Algorithmen II
Vorlesung am 14.11.2013

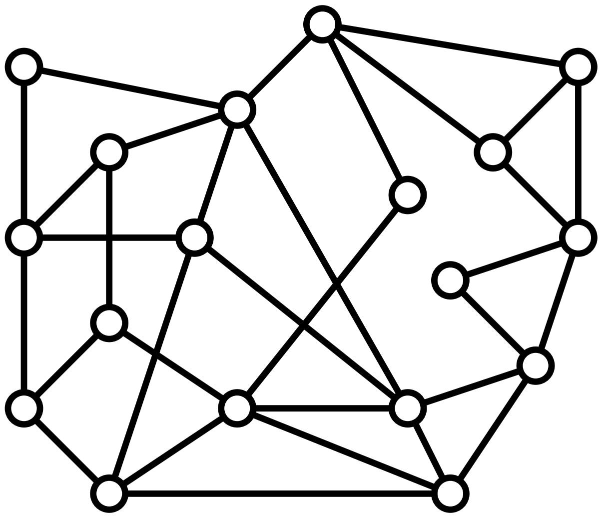
Kreisbasen, Matroide & Greedy Algorithmen

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Kreisbasen

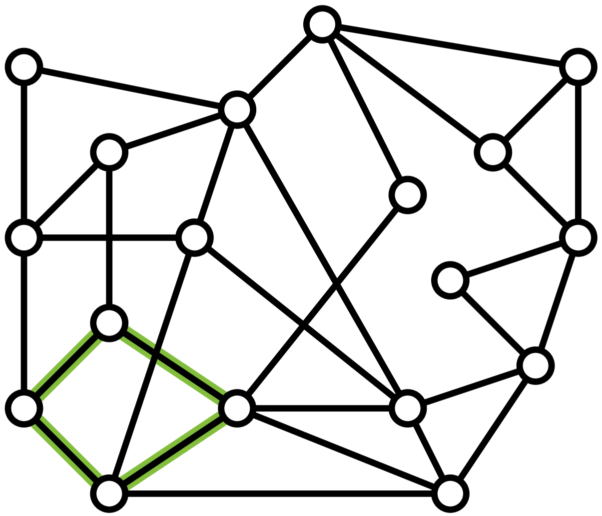
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation



- Ein Graph kann sehr viele Kreise haben.

Motivation

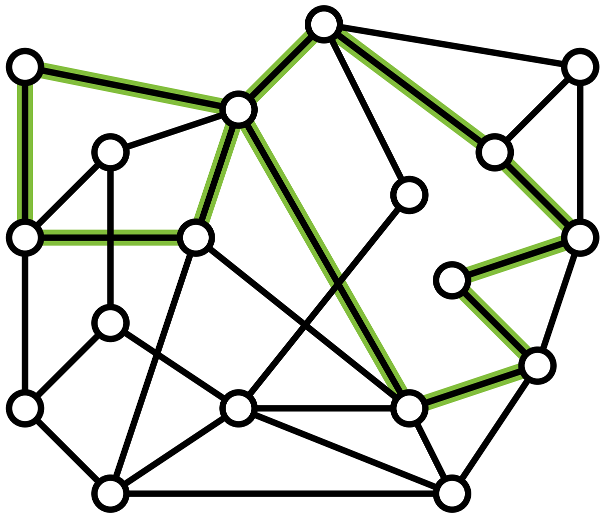


- Ein Graph kann sehr viele Kreise haben.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation

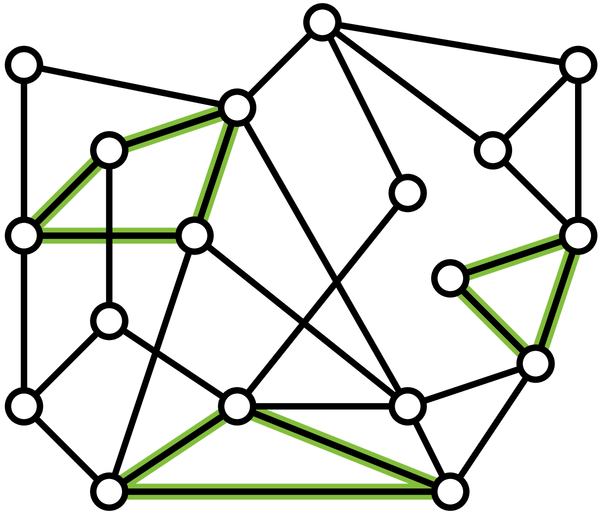
KIT
Karlsruher Institut für Technologie



- Ein Graph kann sehr viele Kreise haben.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

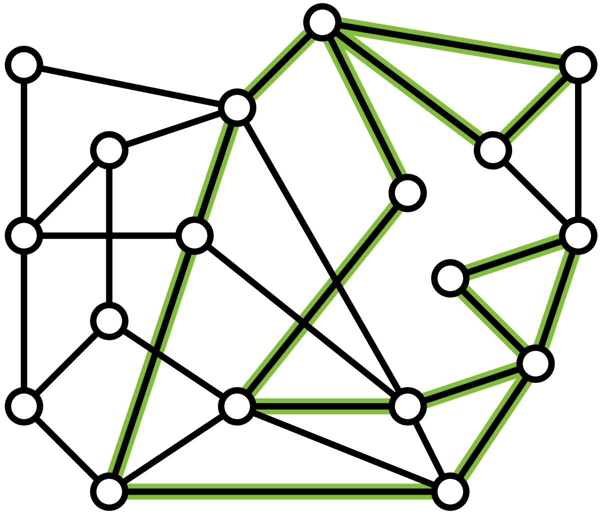
Motivation



- Ein Graph kann sehr viele Kreise haben.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

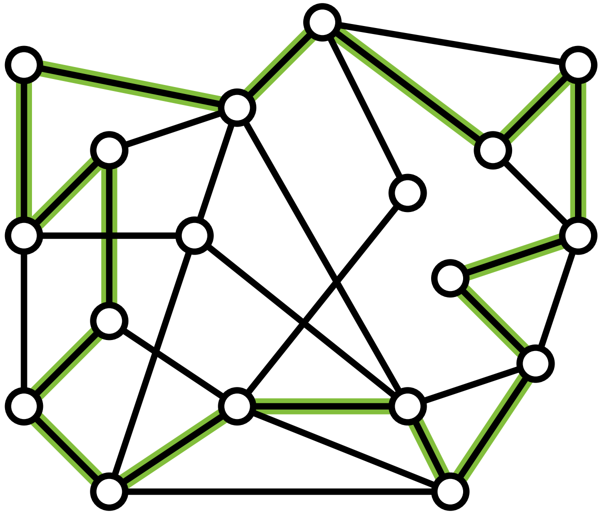
Motivation



- Ein Graph kann sehr viele Kreise haben.

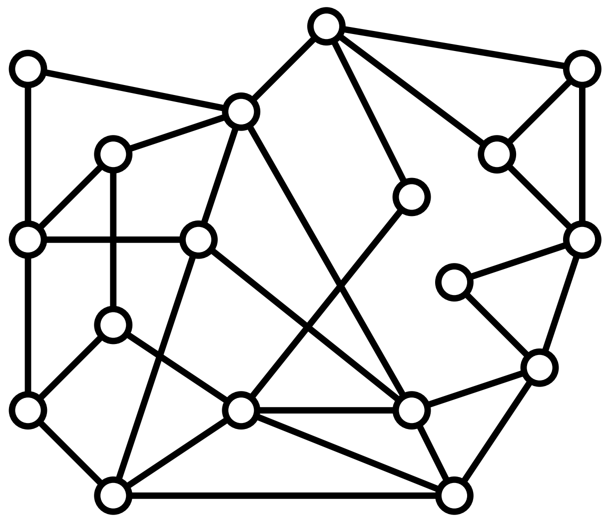
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation

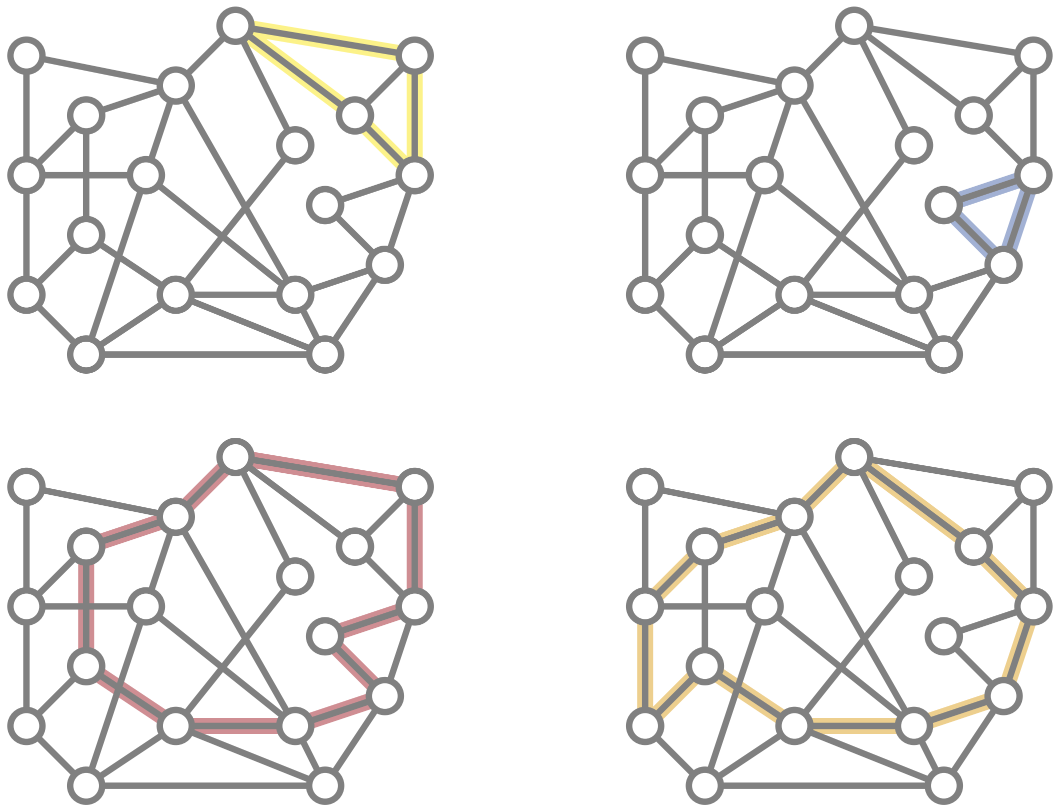


- Ein Graph kann sehr viele Kreise haben.

Motivation

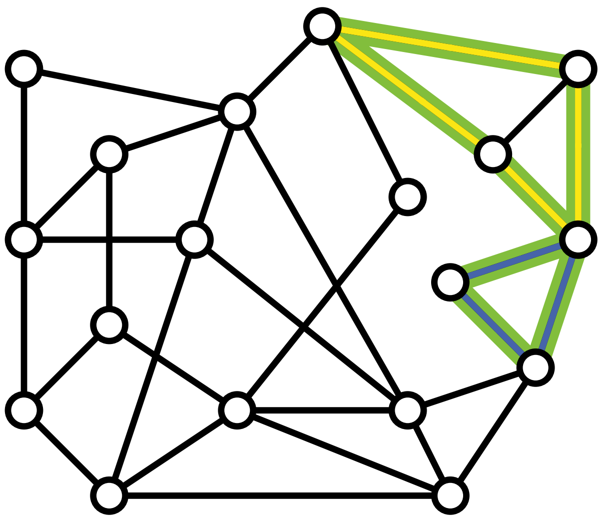


- Ein Graph kann sehr viele Kreise haben.

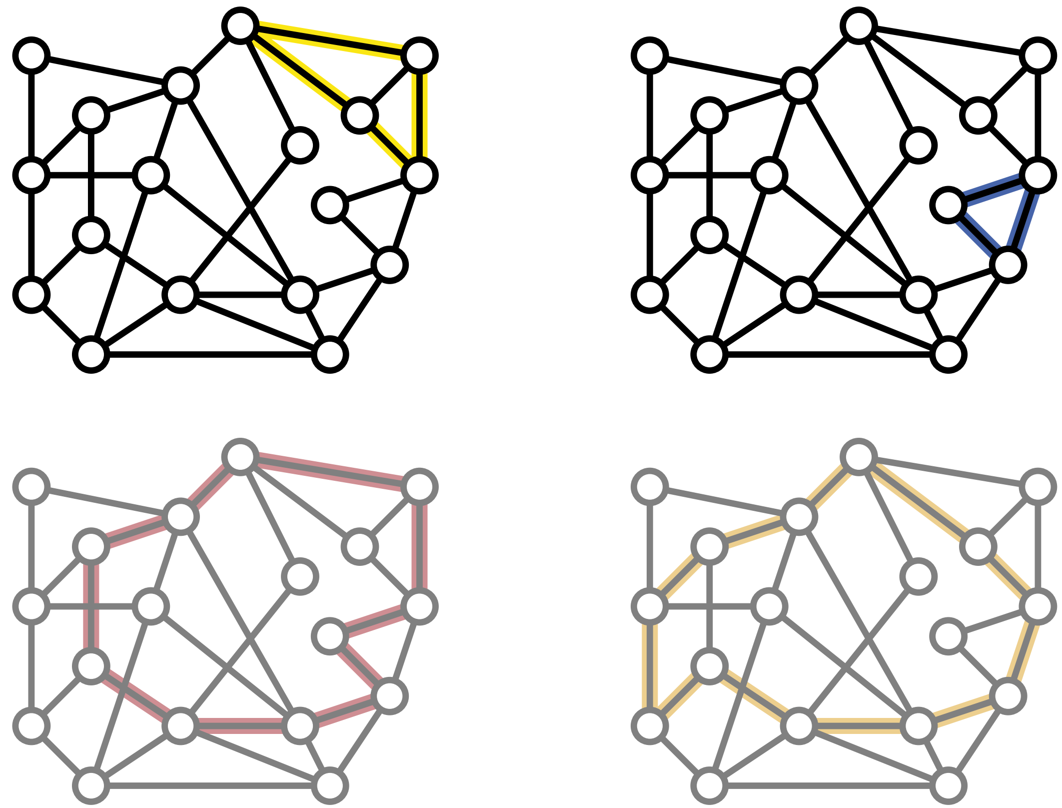


- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Motivation



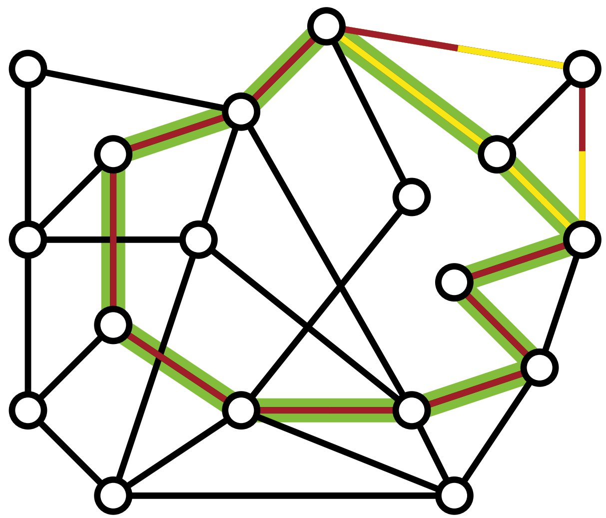
- Ein Graph kann sehr viele Kreise haben.



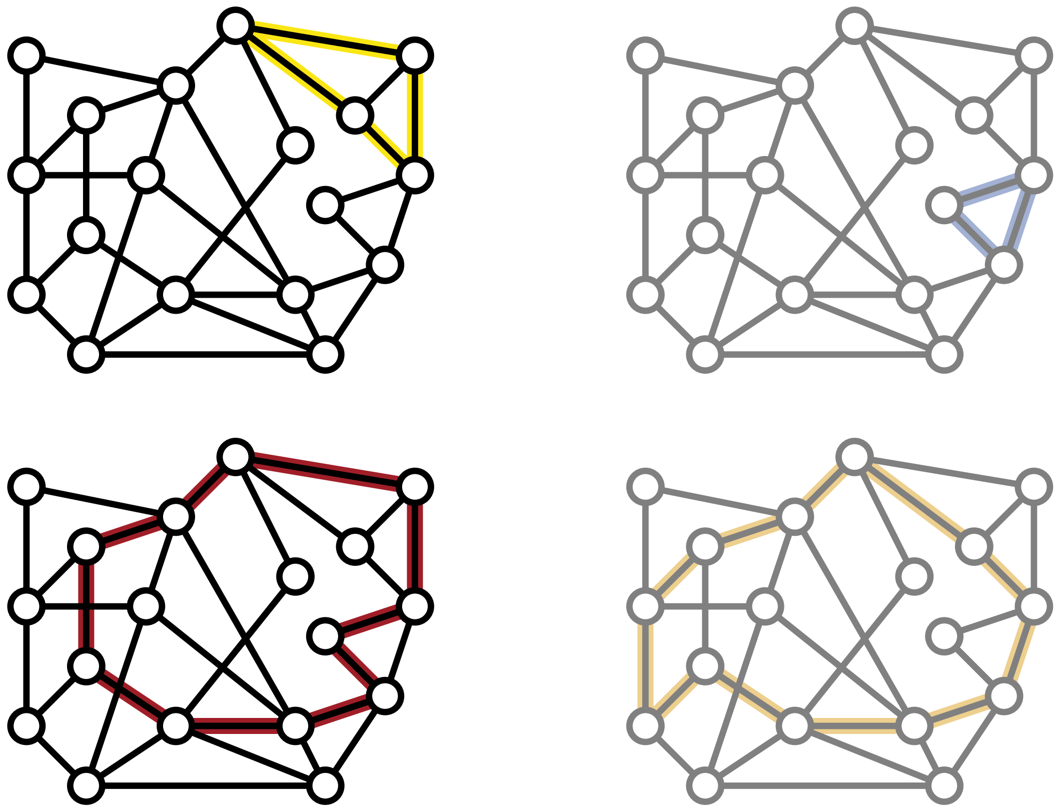
- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation



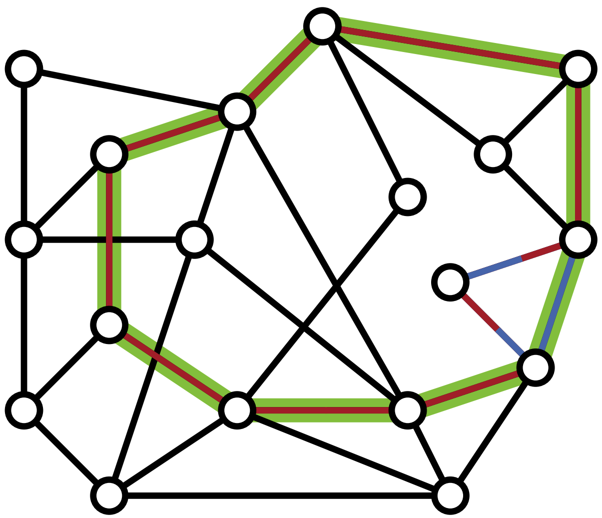
- Ein Graph kann sehr viele Kreise haben.



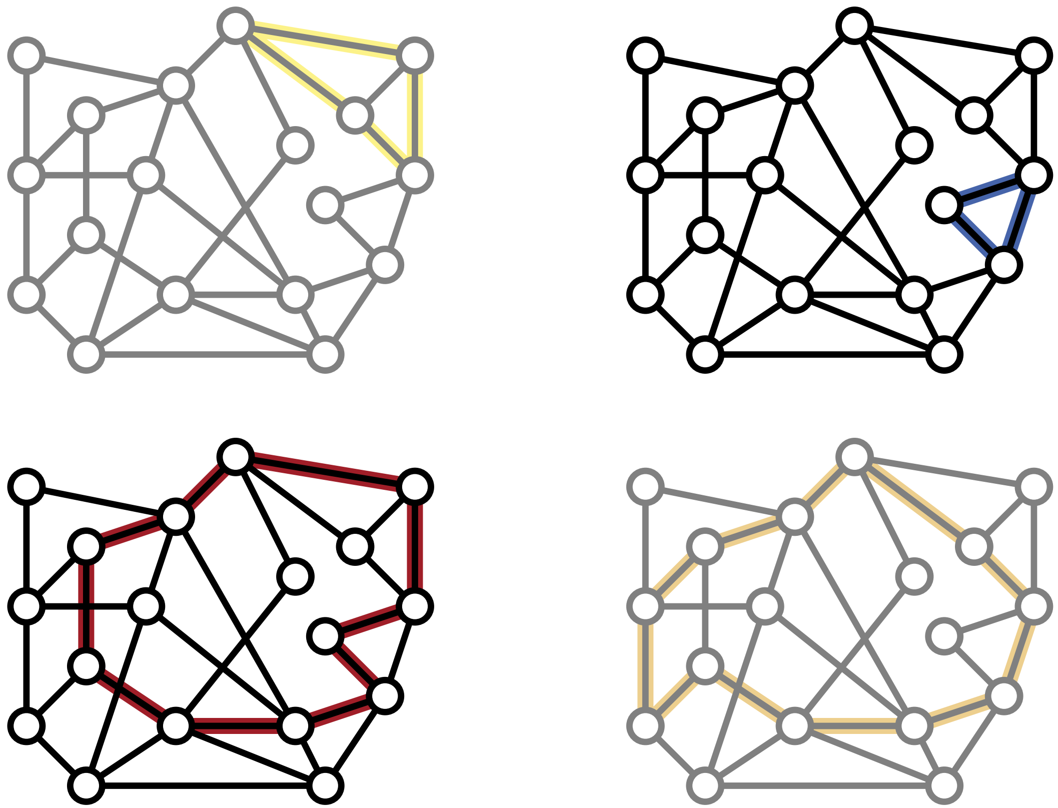
- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation

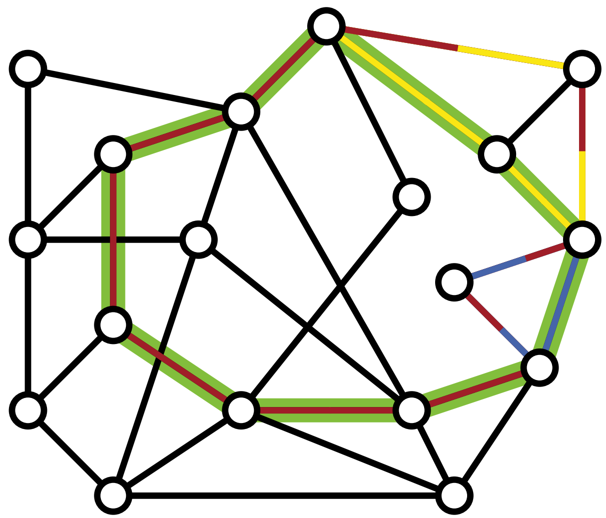


- Ein Graph kann sehr viele Kreise haben.

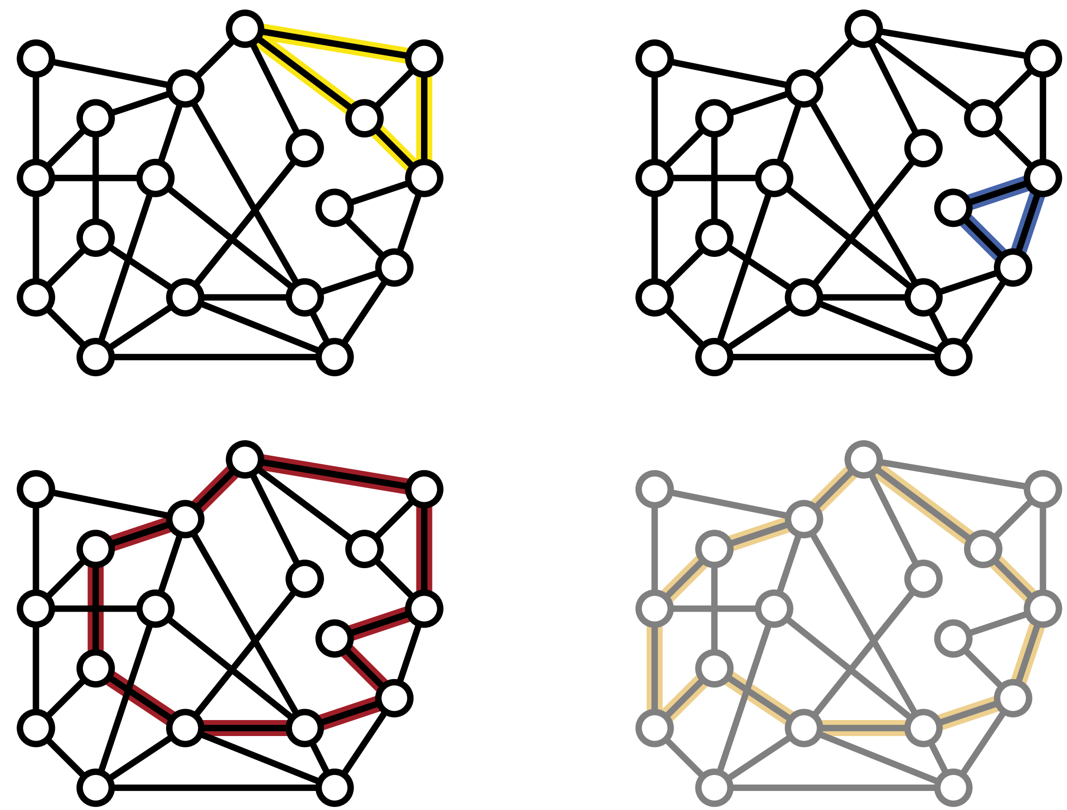


- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Motivation

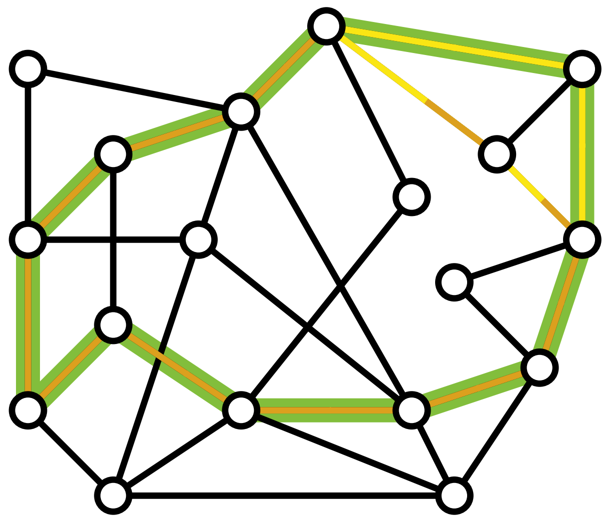


- Ein Graph kann sehr viele Kreise haben.

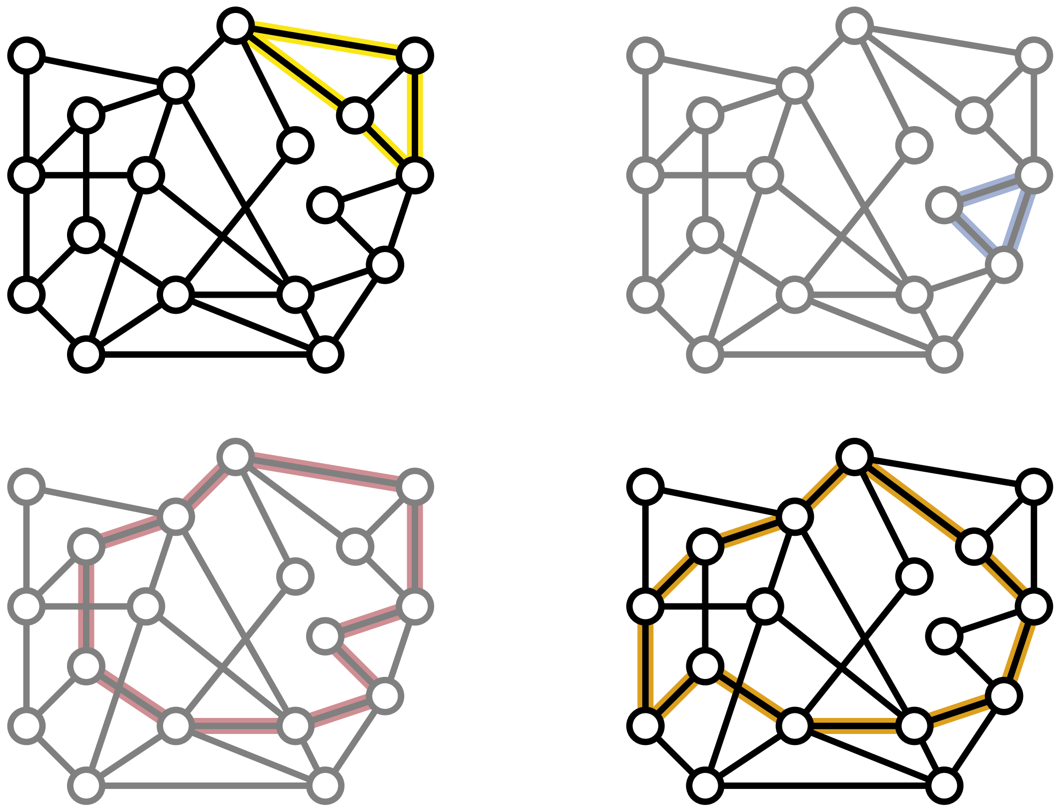


- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Motivation

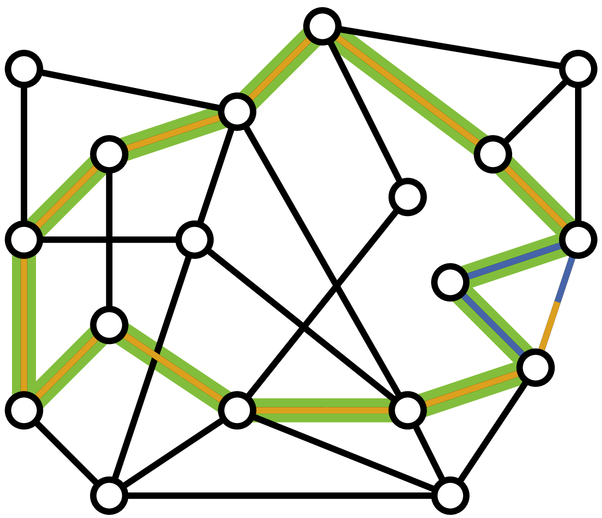


- Ein Graph kann sehr viele Kreise haben.

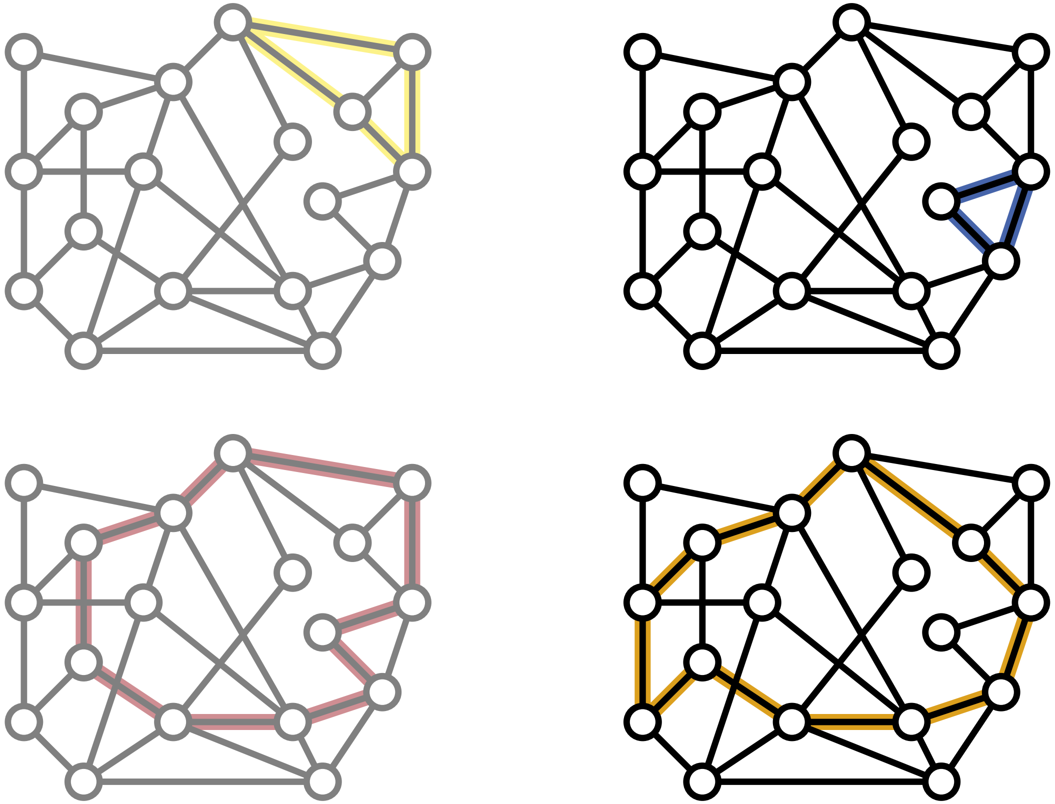


- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Motivation



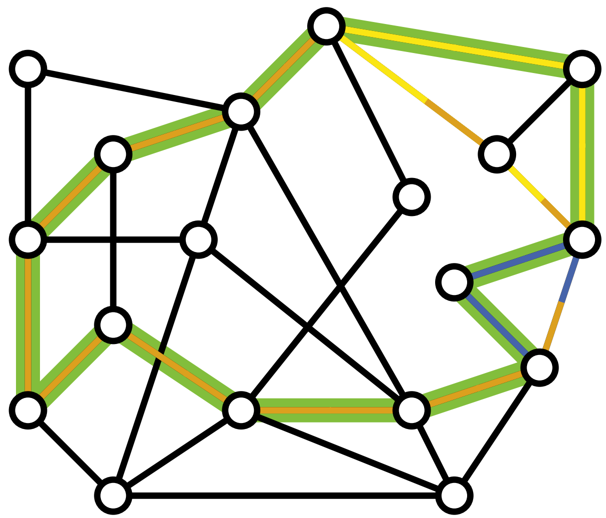
- Ein Graph kann sehr viele Kreise haben.



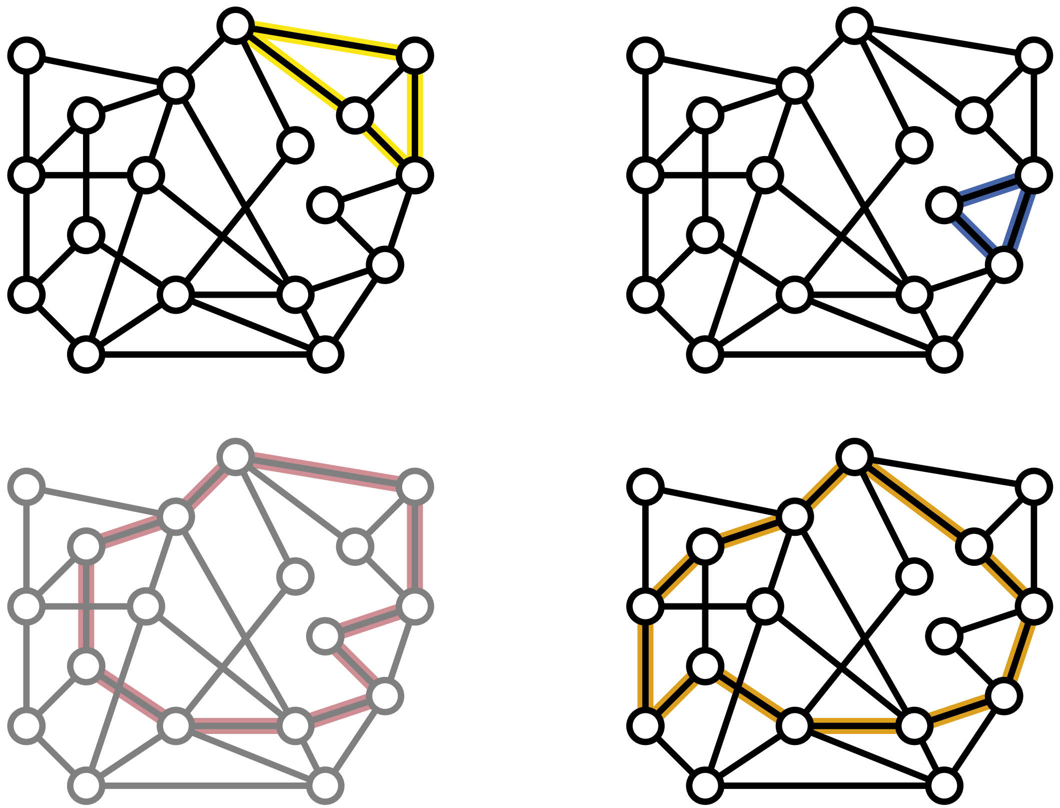
- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation



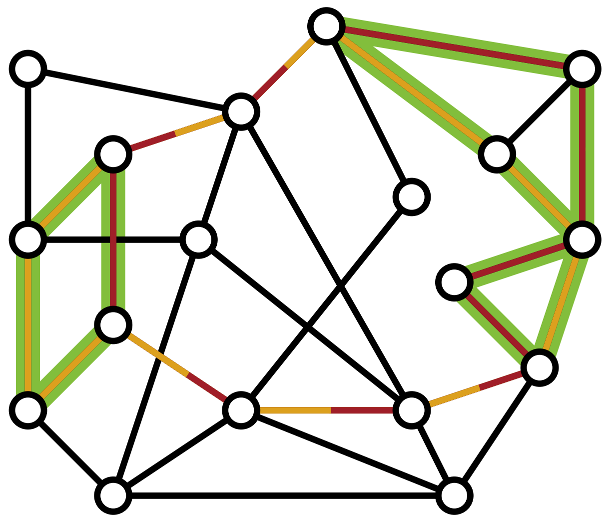
- Ein Graph kann sehr viele Kreise haben.



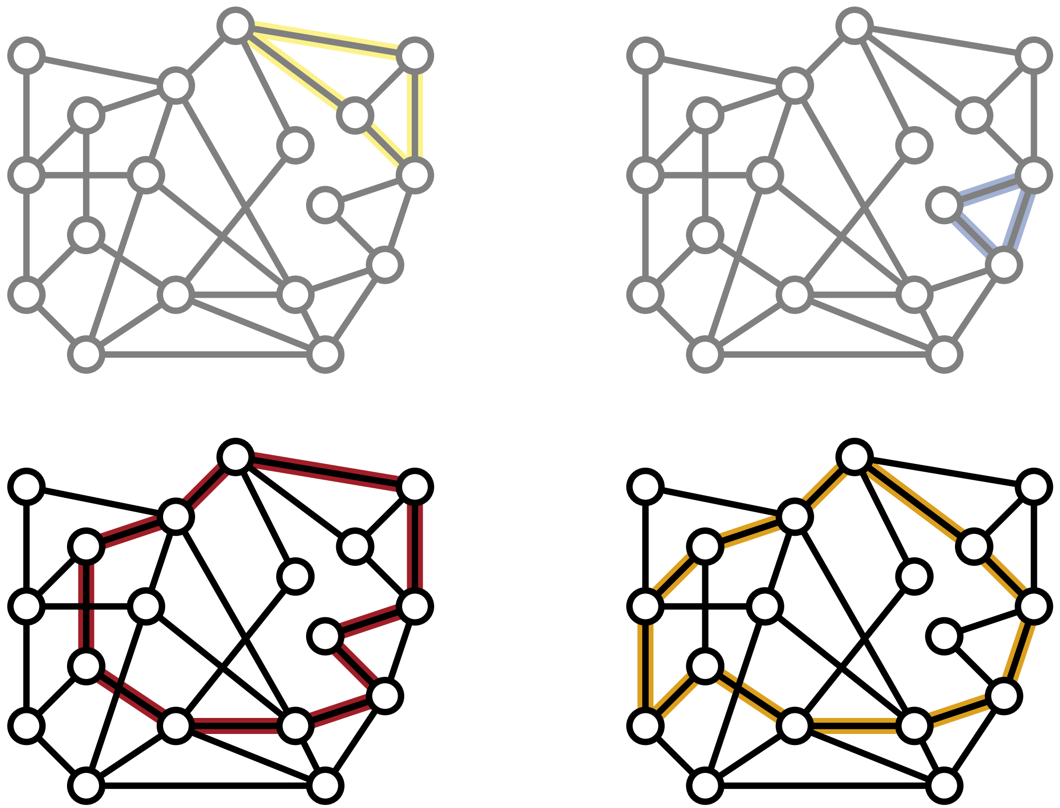
- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation

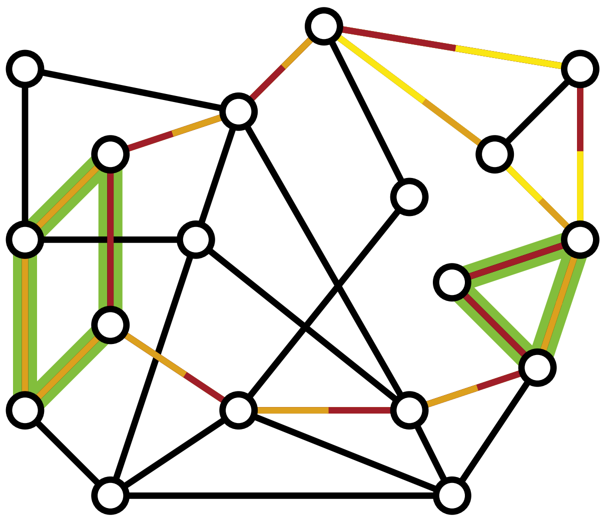


- Ein Graph kann sehr viele Kreise haben.

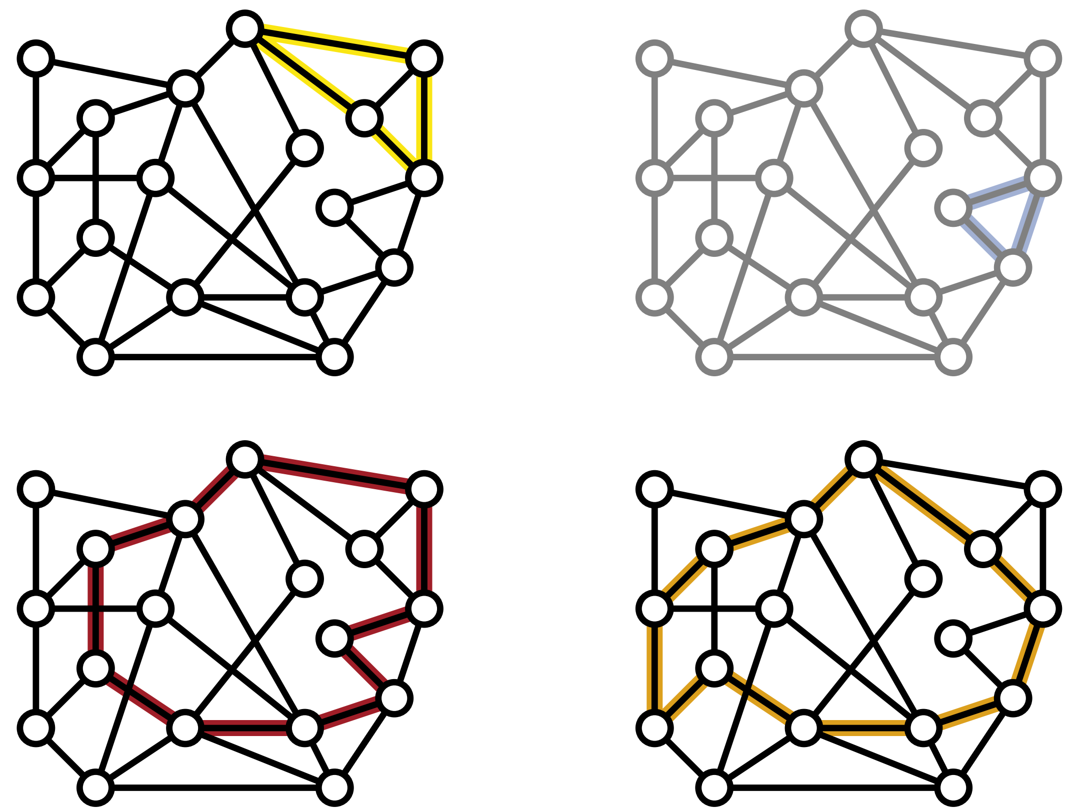


- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Motivation

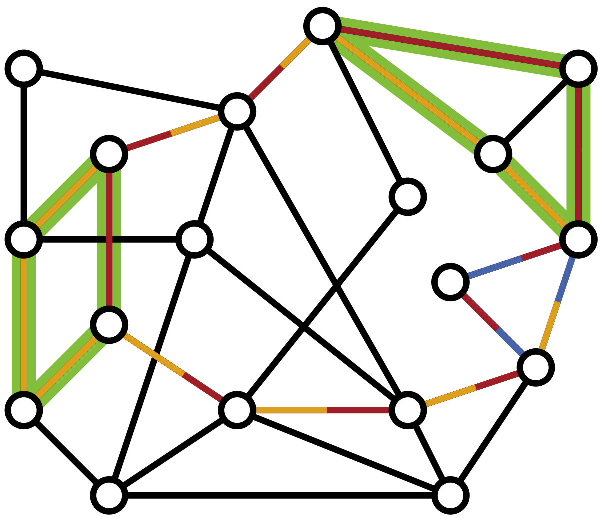


- Ein Graph kann sehr viele Kreise haben.



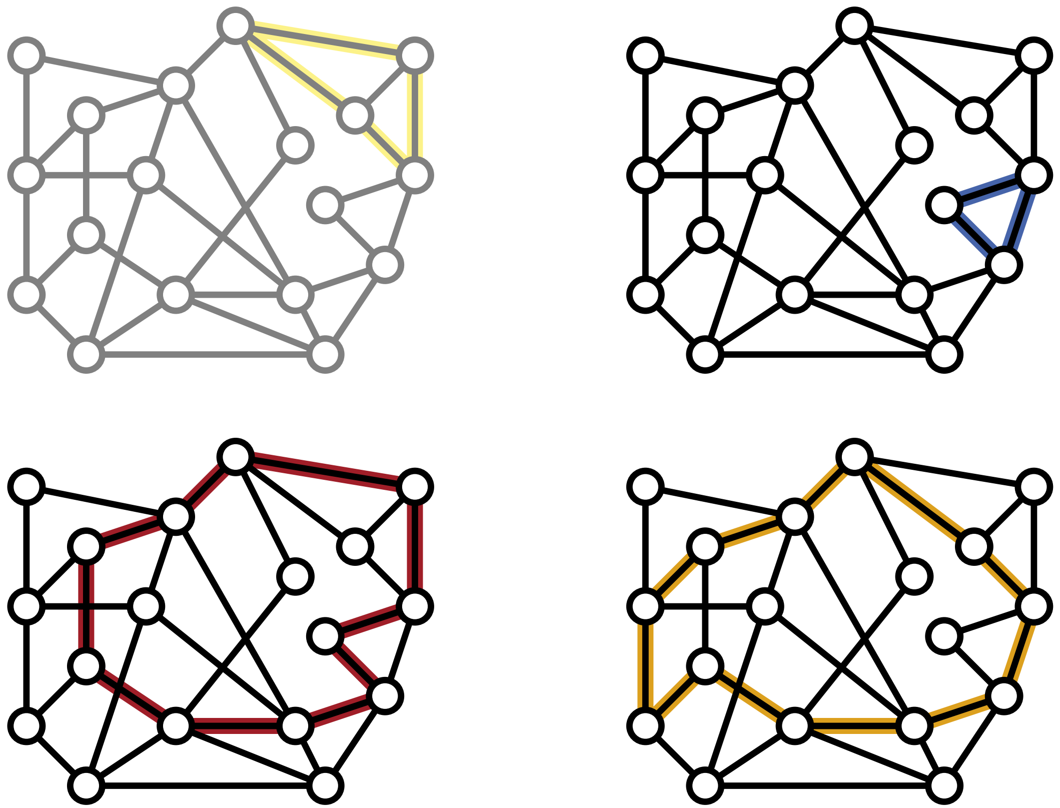
- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

Motivation



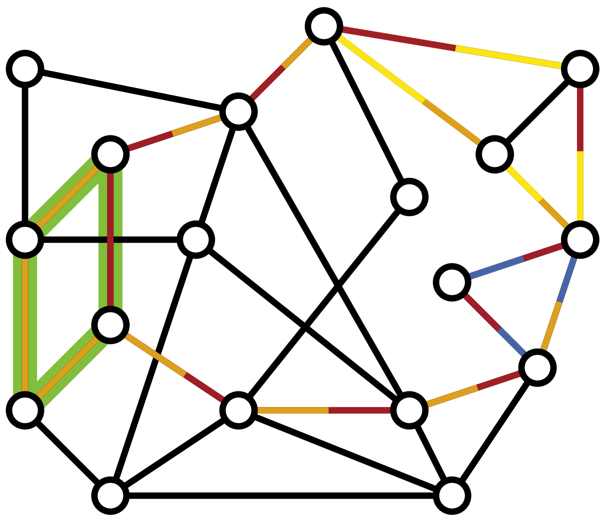
- Ein Graph kann sehr viele Kreise haben.

- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

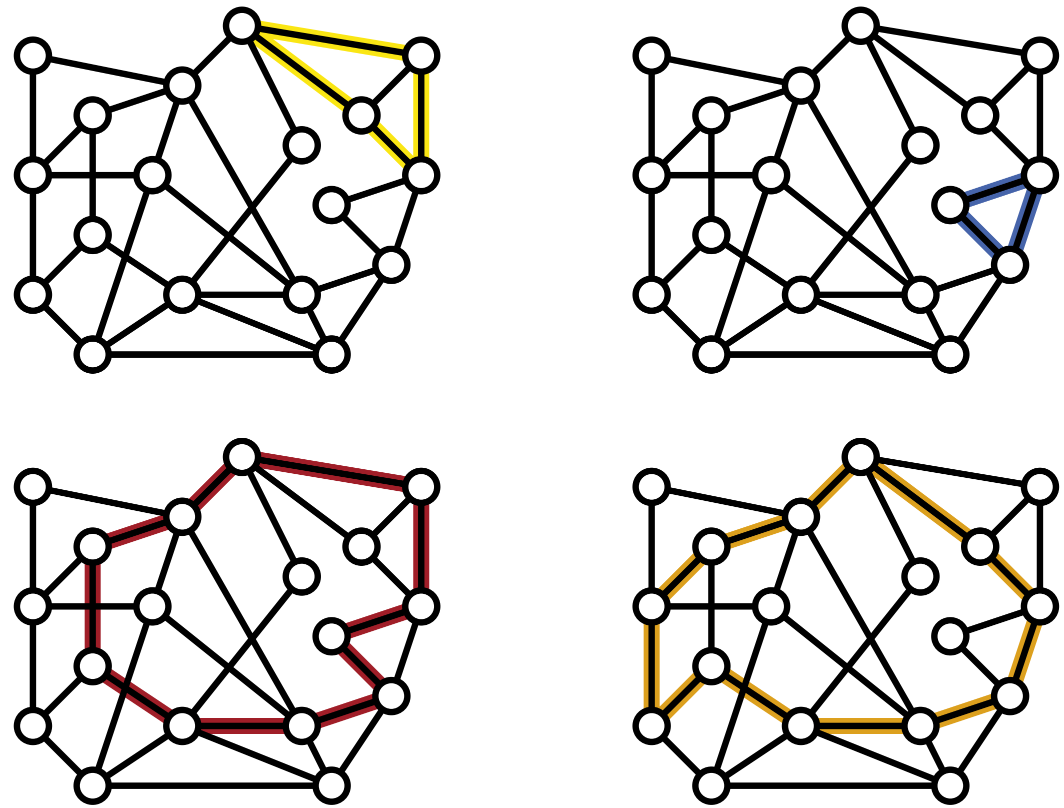


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Motivation



- Ein Graph kann sehr viele Kreise haben.

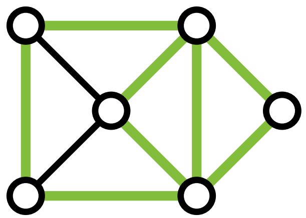


- Man kann aus wenigen Kreisen viele zusammensetzen.
- Wie viele braucht man, um alle Kreise zu erzeugen?

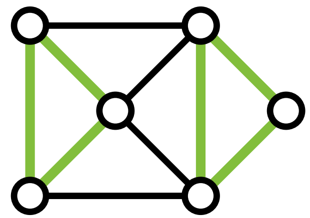
Kreise in Graphen

Definition: Kreis **(Definition 5.1)**

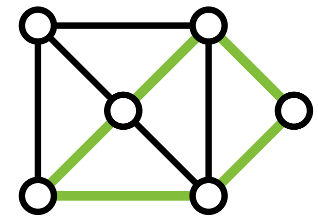
Ein Teilgraph $C = (V_C, E_C)$ von $G = (V, E)$ (d.h. $V_c \subseteq V$, $E_c \subseteq E$) heißt *Kreis* in G, falls alle Knoten aus V_C in C geraden Grad haben. Falls C zusammenhängend ist und alle Knoten aus V_C Grad zwei haben, so heißt C *einfacher Kreis*.



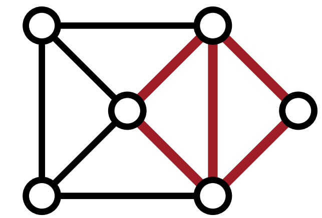
Kreis



Kreis



einfacher Kreis

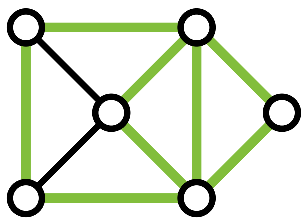


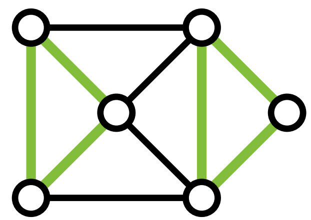
kein Kreis

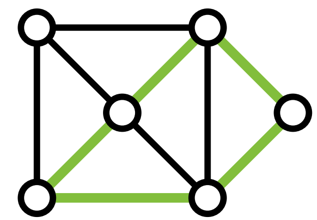
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

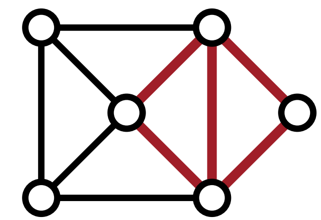
Kreise in Graphen

Definition: Kreis **(Definition 5.1)**
Ein Teilgraph $C = (V_C, E_C)$ von $G = (V, E)$ (d.h. $V_c \subseteq V$, $E_c \subseteq E$) heißt *Kreis* in G, falls alle Knoten aus V_C in C geraden Grad haben. Falls C zusammenhängend ist und alle Knoten aus V_C Grad zwei haben, so heißt C *einfacher Kreis*.


Kreis

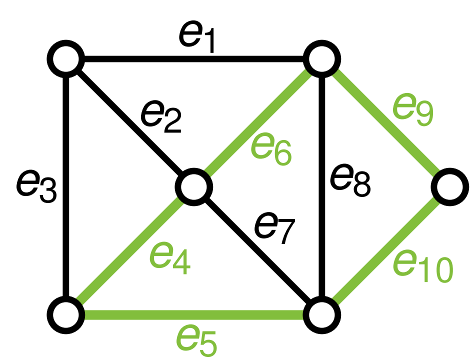

Kreis


einfacher Kreis


kein Kreis

Fasse Kreis als Kantenmenge $E' \subseteq E = \{e_1, \ldots, e_m\}$ auf und kodiere E' als Vektor $X^{E'}$ mit

$$X_i^{E'} := \begin{cases} 1, & \text{falls } e_i \in E' \\ 0, & \text{sonst} \end{cases}$$



$$\longrightarrow \quad X^{E'} = \begin{pmatrix} 0\\0\\0\\1\\1\\1\\0\\0\\1\\1 \end{pmatrix} \quad \begin{array}{l} e_1\\e_2\\e_3\\e_4\\e_5\\e_6\\e_7\\e_8\\e_9\\e_{10} \end{array}$$

Kreisraum

Definition: Kreisraum
Sei $\mathcal{C}$ die Menge aller Kreise in $G = (V, E)$. Dann induziert $\mathcal{C}$ den Vektorraum der Vektoren X^c, $c \in \mathcal{C}$ über dem Körper GF(2), genannt *Kreisraum* von G.

Erinnerung: $GF(2)$ ist der Körper mit zwei Elementen $\{0, 1\}$ und den Verknüpfungen + und · mit

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Kreisraum

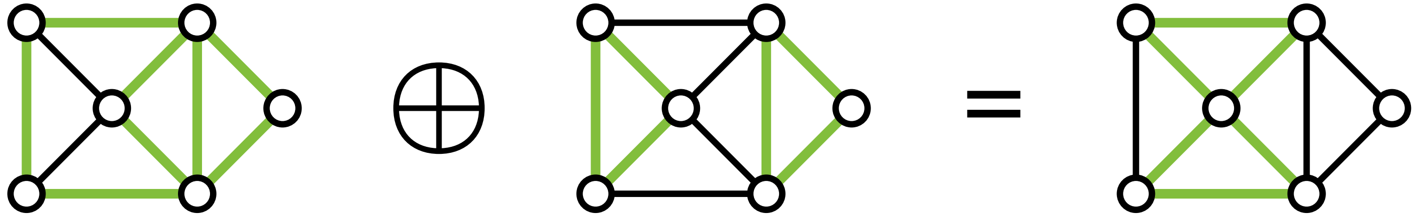
Definition: Kreisraum
Sei $\mathcal{C}$ die Menge aller Kreise in $G = (V, E)$. Dann induziert $\mathcal{C}$ den Vektorraum der Vektoren X^c, $c \in \mathcal{C}$ über dem Körper GF(2), genannt *Kreisraum* von G.

Erinnerung: $GF(2)$ ist der Körper mit zwei Elementen $\{0, 1\}$ und den Verknüpfungen $+$ und $\cdot$ mit

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Definition: Summe von Kreisen – symmetrische Differenz
Die Addition im Kreisraum von G induziert eine Operation $\oplus$ auf C durch $c_1 \oplus c_2 = (E_{c_1} \cup E_{c_2}) \setminus (E_{c_1} \cap E_{c_2})$. Dies ist die *symmetrische Differenz* beider Kantenmengen.



Ist wieder ein Kreis!

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Bemerkungen

- Die Begriffe *Dimension des Kreisraums*, *linear unabhängige* bzw. *abhängige* Menge von Kreisen sowie der Begriff der *Kreisbasis* ergeben sich in kanonischer Weise.

Bemerkungen

- Die Begriffe *Dimension des Kreisraums*, *linear unabhängige* bzw. *abhängige* Menge von Kreisen sowie der Begriff der *Kreisbasis* ergeben sich in kanonischer Weise.

- Man kann eine Kreisbasis wie folgt erhalten:

  (i) Betrachte aufspannenden Baum T von G (bzw. aufspannenden Wald, falls G unzusammenhängend ist).

  (ii) Für jede Nichtbaumkante $e_i = \{u, v\} \in E$ sei $C_i = P(u, v) \cup \{\{u, v\}\}$ der *Fundamentalkreis* zu e_i, wobei $P(u, v)$ der einfache Weg von u zu v in T ist.

  (iii) Die Menge aller Fundamentalkreise heißt *Fundamentalbasis* zu T und ist eine Kreisbasis.

Bemerkungen

- Die Begriffe *Dimension des Kreisraums*, *linear unabhängige* bzw. *abhängige* Menge von Kreisen sowie der Begriff der *Kreisbasis* ergeben sich in kanonischer Weise.

- Man kann eine Kreisbasis wie folgt erhalten:

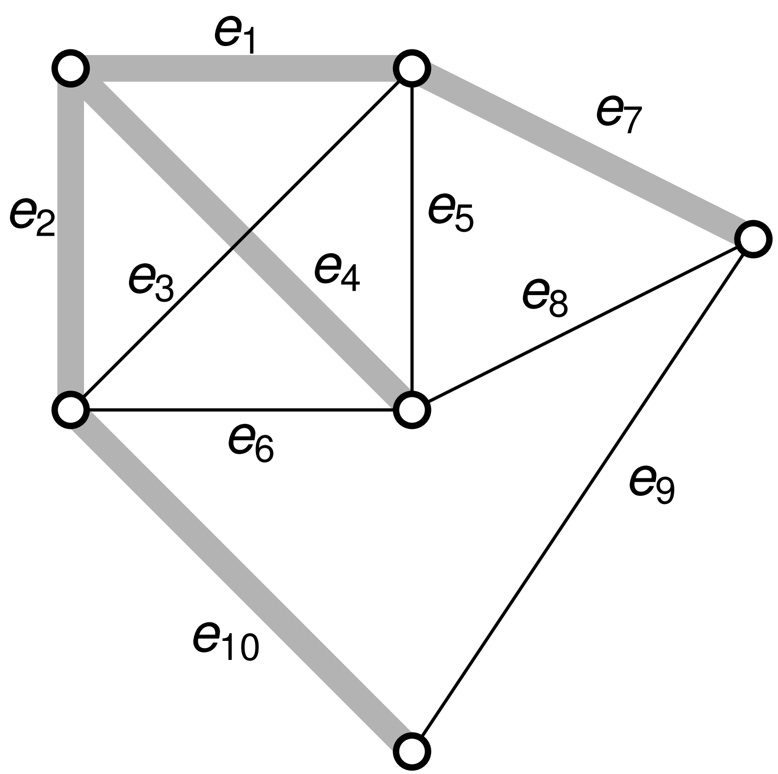
  (i) Betrachte aufspannenden Baum T von G (bzw. aufspannenden Wald, falls G unzusammenhängend ist).

  (ii) Für jede Nichtbaumkante $e_i = \{u, v\} \in E$ sei $C_i = P(u, v) \cup \{\{u, v\}\}$ der *Fundamentalkreis* zu e_i, wobei $P(u, v)$ der einfache Weg von u zu v in T ist.

  (iii) Die Menge aller Fundamentalkreise heißt *Fundamentalbasis* zu T und ist eine Kreisbasis.

- Die Dimension des Kreisraums von $G = (V, E)$ ist $m - n + \mathcal{K}(G)$, wobei $n = |V|$, $m = |E|$ und $\mathcal{K}(G)$ die Anzahl der Zusammenhangskomponenten in G ist.

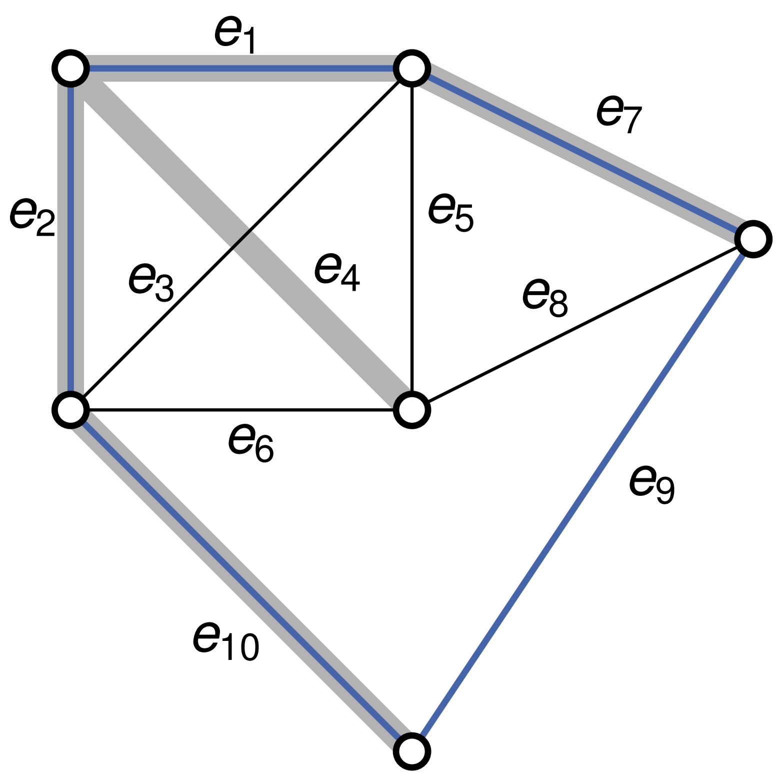
Beispiel



Spannbaum T:

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel

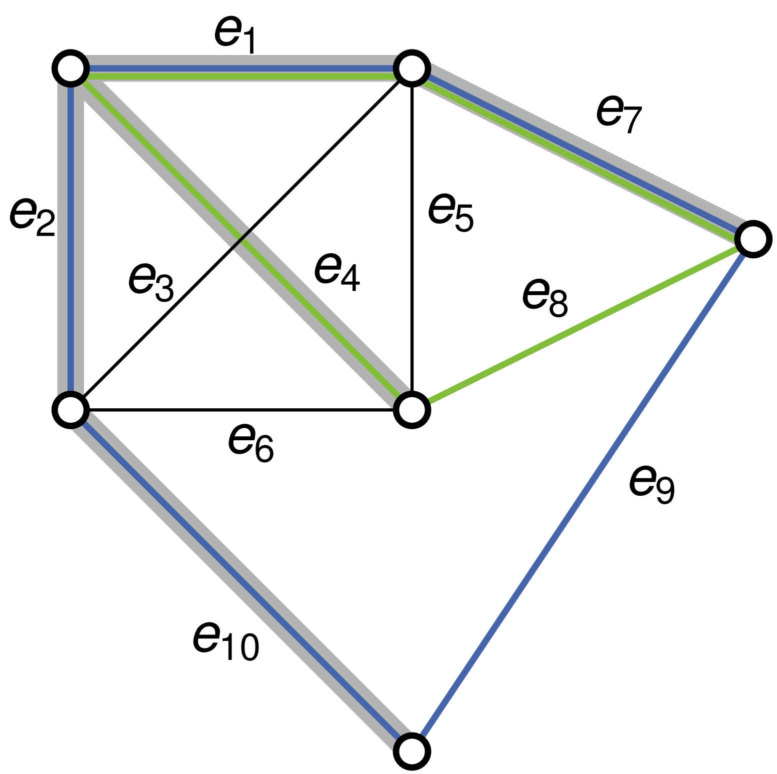


Spannbaum T:

Kante e_9 induziert Kreis:

$e_1 - e_7 - e_9 - e_{10} - e_2$

Beispiel



Spannbaum T:

Kante e_9 induziert Kreis:

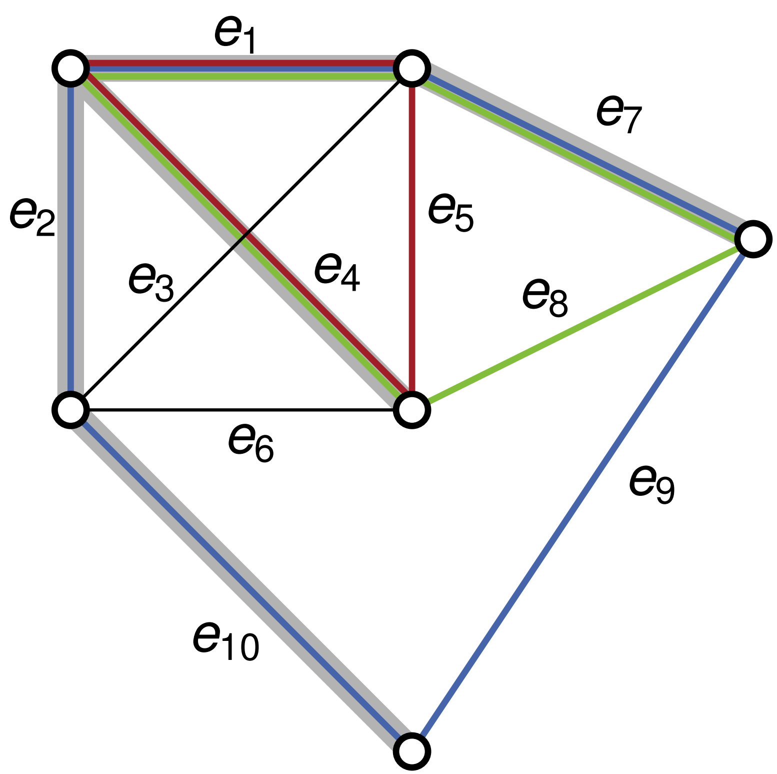
e_1 — e_7 — e_9 — e_{10} — e_2

Kante e_8 induziert Kreis:

e_1 — e_7 — e_8 — e_4

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel



Spannbaum T:

Kante e_9 induziert Kreis:

e_1 — e_7 — e_9 — e_{10} — e_2

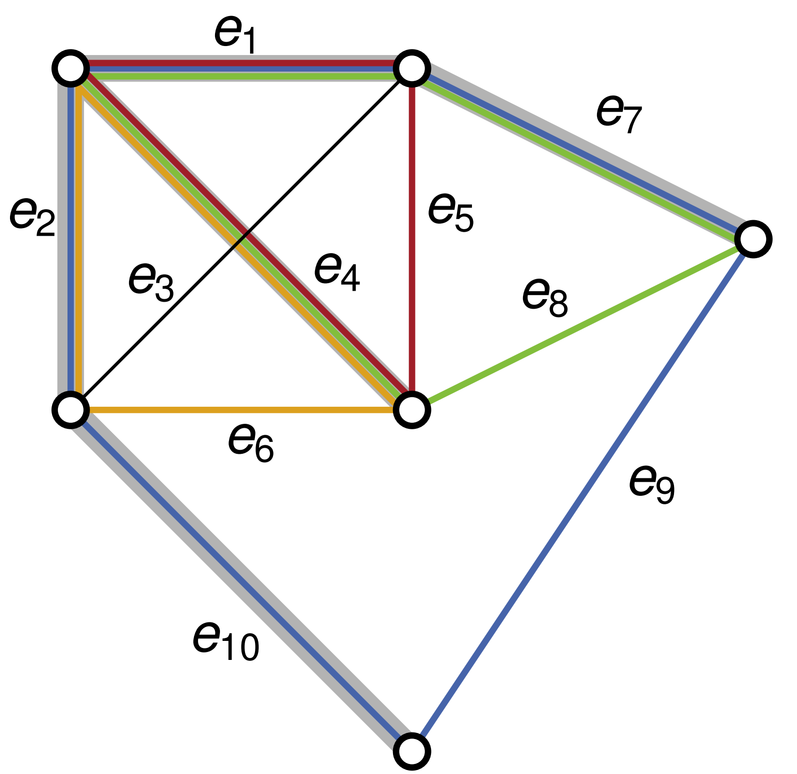
Kante e_8 induziert Kreis:

e_1 — e_7 — e_8 — e_4

Kante e_5 induziert Kreis:

e_1 — e_5 — e_4

Beispiel



Spannbaum T:

Kante e_9 induziert Kreis:

e_1 — e_7 — e_9 — e_{10} — e_2

Kante e_8 induziert Kreis:

e_1 — e_7 — e_8 — e_4

Kante e_5 induziert Kreis:

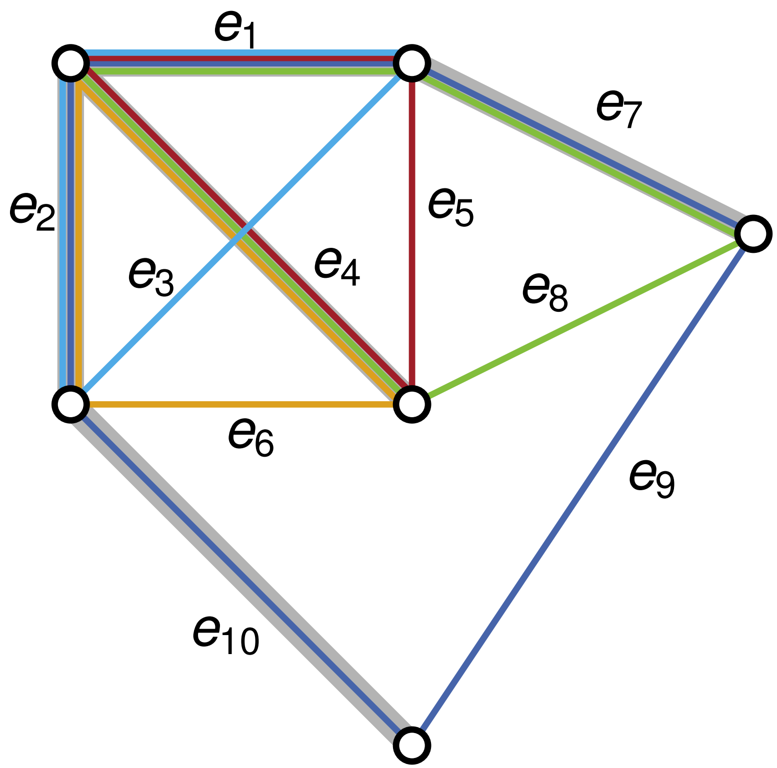
e_1 — e_5 — e_4

Kante e_6 induziert Kreis:

e_2 — e_4 — e_6

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel



Spannbaum T:

Kante e_9 induziert Kreis:

$e_1 - e_7 - e_9 - e_{10} - e_2$

Kante e_8 induziert Kreis:

$e_1 - e_7 - e_8 - e_4$

Kante e_5 induziert Kreis:

$e_1 - e_5 - e_4$

Kante e_6 induziert Kreis:

$e_2 - e_4 - e_6$

Kante e_3 induziert Kreis:

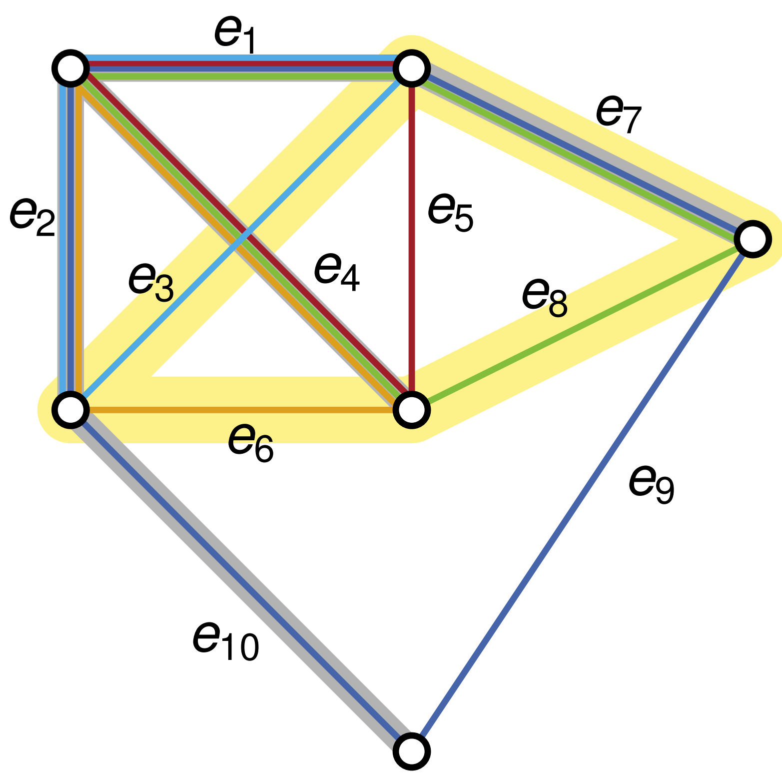
$e_1 - e_3 - e_2$

Kreisbasis

Dimension des Kreisraums: $m - n + 1 = 5$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel



Spannbaum T:

Kante e_9 induziert Kreis:
$e_1 - e_7 - e_9 - e_{10} - e_2$

Kante e_8 induziert Kreis:
$e_1 - e_7 - e_8 - e_4$

Kante e_5 induziert Kreis:
$e_1 - e_5 - e_4$

Kante e_6 induziert Kreis:
$e_2 - e_4 - e_6$

Kante e_3 induziert Kreis:
$e_1 - e_3 - e_2$

Kreisbasis

Dimension des Kreisraums: $m - n + 1 = 5$

Darstellung anderer Kreise bezüglich der gewählten Basis:

$$e_3 - e_7 - e_8 - e_6 = e_1 - e_7 - e_8 - e_4 \oplus e_2 - e_4 - e_6 \oplus e_1 - e_3 - e_2$$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem – MINIMUM CYCLE BASIS

Definition: Gewicht einer Kreisbasis
Sei zu $G = (V, E)$ die Kantengewichtsfunktion $w\colon E \longrightarrow \mathbb{R}_0^+$ gegeben. Das *Gewicht einer Kreisbasis* $\mathcal{B}$ von G ist definiert als

$$w(\mathcal{B}) = \sum_{C \in \mathcal{B}} w(C) = \sum_{C \in \mathcal{B}} \sum_{e \in C} w(e)$$

Problem: MCB
Gegeben sei ein Graph $G = (V, E)$ und eine Gewichtsfunktion $w\colon E \longrightarrow \mathbb{R}_0^+$. Finde eine Kreisbasis $\mathcal{B}$ von G mit minimalem Gewicht.

Algorithmus zur Bestimmung einer MCB: $\rightarrow$ nächste Vorlesung

Matroide & Greedy Algorithmen

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unabhängigkeitssysteme

Definition: Unabhängigkeitssystem **(Definition 2.8)**

Ein Tupel $(M, \mathcal{U})$, wobei $\mathcal{U} \subset 2^M$ ein Mengensystem über einer endlichen Menge M ist, heißt *Unabhängigkeitssystem*, wenn

- $\emptyset \in \mathcal{U}$ und
- $I_1 \in \mathcal{U}, I_2 \subseteq I_1 \Rightarrow I_2 \in \mathcal{U}$.

Die Mengen $I \subseteq M$ mit $I \in \mathcal{U}$ werden *unabhängig*, alle anderen Mengen $I \subseteq M$ *abhängig* genannt.

Bemerkung: Für einen Vektorraum V und die Menge $\mathcal{U}$ aller linear unabhängigen Teilmengen von V ist $(V, \mathcal{U})$ ein Unabhängigkeitssystem. (gilt auch für den Kreisraum)

Unabhängigkeitssysteme

Definition: Unabhängigkeitssystem **(Definition 2.8)**
Ein Tupel $(M, \mathcal{U})$, wobei $\mathcal{U} \subset 2^M$ ein Mengensystem über einer endlichen Menge M ist, heißt *Unabhängigkeitssystem*, wenn

- $\emptyset \in \mathcal{U}$ und
- $I_1 \in \mathcal{U}, I_2 \subseteq I_1 \Rightarrow I_2 \in \mathcal{U}$.

Die Mengen $I \subseteq M$ mit $I \in \mathcal{U}$ werden *unabhängig*, alle anderen Mengen $I \subseteq M$ *abhängig* genannt.

Bemerkung: Für einen Vektorraum V und die Menge $\mathcal{U}$ aller linear unabhängigen Teilmengen von V ist $(V, \mathcal{U})$ ein Unabhängigkeitssystem. (gilt auch für den Kreisraum)

Definition: Basis, Basissystem & Rang **(Definition 2.9)**
Sei $(M, \mathcal{U})$ ein Unabhängigkeitssystem. Für $F \subseteq M$ ist jede unabhängige Menge $U \in \mathcal{U}$, $U \subseteq F$ die bezüglich $\subseteq$ maximal ist eine *Basis* von F. Eine Basis von M wird auch *Basis des Unabhängigkeitssystems* genannt. Die Menge aller Basen von $(M, \mathcal{U})$ heißt *Basissystem von* $(M, \mathcal{U})$.
Für $F \subseteq M$ heißt $r(F) := max\{|B| : B \text{ ist Basis von } F\}$ der *Rang von* F. Der Rang von M wird auch *Rang des Unabhängigkeitssystems* genannt.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Optimierungsprobleme

Karlsruher Institut für Technologie

Sei $(M, \mathcal{U})$ ein Unabhängigkeitssystem mit Basissystem $\mathcal{B}$ und Gewichtsfunktion $w\colon M \longrightarrow \mathbb{R}$.

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist. **(Definition 2.11)**

Problem: Optimierungsproblem über dem Basissystem $\mathcal{B}$
Finde Basis $B^\star \in \mathcal{B}$ sodass $w(B^\star)$ minimal ist. **(Definition 2.11)**

Optimierungsprobleme

Sei $(M, \mathcal{U})$ ein Unabhängigkeitssystem mit Basissystem $\mathcal{B}$ und Gewichtsfunktion $w\colon M \longrightarrow \mathbb{R}$.

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist. **(Definition 2.11)**

Problem: Optimierungsproblem über dem Basissystem $\mathcal{B}$
Finde Basis $B^\star \in \mathcal{B}$ sodass $w(B^\star)$ minimal ist. **(Definition 2.11)**

Mögliche Lösungsstrategie für diese Probleme: Greedy

GREEDY-METHODE für Optimierungsproblem Π

Sortiere M aufsteigend (absteigend), falls Π Optimierungsproblem über Basissystem (Unabhängigkeitessystem) ist, sei $\ell_1, \ldots, \ell_{|M|}$ die Sortierung.

$I^\star \leftarrow \emptyset$

for $i = 1$ **to** $|M|$ **do**
  if $I^\star \cup \{\ell_i\} \in \mathcal{U}$ **then**
    $I^\star \leftarrow I^\star \cup \{\ell_i\}$

Optimierungsprobleme

Sei $(M, \mathcal{U})$ ein Unabhängigkeitssystem mit Basissystem $\mathcal{B}$ und Gewichtsfunktion $w\colon M \longrightarrow \mathbb{R}$.

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist. **(Definition 2.11)**

Problem: Optimierungsproblem über dem Basissystem $\mathcal{B}$
Finde Basis $B^\star \in \mathcal{B}$ sodass $w(B^\star)$ minimal ist. **(Definition 2.11)**

Mögliche Lösungsstrategie für diese Probleme: Greedy

GREEDY-METHODE für Optimierungsproblem Π

Sortiere M aufsteigend (absteigend), falls Π Optimierungsproblem über Basissystem (Unabhängigkeitessystem) ist, sei $\ell_1, \dots, \ell_{|M|}$ die Sortierung.

$I^\star \leftarrow \emptyset$

for $i = 1$ **to** $|M|$ **do**
  if $I^\star \cup \{\ell_i\} \in \mathcal{U}$ **then**
    $I^\star \leftarrow I^\star \cup \{\ell_i\}$

Wann liefert das eine optimale Lösung?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide

Karlsruher Institut für Technologie

Definition: Matroid **(Definition 2.13)**
Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

Äquivalent kann statt $|I| < |J|$ auch $|I| + 1 = |J|$ gefordert werden

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide

Definition: Matroid **(Definition 2.13)**
Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

Äquivalent kann statt $|I| < |J|$ auch $|I| + 1 = |J|$ gefordert werden

Beispiel 1: Jeder Vektorraum ist ein Matroid, denn:
Falls I und J linear unabhängig mit $|I| < |J|$, dann kann nicht jeder Vektor in J als Linearkombination von Vektoren aus I dargestellt werden.
$\Rightarrow$ Der Kreisraum eines Graphen G bildet einen Matroid, den *Kreismatroid* von G.

Matroide

Definition: Matroid **(Definition 2.13)**

Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

Äquivalent kann statt $|I| < |J|$ auch $|I| + 1 = |J|$ gefordert werden

Beispiel 1: Jeder Vektorraum ist ein Matroid, denn:
Falls I und J linear unabhängig mit $|I| < |J|$, dann kann nicht jeder Vektor in J als Linearkombination von Vektoren aus I dargestellt werden.
$\Rightarrow$ Der Kreisraum eines Graphen G bildet einen Matroid, den *Kreismatroid* von G.

Beispiel 2: Sei $G = (V, E)$ ein zusammenhängender Graph. Das Mengensystem $(E, \mathcal{U})$ mit $\mathcal{U} = \{E' \subseteq E : E' \text{ induziert einen Wald in } G\}$ ist ein Matroid.

Matroide

Definition: Matroid **(Definition 2.13)**
Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

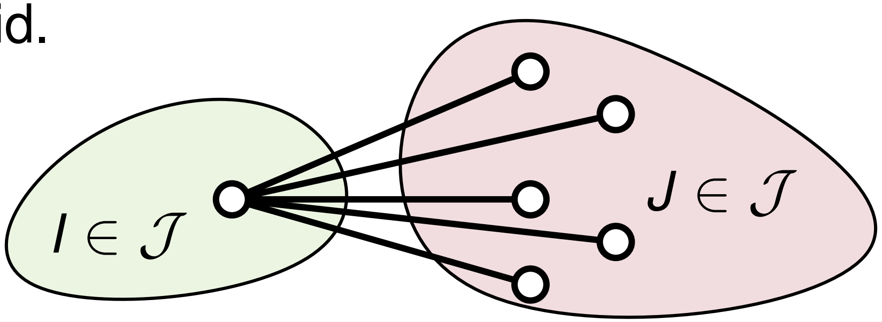
Äquivalent kann statt $|I| < |J|$ auch $|I| + 1 = |J|$ gefordert werden

Beispiel 1: Jeder Vektorraum ist ein Matroid, denn:
Falls I und J linear unabhängig mit $|I| < |J|$, dann kann nicht jeder Vektor in J als Linearkombination von Vektoren aus I dargestellt werden.
$\Rightarrow$ Der Kreisraum eines Graphen G bildet einen Matroid, den *Kreismatroid* von G.

Beispiel 2: Sei $G = (V, E)$ ein zusammenhängender Graph. Das Mengensystem $(E, \mathcal{U})$ mit $\mathcal{U} = \{E' \subseteq E : E' \text{ induziert einen Wald in } G\}$ ist ein Matroid.

Beispiel 3: Es sei $G = (V, E)$. Das Mengensystem $(V, \mathcal{J})$, wobei $\mathcal{J} = \{V' \subseteq V \mid V' \text{ unabhängige Knotenmenge in } G, \text{ d.h. } \forall u, v \in V' \text{ gilt } \{u, v\} \notin E\}$ ist ein Unabhängigkeitsystem, aber **kein** Matroid.

$|I| < |J|$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide und Greedy-Algorithmen

Definition: Matroid **(Definition 2.13)**

Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

Satz: Maximierungsprobleme auf Matroiden **(Satz 2.16)**

Für ein Unabhängigkeitssystem $(M, \mathcal{U})$ sind folgende Aussagen äquivalent:

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

(b) $(M, \mathcal{U})$ ist ein Matroid.

(c) Für eine beliebige Menge $F \subseteq M$ und beliebige inklusionsmaximale unabhängige Mengen $I_1, I_2 \subseteq F$ gilt: $|I_1| = |I_2|$.

Erinnerung:

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist.

Beweis: Wir zeigen: (a) $\Rightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (a)

Matroide und Greedy-Algorithmen

Beweis: Wir zeigen: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

(*a*) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

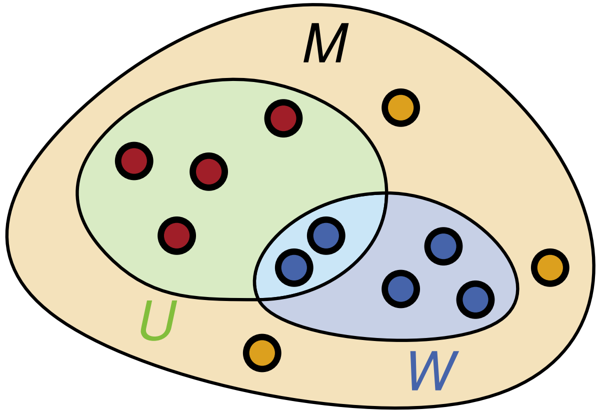
$\Rightarrow$

(*b*) $(M, \mathcal{U})$ ist ein Matroid.

Annahme: Greedy liefert Optimallösung, aber $(M, \mathcal{U})$ ist **kein** Matroid.

- Es gibt $U, W \in \mathcal{U}$ mit $|U| = |W| + 1$, sodass $W \cup \{e\} \notin \mathcal{U}$ für alle $e \in U \setminus W$.

Negation der Matroideigenschaft



Erinnerung:

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist.

Matroide und Greedy-Algorithmen

Beweis: Wir zeigen: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

(*a*) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

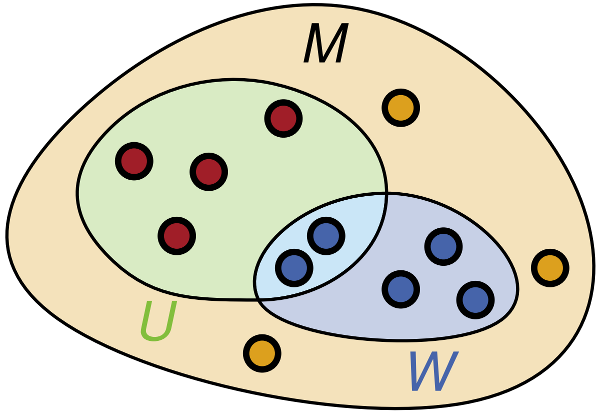
$\Rightarrow$

(*b*) $(M, \mathcal{U})$ ist ein Matroid.

Annahme: Greedy liefert Optimallösung, aber $(M, \mathcal{U})$ ist **kein** Matroid.

- Es gibt $U, W \in \mathcal{U}$ mit $|U| = |W| + 1$, sodass $W \cup \{e\} \notin \mathcal{U}$ für alle $e \in U \setminus W$.
- Setze Gewichte wie folgt:

$$w(e) = \begin{cases} |W| + 2 \text{ falls } e \in W \\ |W| + 1 \text{ falls } e \in U \setminus W \\ -1 \text{ sonst} \end{cases}$$



Erinnerung:

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist.

Matroide und Greedy-Algorithmen

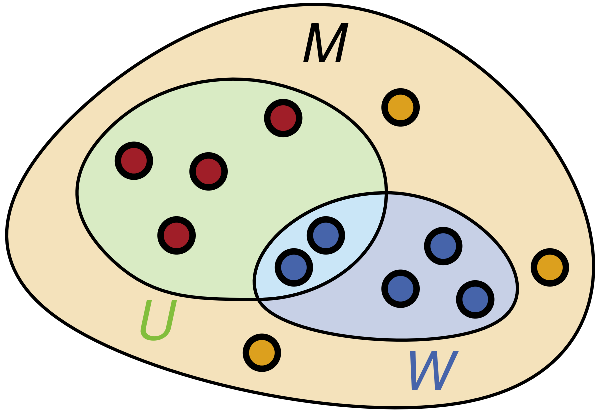
Beweis: Wir zeigen: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.	$\Rightarrow$	(b) $(M, \mathcal{U})$ ist ein Matroid.

Annahme: Greedy liefert Optimallösung, aber $(M, \mathcal{U})$ ist **kein** Matroid.

- Es gibt $U, W \in \mathcal{U}$ mit $|U| = |W| + 1$, sodass $W \cup \{e\} \notin \mathcal{U}$ für alle $e \in U \setminus W$.
- Setze Gewichte wie folgt:

$$w(e) = \begin{cases} |W| + 2 \text{ falls } e \in W \\ |W| + 1 \text{ falls } e \in U \setminus W \\ -1 \text{ sonst} \end{cases}$$



- Greedy-Methode findet: W mit $w(W) = |W| \cdot (|W| + 2) = |W|^2 + 2|W|$
- Besser wäre: U mit $w(U) \geq (|W| + 1) \cdot (|W| + 1) = |W|^2 + 2|W| + 1$

Erinnerung:

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide und Greedy-Algorithmen

Beweis: Wir zeigen: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

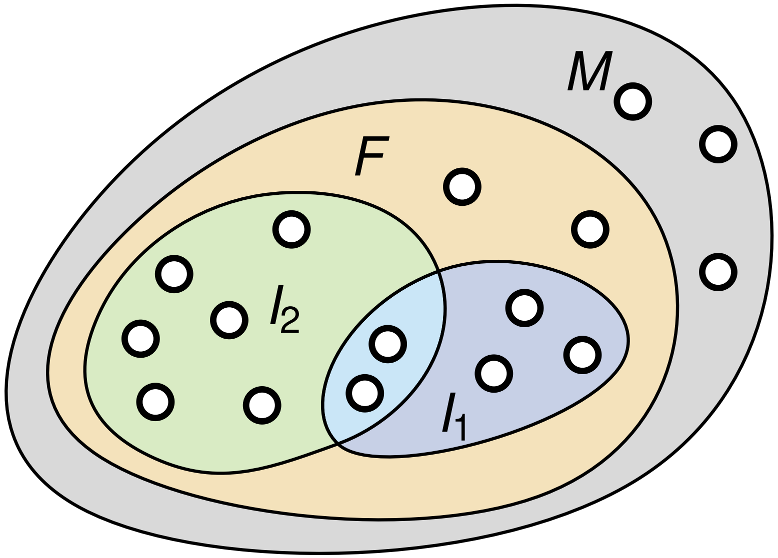
(b) $(M, \mathcal{U})$ ist ein Matroid. $\Rightarrow$ (c) Für eine beliebige Menge $F \subseteq M$ und beliebige inklusionsmaximale unabhängige Mengen $I_1, I_2 \subseteq F$ gilt: $|I_1| = |I_2|$.

Annahme: $(M, \mathcal{U})$ ist Matroid, aber $I_1, I_2 \subseteq F$ sind inklusionsmaximale unabhängige Menge $(I_1, I_2 \in \mathcal{U})$ mit $|I_1| < |I_2|$.

- Es gibt ein Element $e \in I_2 \setminus I_1$, sodass $I_1' = I_1 \cup \{e\} \in \mathcal{U}$.

Matroideigenschaft

- $e \in I_2 \subseteq F \Rightarrow I_1' \subseteq F$
- I_1 ist nicht inklusionsmaximal.



Matroide und Greedy-Algorithmen

Beweis: Wir zeigen: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

(c) Für eine beliebige Menge $F \subseteq M$ und beliebige inklusionsmaximale unabhängige Mengen $I_1, I_2 \subseteq F$ gilt: $|I_1| = |I_2|$.

$\Rightarrow$

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

Annahme: Greedy liefert I, aber es gibt $J \in \mathcal{U}$ mit $w(I) < w(J)$.

- o.B.d.A.: $I, J \subseteq F = \{e \in M \mid w(e) > 0\}$ und I, J sind inklusionsmaximal und unabhängig in F.

$|I| = |J|$ (da (c) gilt)

$I \rightarrow$ $w(e_1) \geq w(e_2) \geq \ldots \geq w(e_k) \geq \ldots \geq w(e_i)$

$J \rightarrow$ $w(e'_1) \geq w(e'_2) \geq \cdots \geq w(e'_k) \geq \cdots \geq w(e'_i)$

- Aus Annahme $w(I) < w(J)$ folgt: es gibt ein k, sodass $w(e_k) < w(e'_k)$

Matroide und Greedy-Algorithmen

Beweis: Wir zeigen: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

(c) Für eine beliebige Menge $F \subseteq M$ und beliebige inklusionsmaximale unabhängige Mengen $I_1, I_2 \subseteq F$ gilt: $|I_1| = |I_2|$.

$\Rightarrow$

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

Annahme: Greedy liefert I, aber es gibt $J \in \mathcal{U}$ mit $w(I) < w(J)$.

- o.B.d.A.: $I, J \subseteq F = \{e \in M \mid w(e) > 0\}$ und I, J sind inklusionsmaximal und unabhängig in F.

$|I| = |J|$ (da (c) gilt)

$I \rightarrow$ $w(e_1) \geq w(e_2) \geq \ldots \geq w(e_k) \geq \ldots \geq w(e_i)$

$J \rightarrow$ $w(e'_1) \geq w(e'_2) \geq \cdots \geq w(e'_k) \geq \cdots \geq w(e'_i)$

- Aus Annahme $w(I) < w(J)$ folgt: es gibt ein k, sodass $w(e_k) < w(e'_k)$
- Betrachte Menge $F' = \{e \in M \mid w(e) \geq w(e'_k)\}$, sei $I' = I \cap F'$ und $J' = J \cap F'$.
- Es gibt kein $e \in F' \setminus I'$ mit $I' \cup \{e\} \in \mathcal{U}$, da Greedy sonst e vor e_k gewählt hätte.
  $\Rightarrow$ I' ist inklusionsmaximal unabhängig in F'.
- Es gilt aber: $|I'| < |J'| \Rightarrow (c)$ ist nicht erfüllt.

Matroide und Greedy-Algorithmen

Definition: Matroid **(Definition 2.13)**
Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

Satz: Maximierungsprobleme auf Matroiden **(Satz 2.16)**
Für ein Unabhängigkeitssystem $(M, \mathcal{U})$ sind folgende Aussagen äquivalent:

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

(b) $(M, \mathcal{U})$ ist ein Matroid.

(c) Für eine beliebige Menge $F \subseteq M$ und beliebige inklusionsmaximale unabhängige Mengen $I_1, I_2 \subseteq F$ gilt: $|I_1| = |I_2|$.

Erinnerung:

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist.

Beweis: Gerade gezeigt: $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$

Matroide und Greedy-Algorithmen

Definition: Matroid **(Definition 2.13)**

Ein Unabhängigkeitssystem $(M, \mathcal{U})$ heißt *Matroid*, wenn für alle $I, J \in \mathcal{U}$ mit $|I| < |J|$, ein $e \in J \setminus I$ existiert, sodass $I \cup \{e\} \in \mathcal{U}$.

Satz: Minimierungsprobleme auf Matroiden **(Satz 2.17)**

Für ein Unabhängigkeitssystem $(M, \mathcal{U})$ mit Basissystem $\mathcal{B}$ sind äquivalent:

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Basissystem $\mathcal{B}$.

Erinnerung: $\mathcal{B}$ enthält genau die unabhängigen Mengen, die inklusionsmaximal sind.

(b) $(M, \mathcal{U})$ ist ein Matroid.

(c) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

Erinnerung:

Problem: Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$
Finde unabhängige Menge $U^\star \in \mathcal{U}$ sodass $w(U^\star)$ maximal ist.

Problem: Optimierungsproblem über dem Basissystem $\mathcal{B}$
Finde Basis $B^\star \in \mathcal{B}$ sodass $w(B^\star)$ minimal ist.

Beweis: $(b) \Leftrightarrow (c)$ wurde schon gezeigt. Wir zeigen: $(a) \Leftrightarrow (c)$.

Matroide und Greedy-Algorithmen

Beweis: $(b) \Leftrightarrow (c)$ wurde schon gezeigt. Wir zeigen: $(a) \Leftrightarrow (c)$.

(*a*) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Basissystem $\mathcal{B}$.

Problem Min

$\Rightarrow$

Problem Max

(*c*) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

- Sei $(M = \{e_1, \ldots, e_n\}, \mathcal{U})$ Unabhängigkeitssystem mit $w(e_1) \geq \cdots \geq w(e_n)$.
- Betrachte Basissystem $\mathcal{B}$ mit folgenden Gewichten:

$$w'(e) = \begin{cases} -w(e) \text{ wenn } w(e) > 0 \\ 0 \text{ sonst} \end{cases} \qquad \Rightarrow w'(e_1) \leq \cdots \leq w'(e_n)$$

- (a) gilt $\Rightarrow$ Greedy für Min liefert minimale Basis $B^\star \in \mathcal{B}$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide und Greedy-Algorithmen

Beweis: $(b) \Leftrightarrow (c)$ wurde schon gezeigt. Wir zeigen: $(a) \Leftrightarrow (c)$.

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Basissystem $\mathcal{B}$.

Problem MIN

$\Rightarrow$

Problem MAX

(c) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

- Sei $(M = \{e_1, \dots, e_n\}, \mathcal{U})$ Unabhängigkeitssystem mit $w(e_1) \geq \cdots \geq w(e_n)$.
- Betrachte Basissystem $\mathcal{B}$ mit folgenden Gewichten:

$$w'(e) = \begin{cases} -w(e) \text{ wenn } w(e) > 0 \\ 0 \text{ sonst} \end{cases} \qquad \Rightarrow w'(e_1) \leq \cdots \leq w'(e_n)$$

- (a) gilt $\Rightarrow$ Greedy für MIN liefert minimale Basis $B^\star \in \mathcal{B}$
- Greedy für MAX liefert $I = \{e \in B^\star \mid w(e) > 0\}$, da $B^\star \in \mathcal{U}$ und

Annahme: Es gibt $I' \in \mathcal{U}$ mit $w(I') > w(I)$ (I ist also nicht maximal)

- o.B.d.A. enthält I' nur Elemente mit $w(e) > 0$.
- Es folgt: $w'(I') < w'(I) = w'(B^\star)$.
- Da $I' \in \mathcal{U}$ gilt, ist I' in einer Basis B' enthalten mit $w'(B') < w'(B^\star)$. ↯

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide und Greedy-Algorithmen

Beweis: $(b) \Leftrightarrow (c)$ wurde schon gezeigt. Wir zeigen: $(a) \Leftrightarrow (c)$.

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Basissystem $\mathcal{B}$.

Problem MIN

$\Leftarrow$

Problem MAX

(c) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

- Sei $(M = \{e_1, \dots, e_n\}, \mathcal{U})$ Unabhängigkeitssystem mit $w(e_1) \leq \dots \leq w(e_n)$.
- Betrachte Problem MAX mit folgenden Gewichten:
  $w'(e) = m - w(e)$ mit $m = \max_{e \in M}\{w(e) + 1\}$ $\Rightarrow w'(e_1) \geq \dots \geq w'(e_n) > 0$
- (c) gilt $\Rightarrow$ Greedy für MAX liefert maximales $I^\star \in \mathcal{U}$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide und Greedy-Algorithmen

Karlsruher Institut für Technologie

Beweis: $(b) \Leftrightarrow (c)$ wurde schon gezeigt. Wir zeigen: $(a) \Leftrightarrow (c)$.

(*a*) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Basissystem $\mathcal{B}$.

Problem Min

$\Leftarrow$

Problem Max

(*c*) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

- Sei $(M = \{e_1, \ldots, e_n\}, \mathcal{U})$ Unabhängigkeitssystem mit $w(e_1) \leq \cdots \leq w(e_n)$.
- Betrachte Problem Max mit folgenden Gewichten:
  $w'(e) = m - w(e)$ mit $m = \max_{e \in M}\{w(e) + 1\}$ $\quad \Rightarrow w'(e_1) \geq \cdots \geq w'(e_n) > 0$
- (c) gilt $\Rightarrow$ Greedy für Max liefert maximales $I^\star \in \mathcal{U}$
- Greedy für Min liefert ebenfalls $B^\star = I^\star$ da $I^\star \in \mathcal{B}$ und

  $(w'(e) > 0)$
- Für alle $B \in \mathcal{B}$ gilt: $w(B) = \sum_{e \in B} \underbrace{(m - w'(e))}_{= w(e)} = |B| \cdot m - w'(B)$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Matroide und Greedy-Algorithmen

Beweis: $(b) \Leftrightarrow (c)$ wurde schon gezeigt. Wir zeigen: $(a) \Leftrightarrow (c)$.

(a) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Basissystem $\mathcal{B}$.

Problem MIN

$\Leftarrow$

Problem MAX

(c) Die Greedy-Methode liefert eine Optimallösung für das Optimierungsproblem über dem Unabhängigkeitssystem $(M, \mathcal{U})$.

- Sei $(M = \{e_1, \ldots, e_n\}, \mathcal{U})$ Unabhängigkeitssystem mit $w(e_1) \leq \cdots \leq w(e_n)$.
- Betrachte Problem MAX mit folgenden Gewichten:
  $w'(e) = m - w(e)$ mit $m = \max_{e \in M}\{w(e) + 1\}$ $\quad \Rightarrow w'(e_1) \geq \cdots \geq w'(e_n) > 0$
- (c) gilt $\Rightarrow$ Greedy für MAX liefert maximales $I^\star \in \mathcal{U}$
- Greedy für MIN liefert ebenfalls $B^\star = I^\star$ da $I^\star \in \mathcal{B}$ und
  $(w'(e) > 0)$
- Für alle $B \in \mathcal{B}$ gilt: $w(B) = \sum_{e \in B} \underbrace{(m - w'(e))}_{= w(e)} = \underbrace{\underbrace{|B| \cdot m}_{\text{konstant, da alle Basen gleich groß}} - \underbrace{w'(B)}_{\text{maximal für } B = I^\star = B^\star}}_{\text{minimal für } B = I^\star = B^\star}$

Lösung von Greedy ist optimal

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiele

Karlsruher Institut für Technologie

Beispiel 1: Das Finden einer minimalen Kreisbasis eines Graphen G entspricht dem Optimierungsproblem über dem Basissystem des Kreismatroids:
Gesucht ist eine Kreisbasis minimalen Gewichts.
$\Rightarrow$ MCB kann mittels Greedy-Algorithmus gelöst werden

Beispiele

Beispiel 1: Das Finden einer minimalen Kreisbasis eines Graphen G entspricht dem Optimierungsproblem über dem Basissystem des Kreismatroids:
Gesucht ist eine Kreisbasis minimalen Gewichts.
$\Rightarrow$ MCB kann mittels Greedy-Algorithmus gelöst werden

Beispiel 2: Sei $G = (V, E)$ ein zusammenhängender Graph. Das Mengensystem $(E, \mathcal{U})$ mit $\mathcal{U} = \{E' \subseteq E : E'$ induziert einen Wald in $G\}$ ist ein Matroid.
Die Basen dieses Matroids sind gerade Spannbäume.
$\Rightarrow$ Das Optimierungsproblem über dem Basissystem entspricht also dem Problem einen minimalen Spannbaum (MST) in G zu finden.

Die Korrektheit vieler MST-Algorithmen beruht darauf, dass die Greedy-Methode eine optimale Lösung liefert.