

# Parameterisierte Algorithmen

WS 2007/08 in Trier

Henning Fernau

[fernau@uni-trier.de](mailto:fernau@uni-trier.de)

# Parameterisierte Algorithmen

## Gesamtübersicht

- Einführung
- Grundbegriffe
- Problemkerne
- Suchbäume
- Graphparameter
- Weitere Methoden
- Komplexitätstheorie—parameterisiert

## Aufzählungsprobleme

In manchen Anwendungen gefragt:

Auflistung aller minimalen Mengen einer vorgegebenen Obergrenze

oder:

Auflistung aller maximalen Mengen einer vorgegebenen Untergrenze

~> auch hier FPT-Algorithmen (parameterisiert nach Grenzgröße) von Interesse

**Satz 1** *Der aufgeführte Algorithmus HSEnum listet bei Eingabe  $(G, k, \emptyset)$  alle minimalen HS der Maximalgröße  $k$  in Zeit  $\mathcal{O}^*(d^k)$ ; hierbei ist  $d$  die Anzahl der Elemente der größten Hyperkante von  $G$ .*

---

**Algorithm 1** A search tree algorithm HSEnum enumerating HS

---

**Input(s):** a hypergraph  $G = (V, E)$ , a positive integer  $k$ , a vertex set  $X$

**Output(s):**  $\mathcal{S}$ : a set of HSs of  $G$  s.t. each minimal HS of size  $\leq k$  appears in  $\mathcal{S}$

**if**  $E = \emptyset$  **then**

    return  $\{X\}$

**else if**  $k = 0$  **then**

    return  $\emptyset$

**else**

    Choose edge  $e \in E$

$\mathcal{S} = \emptyset$

**for**  $v \in e$  **do**

$\mathcal{S} = \mathcal{S} \cup \text{HSEnum}(G[V \setminus \{v\}], k - 1, X \cup \{v\})$

**end for**

    return  $\mathcal{S}$

**end if**

**Induktionsbeweis** Wir beweisen eine stärkere Aussage:

HSEnum( $G, k, X$ ) liefert ein Mengensystem  $\mathcal{S}$  zurück, sodass gilt:

(i)  $\forall S \in \mathcal{S}: X \subseteq S$  und  $S \setminus X$  ist HS von  $G$  der Mächtigkeit  $\leq k$ .

(ii)  $\forall S: S$  minim. HS von  $G, |S| \leq k \Rightarrow \exists X : S \cup X \in \mathcal{S}$ .

IA: Dies gilt für  $E = \emptyset$ , d.h.,  $|E| = 0$ .

IS: Angenommen,  $|E| > 0$  und die Aussage gelte für Hypergraphen mit weniger als  $|E|$  Kanten.

Es sei  $\mathcal{S}$  das bei Eingabe von  $G = (V, E)$  (etc.) zurückgelieferte Mengensystem.

Für (i) ist  $\mathcal{S} = \emptyset$  trivial; betrachte also  $S \in \mathcal{S}$  beliebig.

Für die ausgewählte Kante  $e \in E$  sei  $\mathcal{S}_v$  das für den Fall  $v \in e$  durch Aufruf HSEnum( $G[V \setminus \{v\}], k - 1, X \cup \{v\}$ ) zurückgelieferte Mengensystem.

$IV \rightsquigarrow X \cup \{v\} \subseteq S_v \subseteq S$  und  $S \setminus (X \cup \{v\})$  ist HS von  $G[V \setminus \{v\}] = (V_v, E_v)$ .

$IV \rightsquigarrow \forall e' \in E_v : e' \cap S \setminus X \neq \emptyset$ ; wegen  $v \in S$  gilt auch:  $\forall e' \in E \setminus E_v : e' \cap S \setminus X \neq \emptyset$

$\rightsquigarrow S \setminus X$  ist HS von  $G$  (mit höchstens  $k$  Knoten).

Für (ii) betrachte ein minimales HS  $S$  von  $G$  mit  $|S| \leq k$ .

Wegen  $E \neq \emptyset$  gilt  $k > 0$  (sonst "Fehlerfall").

Betrachte gewählte Kante  $e \in E$  sowie  $v \in e \cap S$  (ex., da  $S$  HS).

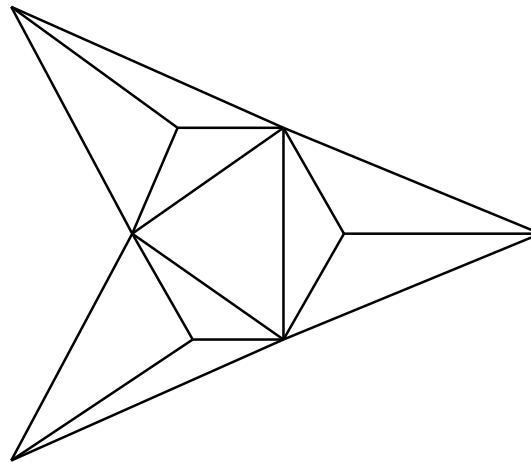
$S_v$  sei wieder das Ergebnis des Aufrufs mit  $G[V \setminus \{v\}] = (V_v, E_v)$ .

$S \setminus \{v\}$  ist minimales HS von  $(V_v, E_v)$ .

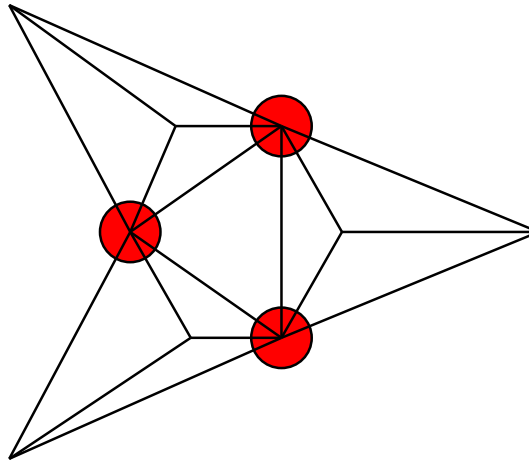
$IV \rightsquigarrow S \cup X = (S \setminus \{v\}) \cup (X \cup \{v\}) \in S_v \subseteq S$ .

## Erinnerung: Dominierende Mengen

Wir werden noch sehen: Das Problem DS, dominierende Mengen mit höchstens  $k$  Knoten in einem Graphen zu finden, liegt “wahrscheinlich” nicht in *FPT*. (W[2]-hart)



**... mit einer zulässigen dominierenden Menge**



**Ist diese dominierende Menge kleinstmöglich ?**

## Kantengraphen (line graphs)

Adjazenzrelation aus Kantennachbarschaft eines anderen Graphen, genauer:

Ist  $G = (V, E)$  ein Graph, so hat sein **Kantengraph**  $L(G)$

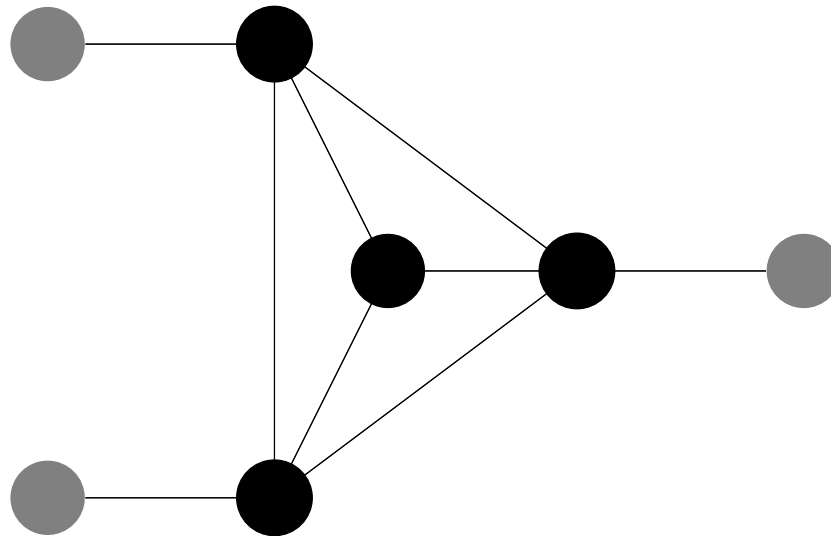
$E$  als “Knotenmenge” und

es gibt eine Kante (in  $L(G)$ ) zwischen  $e_1, e_2 \in E$

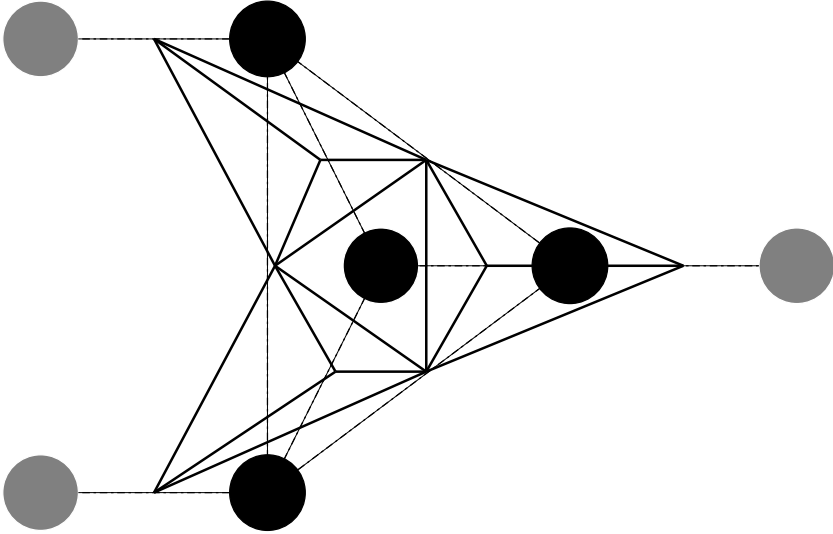
gdw.  $e_1$  und  $e_2$  haben gemeinsamen Knoten in  $G$ .



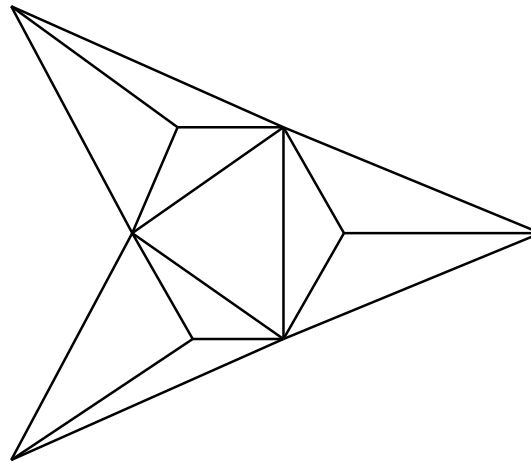
## Ein einfacher Graph



**... und sein Kantengraph**



**... liefert unser Beispiel**



## Unser Entscheidungsproblem:

DS auf Kantengraphen; für “normale” Graphen heißt dies übersetzt:

KANTENDOMINIERUNGSPROBLEM (EDS)

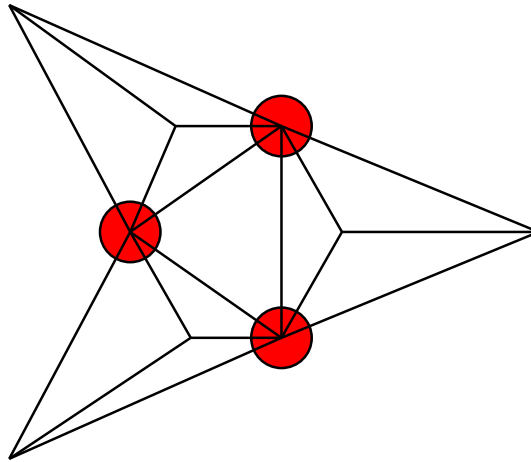
**Eingabe:** Ein Graph  $G = (V, E)$

**Parameter:** ein  $k \in \mathbb{N}$

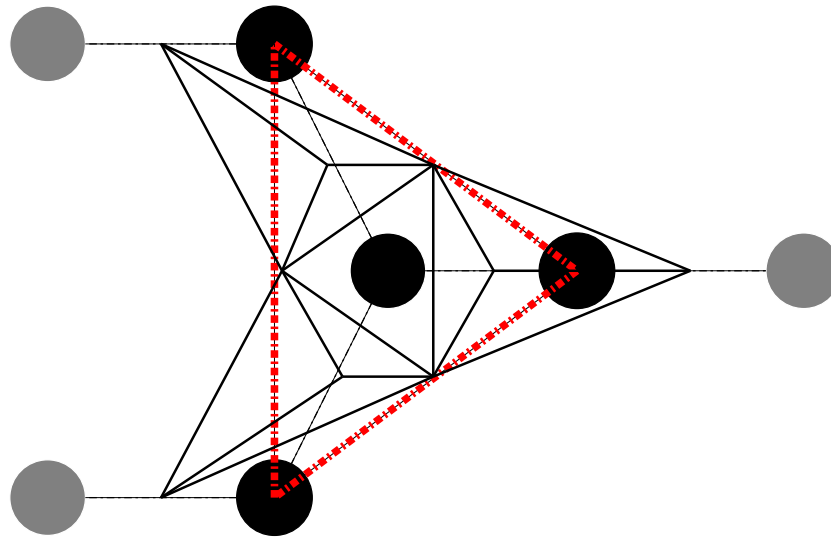
**Frage:** Gibt es eine Kantendominierungsmenge  $D \subseteq E$  mit  $|D| \leq k$ ?

Dabei heißt  $D$  **Kantendominierungsmenge** von  $G = (V, E)$ , wenn es zu jedem  $e \in E$  ein  $d \in D$  gibt mit  $e \cap d \neq \emptyset$ .

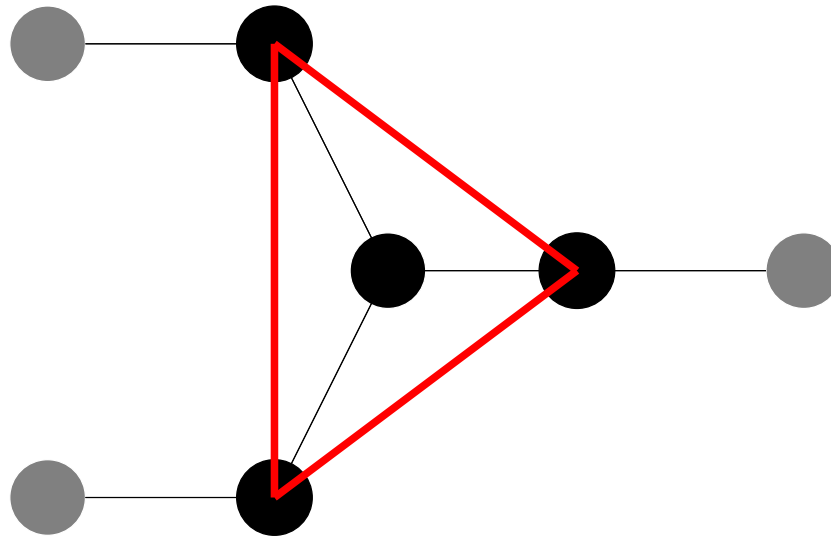
## Die dominierende Menge



**... wird übersetzt in eine Kantendominierungsmenge:**



**... etwas deutlicher:**



## Kantengraphen machen das Leben manchmal einfacher...

Eine klassische Sicht

	allg. Graphen	Kantengraphen
VC	<i>NP</i> -hart	in <i>P</i>
DS	<i>NP</i> -hart	<i>NP</i> -hart



## Kantengraphen machen das Leben manchmal einfacher...

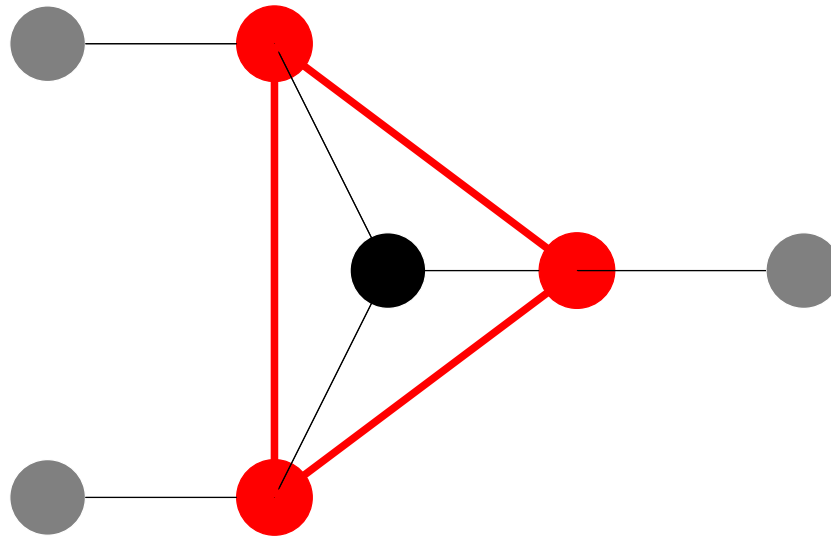
Approximation und *FPT*(für (E)DS)

	allg. Graphen	Kantengraphen
Approximation	nicht besser als $\ln n$	Faktor-2-Approx.
Parameterisierung	$W[2]$ -hart	in <i>FPT</i>

## Eine einfache Beobachtung

$D \text{ EDS} \rightsquigarrow$

alle Knoten von Kanten in  $D$  zusammen ergeben eine Knotenüberdeckung



## Wieso liegt nun EDS in *FPT*?

**Beobachtung 1:** Ein EDS der Größe  $k$  liefert VC der Größe  $\ell$ ,  $k \leq \ell \leq 2k$ .

**Beobachtung 2:** Wir können alle minimalen VC bis zu der Größe  $2k$  auflisten, (aber nicht alle VC bis zur Größe  $2k$ ).

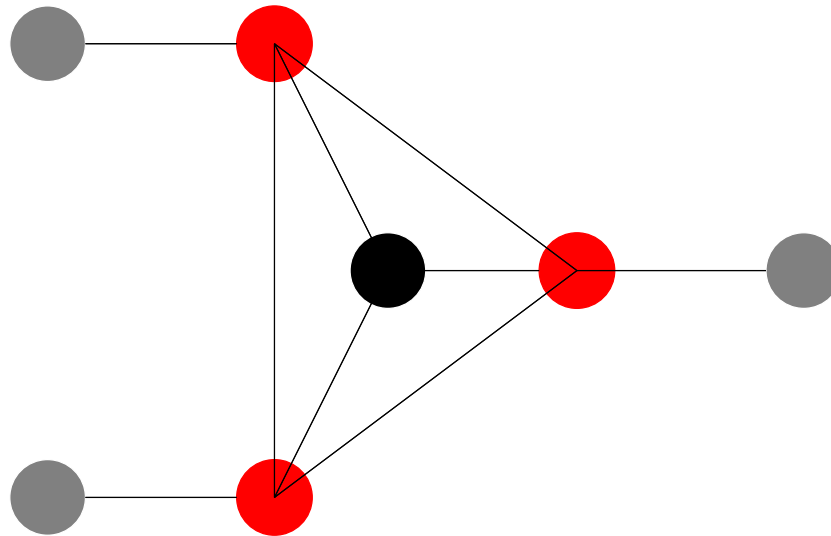
**Problem:** Wie hilft dies ?

**Entscheiden durch Aufzählen**

## Was benötigen wir...

Eingabe: Ein (minimales) VC  $C$

Ausgabe: Ein kleinstmögliches EDS  $D$ , dessen Knotenmenge  $C$  umfasst



## Was der Algorithmenladen so auf Lager hat...

Betrachte Hilfs-Hypergraph  $G' = (V', E')$ :

$G'$  enthält die Kanten  $E$  von  $G$  als seine Knoten, d.h.,  $V' = E$ , und für jedes  $x \in C$  führen wir eine Hyperkante  $h_x$  ein, welche alle Kanten von  $G$  enthält, die inzident mit  $x$  sind.

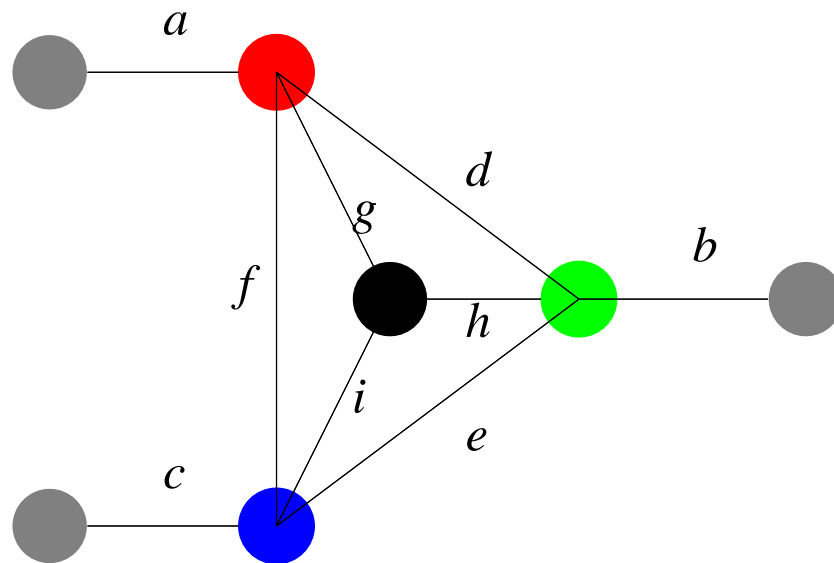
Offenbar:  $|E'| \leq |C|$

Gute Nachricht aus dem Algorithmenladen:

HS, parameterisiert nach der Anzahl der Kanten, ist in *FPT*. (s.u.)

**Folgerung 2** *EDS kann in Zeit  $\mathcal{O}^*(2^{2k} * 2^{2k}) = \mathcal{O}^*(16^k)$  gelöst werden.*

## Zurück zu unserem Beispiel



$$h = \{a, d, f, g\}$$

$$h = \{b, d, e, h\}$$

$$h = \{c, f, e, i\}$$

Offenbar gibt es keine Lösung der Größe Eins.

ABER:  $f$  zusammen mit irgendeinem "Knoten" aus  $h$  liefert Lösung der Größe zwei.

Nach dem Gesagten ist dies eine kleinstmögliche Lösung.

**Speedy...**



**Beobachte:** Die gesuchte Kantendominierungsmenge enthält ein Maximum Matching für die vorzuliegende Knotenüberdeckungsmenge.

**Warum?** Eine Matchingkante benutzt zwei Knoten der Überdeckung, nicht nur einen, und führt daher bei vorzuliegender Überdeckung zu einer kleinstmöglichen Kantendominierungsmenge.

**Bekannt:** Solche Matchings kann man in Polynomzeit berechnen.

**Folgerung 3** *EDS ist in Zeit  $\mathcal{O}^*(2^{2k}) = \mathcal{O}^*(4^k)$  lösbar.*

**Einzelheiten:** Verallgemeinertes Kantenüberdecken

VERALLGEMEINERTES KANTENÜBERDECKUNGSPROBLEM (GEC)

**Eingabe:** Ein Graph  $G = (V, E)$ , eine Menge  $R \subseteq V$  von roten Knoten

**Parameter:**  $k \in \mathbb{N}$

**Frage:** Gibt es eine Kantenüberdeckungsmenge  $C \subseteq E$  mit  $|C| \leq k$ , welche alle Knoten aus  $R$  abdeckt ?

**GECmin** ist das zugehörige Minimierungsproblem.

**Lemma 4** *GECmin kann in Polynomzeit gelöst werden.*



---

**Algorithm 2** GECmatch: A fast algorithm for GECmin, based on matching

---

**Input(s):** a graph  $G = (V, E)$ , a set of red vertices  $R$

**Output(s):** a minimum-size set of edges  $C$  that covers all vertices from  $R$

Create  $H = G[R]$ .

Find a maximum matching  $M \subseteq E$  in  $H$ .

Let  $V'$  be all vertices in  $H$  that are not covered by  $M$ .

Let  $C := M$ .

**for all**  $v \in V'$  **do**

    Choose an edge  $e \in E$  such that  $v \in e$ .

    Add  $e$  to  $C$ .

**end for**

return  $C$

---

---

**Algorithm 3** EDS-enum: An enumeration-based search tree algorithm for EDS

---

**Input(s):** a graph  $G = (V, E)$ , a positive integer  $k$

**Output(s):** if possible: a subset  $D \subseteq E$ ,  $|D| \leq k$ , that dominates all edges or  
NO if no such set exists.

Create a list  $L$  of minimal vertex covers  $C$  of  $G$  with  $|C| \leq 2k$

{This hides the search tree part.}

**for all**  $C \in L$  **do**

    Set  $D := \text{GECmatch}(G, C)$

**if**  $|D| \leq k$  **then**

        return  $D$

**end if**

**end for**

return NO

---

**Noch schneller...**



**Idee:** Vermeide Verzweigungen auf Knoten vom Grad Eins in der Knotenüberdeckungs-Aufzählungsphase.

~> An den Blättern des Aufzählungs-Suchbaums finden wir eine partielle Überdeckung sowie eine Menge  $E'$  von isolierten Kanten.

Hat  $e' \in E'$  keine inzidenten Kanten im Originalgraph  $G$ , so nimm  $e'$  ins EDS.

Sonst: kontrahiere  $e' = \{x, y\}$  und füge  $[x = y]$  zum partiellen VC hinzu.

**Folgerung 5** EDS kann man in Zeit  $\mathcal{O}^*(1.62^{2k}) = \mathcal{O}^*(2.62^k)$  lösen.

## Noch genauer... (Eingabegraph $G = (V, E)$ )

Eingangs können wir alle isolierten Kanten ins EDS nehmen, also hat  $G$  o.E. keine isolierten Kanten.

Durch Aufzählungsphase erhalten wir partielle Knotenüberdeckung  $C \subseteq V$  und Kantenmenge  $E'$ .  
Danach suchen wir EDS  $D \subseteq E$ , welches erfüllt:

A  $\forall v \in C \exists e \in D : v \in e$ ; (Abdeckung)

D  $\forall e' \in E' \exists e \in D : e' \cap e \neq \emptyset$  (Dominierung).

Da  $e'$  nicht isoliert ist in  $G$ , wird  $e'$  o.E. durch  $e \neq e'$  dominiert.

Bilde neuen Graphen  $G'$  aus  $G$  durch Kontraktion aller Kanten von  $E'$ .

Sei  $M$  die Menge der "Kontraktionsknoten"  $[x, y]$ .

Bilde  $C' = C \cup M$ .

**Beh.:**  $D$  ist kleinstmögliches EDS für  $G$ , welches A und D erfüllt gdw. es gibt kleinstmögliches GEC  $D'$  for  $G'$  (mit roter Knotenmenge  $C'$ ) mit  $|D'| = |D|$ .

## Entscheiden durch Aufzählen

Diese Technik ist insbesondere geeignet für “schnelle” Algorithmenentwürfe.

~> Klassifizierung eines Problems in *FPT*

Da Aufzählungsphase relativ teuer, sollte nach Beschleunigungsmöglichkeiten gesucht werden.

Oft hilft auch, die Aufzählung nicht “bis zu Ende” durchzuführen.

## Ähnliche Probleme I

- Gewichtetes Kantendominierungsproblem
- Auffinden von maximalen Matchings der Größe  $\leq k$  (Min. max. matching)
- MATRIXDOMINIERUNG (MDS)  
Eingabe:  $n \times n$  Matrix mit Einträgen aus  $\{0, 1\}$   
Parameter:  $k \in \mathbb{N}$   
Frage: Gibt es eine Menge  $D$  von Eins-Einträgen,  $|D| \leq k$ , sodass jeder Eins-Eintrag eine Zeile oder eine Spalte mit einem Eins-Eintrag aus  $D$  gemein hat ?

---

**Algorithm 4** Reducing MDS to EDS.

---

**Input(s):** a matrix instance  $(M, k)$  of MDS.

**Output(s):** a graph instance  $(G, k)$  of EDS such that  $(M, k)$  is a YES-instance iff  $(G, k)$  is a YES-instance.

Let  $C$  be the set of columns of  $M$ .

Let  $R$  be the set of rows of  $M$ .

Form the vertex set  $V = C \cup R$  of  $G = (V, E)$ .

**for all**  $i \in R, j \in C$  **do**

    Put  $\{i, j\} \in E$  iff entry  $(i, j)$  of  $M$  is one.

**end for**

---

## Ähnliche Probleme II

Gibt es parameterisierten Aufzählungsalgorithmus für ein “Grundproblem”, z.B. für das Knotenüberdeckungsproblem, so gibt es auch Entscheidungsalgorithmen, die im Beispiel Knotenüberdeckungen mit bestimmten Eigenschaften erfragen.

**Beispiel 1:** Eine Knotenüberdeckung  $C$  von  $G$  heiße **total**, wenn  $G[C]$  keine isolierte Knoten enthält.

TVC fragt nach der Existenz einer totalen Knotenüberdeckung der Größe höchstens  $k$ .

**Beispiel 2:** CVC fragt nach der Existenz einer zusammenhängenden Knotenüberdeckung der Größe höchstens  $k$ .



## Aufzählen und Zählen per se

Angesagt, falls spätere Auswahl einer Lösung erwünscht.

Es gibt auch **Aufzählungskerne**, z.B. einen quadratischen für VC.

**Frage:** Wie geht das ? Warum nicht (so leicht) linear ?!

**Verwandtes Problem:**

**Zählen** aller minimalen VC einer vorgegebenen Maximalgröße  $k$ .

Zählen kann man durch Aufzählen lösen, es geht aber oft besser (**wie ?**):

**Satz 6** VC-Zählen geht in Zeit  $\mathcal{O}^*(1.62^k)$ .

## Schlimme Suchbäume

Eine *defensive Allianz* ist eine nicht-leere Knotenmenge  $S \subseteq V$ , sodass gilt:  
 $\forall v \in S (|N[v] \cap S| \geq |N[v] \setminus S|)$ .

DA fragt nach der Existenz einer defensiven Allianz der Größe höchstens  $k$ .

**Satz 7** DA liegt in FPT.

Beweis O.E. sei die Eingabe zusammenhängend. (Sonst getrennte Suche auf Komponenten)  
Gibt es Lösung  $S$ , so gilt nach Voraussetzung  $S \neq \emptyset$ . Wir verzweigen daher eingangs **katalytisch**  
und nehmen für jeden Knoten in einem Zweig einmal an, er gehöre zu  $S$ .

Falls  $v$  in einer DA liegt, so  $\deg(v) < 2k$  (sonst Widerspruch zur Def.).

Setze  $S_0 = \{v\}$ .

Falls  $|S_i| > k$ , terminiere diesen Suchbaumast.

Falls  $|S_i| \leq k$  DA, so YES-Instanz.

Sonst: Verzweige **auf allen**  $x \in N(S_i)$ : Mache rekursiv mit  $S_{i+1} = S_i \cup \{x\}$  weiter.

**Warum FPT?**