
TECHNISCHE UNIVERSITÄT DARMSTADT

Berechne in polynomieller Zeit aus einer 3-KNF Formel Φ einen Graphen G und eine Zahl k, so dass gilt: Φ *ist genau dann erfüllbar, wenn es in* G k *unabhängige Knoten gibt.*

z.B. $(u \vee .. \vee ..) \wedge (.. \vee \neg u \vee ..) \wedge (.. \vee .. \vee u) \wedge (u \vee .. \vee ..)$

$\Phi = C_1 \wedge C_2 \ldots \wedge C_k$, $C_i = x_{i1} \vee x_{i2} \vee x_{i3}$, x_{is} Literale

$V := \{ (i,1), \ldots (i,3) : i \leq k \}$, $E := \{ \{(i,s),(j,t)\} : i=j$ oder $\overline{x}_{is} = x_{jt} \}$



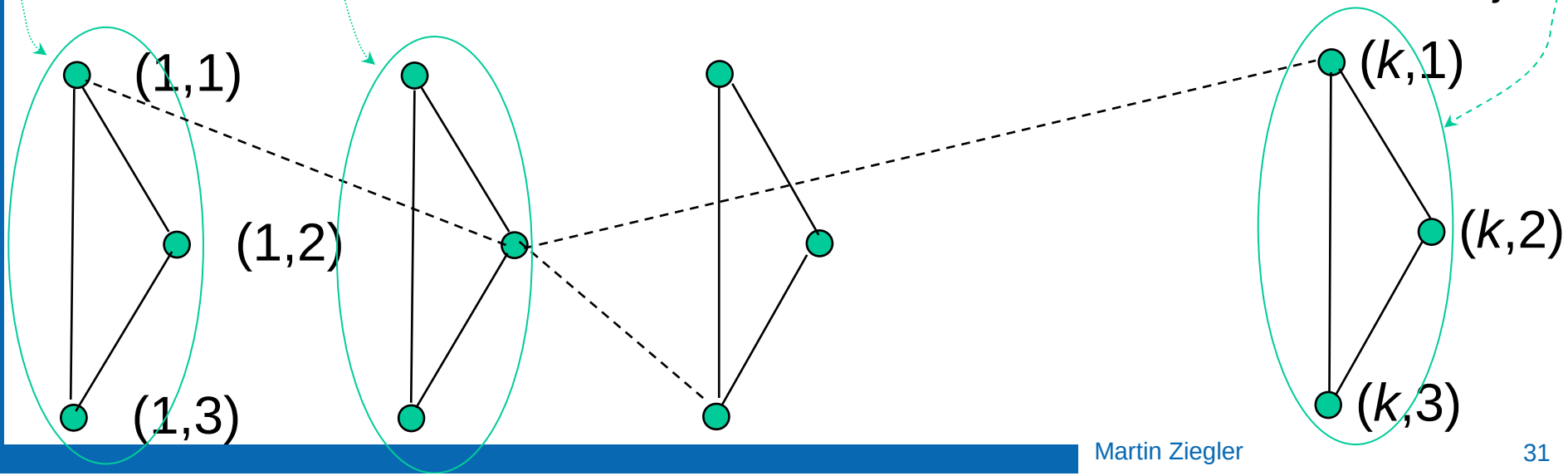
Reduktion $3SAT \leqslant_p IS$

„$\Rightarrow$“ Sei x erfüllende Belegung von Φ. Dann ist in jeder Klausel mindestens ein Literal **wahr**. Die zugehörigen Knoten in G sind nicht miteinander verbunden.

„$\Leftarrow$“ Seien $(v_1,..,v_k)$ unabhängig in G. Dann gehört zu jedem v_i ein Literal x_{is} in Φ. Diese haben alle gleiches Vorzeichen. Belege sie mit **wahr**: konsistent!

$\Phi = C_1 \wedge C_2 \ldots \wedge C_k$, $C_i = x_{i1} \vee x_{i2} \vee x_{i3}$, x_{is} Literale

$V := \{ (i,1),\ldots(i,3): i \leq k \}$, $E := \{ \{(i,s),(j,t)\} : i=j$ oder $\overline{x}_{is} = x_{jt} \}$



Viele gleich schwere Probleme

Gezeigt: $\mathbf{CLIQUE} \equiv_p \mathbf{IS} \leqslant_p \mathbf{SAT} \equiv_p \mathbf{3SAT} \leqslant_p \mathbf{IS}$.

Diese 4 Probleme sind *alle* ungefähr gleich schwer:

Entweder alle sind in $\mathcal{P}$ oder keines.

Wir werden noch zeigen: auch **TSP**, **HC**, **VC** und viele weitere Probleme in $\mathcal{NP}$ gehören zu dieser Klasse, genannt $\mathcal{NP}$c.

Und wir werden zeigen: Dies sind die ‚schwersten' Probleme in $\mathcal{NP}$: Für jedes $L \in \mathcal{NP}$ gilt: $L \leqslant_p \mathbf{SAT}$. (*Satz von Cook*)

D.h. wenn a) irgendwer einen polynomialzeit-Algorithmus für irgendein Problem aus $\mathcal{NP}$c fände, so folgte $\mathcal{P}=\mathcal{NP}$:

Eine DTM könnte jede NTM in Polynomialzeit simulieren!

Und umgekehrt: Aus einem Beweis b), daß irgendeines dieser Probleme nicht in Polynomialzeit lösbar ist, folgt $\mathcal{P}\neq\mathcal{NP}$:

kein Problem in $\mathcal{NP}$c ließe sich in Polynomialzeit lösen.

die Klasse $\mathcal{NP}$

Def: Eine Sprache $L \subseteq \Sigma^*$ gehört zur Klasse $\mathcal{NP}$, wenn es $K \in \mathcal{P}$ und $p(N) \in \mathbb{N}[N]$ gibt mit:

$$L = \{ x : \exists\, y \in \Sigma^{\leq p(|x|)} : \langle x,y \rangle \in K \}$$

Beispiele:

- $\mathcal{P} \subseteq \mathcal{NP}$
- $\mathrm{SAT} \in \mathcal{NP}$
- $\mathrm{HC} \in \mathcal{NP}$
- $\mathrm{VC} \in \mathcal{NP}$
- $\mathrm{TSP} \in \mathcal{NP}$
- $\mathrm{IS} \in \mathcal{NP}$

Übung: $\mathcal{NP}$ via nichtdeterministische Turingmaschinen

$\mathcal{NP}$-Vollständigkeit

A heißt *polynomiell* reduzierbar auf *B* („$A \preceq_p B$"),

falls es eine in *polynomieller* Zeit berechenbare totale Funktion $f:\Sigma^* \to \Sigma^*$ gibt mit $\underline{x} \in A \Leftrightarrow f(\underline{x}) \in B \quad \forall \underline{x} \in \Sigma^*$.

Lemma: a) $A \preceq_p B$ und $B \in \mathcal{P} \Rightarrow A \in \mathcal{P}$

b) $A \preceq_p B$ und $B \preceq_p C \Rightarrow A \preceq_p C$ (*Transitivität*)

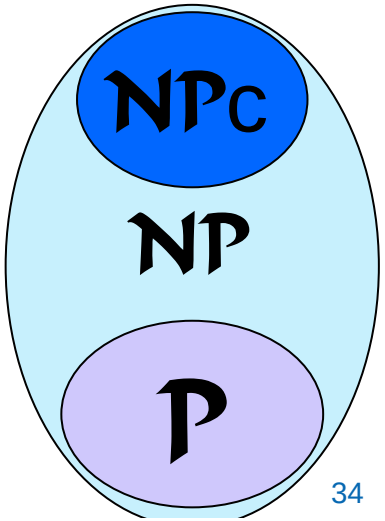
L heißt $\mathcal{NP}$-schwer, falls für jedes $A \in \mathcal{NP}$ gilt: $A \preceq_p L$.

L heißt $\mathcal{NP}$-vollständig, falls $\mathcal{NP}$-schwer und $\in \mathcal{NP}$ ist.

Bem: Ist ein *L* $\mathcal{NP}$-vollständig und $L \in \mathcal{P}$,

so folgt $\mathcal{P} = \mathcal{NP}$ (und 1 000 000 $US)

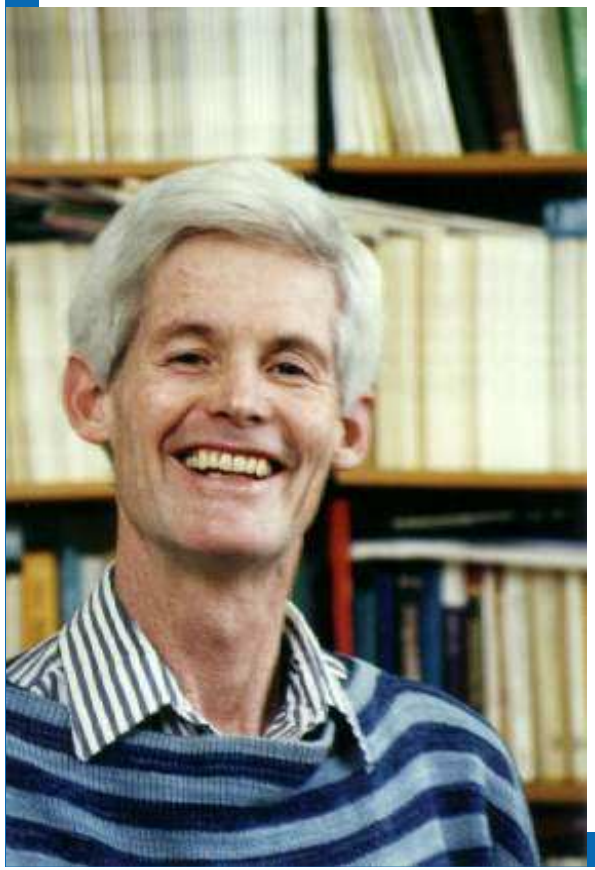
Korollar: Falls $\mathcal{NP} \neq \mathcal{P}$ gilt, dann sind alle $\mathcal{NP}$-vollständigen Sprachen in $\mathcal{NP} \setminus \mathcal{P}$, also insbesondere nicht in $\mathcal{P}$.



Die Master-Reduktion

Satz von Cook-Levin (1971/72)

SAT ist $\mathcal{NP}$-vollständig.



Zu zeigen:

- $\mathbf{SAT} \in \mathcal{NP}$

  (schon erledigt)

- Für jedes $L \in \mathcal{NP}$ gilt:

  $L \leqslant_p \mathbf{SAT}$

Beweisidee

$\forall L \in \mathcal{NP}: L \leqslant_p SAT$

Sei $L \in \mathcal{NP}$, $N=(Q, \Sigma, \Gamma, \delta, q_0, F)$ eine 1-NTM,

die L in Zeit $T(n)$ entscheidet für ein Polynom T.

O.B.d.A.: N macht immer genau $T(n)$ Schritte

Aufgabe: Beschreibe eine in polynomieller Zeit berechenbare Funktion f, die bei Eingabe $\underline{w} \in \Sigma^*$

eine Boole'sche Formel $\Phi_{\underline{w}}=f(\underline{w})$ liefert, so dass gilt:

N akzeptiert $\underline{w}$ $\Leftrightarrow$ $\Phi_{\underline{w}}$ ist erfüllbar

Idee: Konstruiere Formel Φ so, dass erfüllende Belegungen für Φ zu akzeptierenden Rechnungen von N korrespondieren.

Boole'sche Variablen von Φ und ihre intuitive Bedeutung:

$d_{s,a,t}$: „Nach Schritt t steht in Bandzelle #s das Symbol a"

$h_{s,t}$: „Nach Schritt t steht der Kopf auf Zelle #s"

$z_{q,t}$: „Nach Schritt t ist N im Zustand q"

$a \in \Gamma$, $q \in Q$

$s=1..S(|\underline{w}|)$, $t=0\ldots T(|\underline{w}|)$

Beweisskizze

TECHNISCHE UNIVERSITÄT DARMSTADT

Konstruiere Formel $\Phi_{\underline{w}}$ so, dass erfüllende Belegungen für Φ zu akzeptierenden Rechnungen von N korrespondieren.

Boole'sche Variablen von Φ und ihre intuitive Bedeutung:

$d_{s,a,t}$: „Nach Schritt t steht in Bandzelle s das Symbol a“ $a \in \Gamma$, $q \in Q$

$h_{s,t}$: „Nach Schritt t steht der Kopf auf Zelle s“ $s=1..S(|\underline{w}|)$

$z_{q,t}$: „Nach Schritt t ist die Maschine im Zustand q“ $t=0\ldots T(|\underline{w}|)$

$T(n)\cdot S(n)\cdot|\Gamma| + T(n)\cdot S(n) + |Q|\cdot T(n)$ = polynomiell viele Var.en

Jede Rechnung von N gestartet mit $\underline{w}$ kann durch passende Belegung der Var.en beschrieben werden.

Belegungen können aber auch Unsinn beschreiben.

Ziel: Entwerfe KNF Φ, die genau für diejenigen Belegungen der Variablen **wahr** wird, die eine akzeptierende Rechnung von N beschreiben.