

Aufgabe 1

(a)

(b)

Wir wissen, dass ein k -regulärer bipartiter Graph mit $k \geq 1$ ein perfektes Matching besitzt. Zeige die Aussage durch Induktion.

Für $k = 1$ existiert ein perfektes Matching, da jeder Knoten nur genau eine inzidente Kante hat (k -regulär), kann kein anderes, kantendisjunktes Matching existieren. Die Aussage gilt also für $k = 1$.

Die Aussage gelte nun für n . Entfernt man aus einem $n + 1$ -regulären bipartiten Graphen die Kanten eines perfekten Matchings, so wird der Grad jedes Knotens um 1 verringert. Es ergibt sich also ein n -regulärer bipartiter Graph, der nach Voraussetzung n kantendisjunkte Matchings besitzt. Der $n + 1$ -reguläre bipartite Graph besitzt ein weiteres Matching, das soeben entfernt wurde um den n -regulären bipartiten Graph zu erhalten.

Die Aussage gilt somit für alle k durch vollständige Induktion.

(c)

Für gerade k :

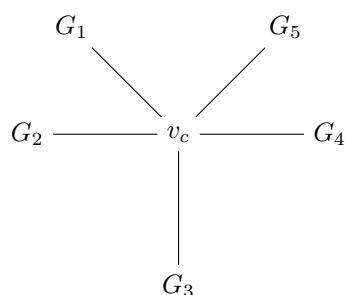
Sei G die $k+1$ -Clique. Jeder Knoten in G hat dann k Kanten, da er mit jedem anderen Knoten verbunden ist, der Graph also k -regulär. Die Gesamtanzahl der Knoten ist allerdings ungerade, was ein perfektes Matching bereits unmöglich macht.

Für ungerade k :

Sei G_i mit $i = 1, \dots, k$ eine Familie von Graphen. Jedes G_i ist k -regulär bis auf einen ausgezeichneten Knoten $v_{i,in}$, der nur $k - 1$ inzidente Kanten besitzt. Die Existenz solcher Graphen wird im folgenden gezeigt.

Aus diesen G_i lässt sich nun G erzeugen, indem ein weiterer Knoten v_c eingefügt wird, der mit einer Kante mit allen $v_{i,in}$ verbunden wird. Alle $v_{i,in}$ haben dann, ebenso wie v_c , k viele inzidente Kanten und der Graph G ist k -regulär.

Beispiel für $k = 5$.



Die Graphen G_i können wie folgt konstruiert werden: $G_i = (V_i, E_i)$ mit

$$V_i = \{v_{i,in}, v_{i,1}, \dots, v_{i,k-1}, v_{i,t_1}, v_{i,t_2}\}$$

und

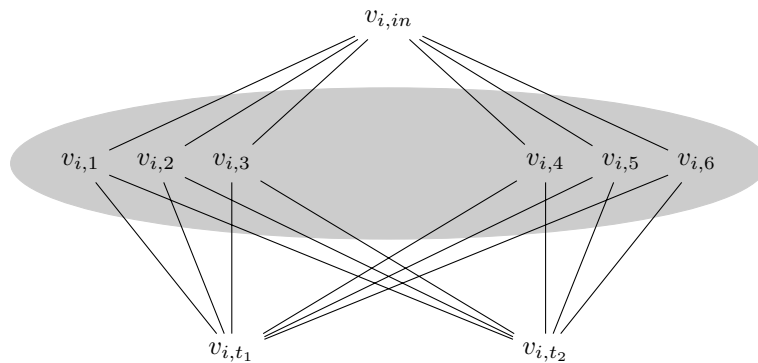
$$\begin{aligned} E_i = & \{ \{v_{i,in}, v_{i,j}\} \mid j \in \{1, \dots, k-1\} \} \\ & \cup \{ \{v_{i,t_1}, v_{i,j}\}, \{v_{i,t_2}, v_{i,j}\} \mid j \in \{i, \dots, k-1\} \} \\ & \cup E_{k-3\text{-reg}} \end{aligned}$$

$E_{k-3\text{-reg}}$ ist dabei eine Menge von Kanten, so dass der Graph

$$G_{k-3\text{-reg}} = (\{v_{i,1}, \dots, v_{i,k-1}\}, E_{k-3\text{-reg}})$$

$k - 3$ -regulär ist.

Beispiel eines G_i für $k = 7$. Der graue Bereich soll den $k - 3$ -regulären Teilgraphen andeuten, in dem weitere Kanten existieren.



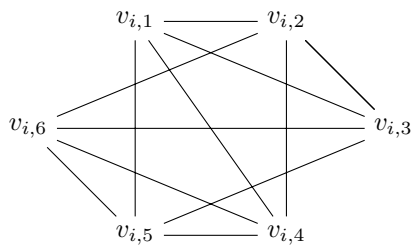
Für diesen $k - 3$ -regulären Teilgraphen kann beispielsweise eine $k - 1$ -Clique ohne die Kanten

$$E_- = \{\{v_j, v_{k-j}\} \mid j \in \{1, \dots, \frac{k-1}{2}\}\}$$

sein, also

$$G_{k-3\text{-reg}} = (\{v_{i,1}, \dots, v_{i,k-1}\}, \{\{v_j, v_l\} \mid j \neq l \wedge j + l \neq k\})$$

Beispiel für $k = 7$.



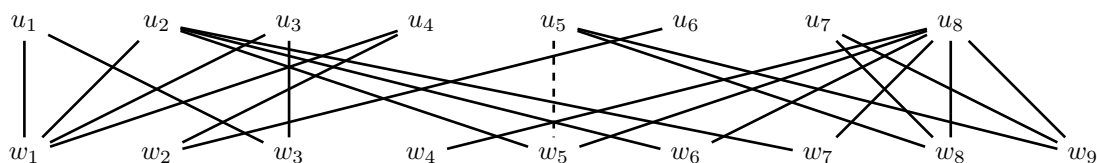
Der so entstandene Graph $G_i = (V_i, E_i)$ erfüllt die an G_i gestellten Bedingungen.

Jeder Graph G_i hat genau $k + 2$, also ungerade viele Knoten. Bei einem Matching innerhalb von G_i bleibt also stets ein Knoten übrig. Die einzige Möglichkeit ein perfektes Matching zu erreichen besteht darin, dass dieser übrige Knoten $v_{i,in}$ ist. So bleiben jedoch alle $v_{i,in}$ sowie v_c übrig. Da in dem durch diese Knoten induzierten Teilgraphen alle $v_{i,in}$ jedoch nur genau mit v_c verbunden sind, kann nur eine weitere Matchingkante zum Matching hinzugefügt werden. Die anderen $v_{i,in}$ bleiben als isolierte Knoten zurück. Daher kann kein perfektes Matching gefunden werden.

Aufgabe 2

Suche in jedem Schritt einen M -erweiternden Pfad. Wähle den Pfad so, dass der Pfad im u_i mit kleinstem i beginnt. Wähle den Pfad weiterhin so, dass er im w_k mit kleinstem k endet.

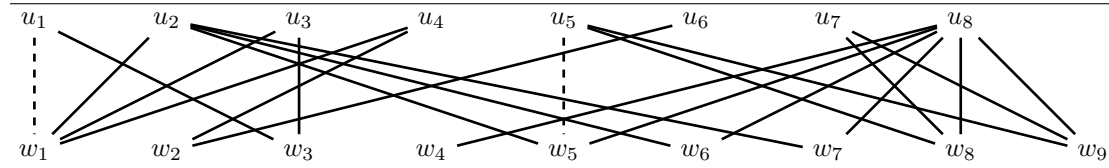
Die im jeweiligen Schritt im Matching enthaltenen Kanten sind gestrichelt markiert.



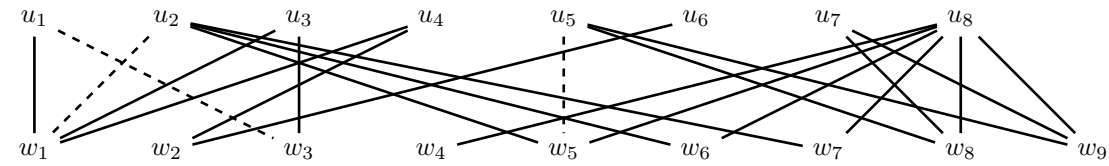
Pfad: u_1w_1

Optimierung B

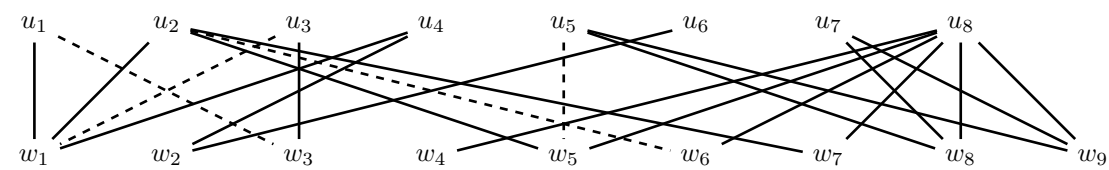
Übung 4



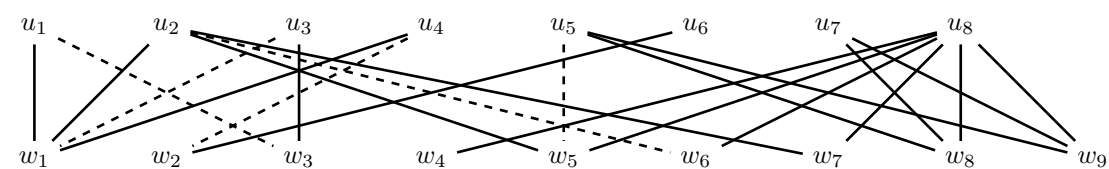
Pfad: $u_2 w_1 u_1 w_3$



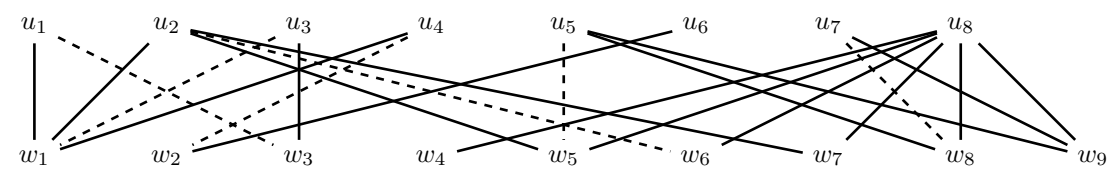
Pfad: $u_3 w_1 u_2 w_6$



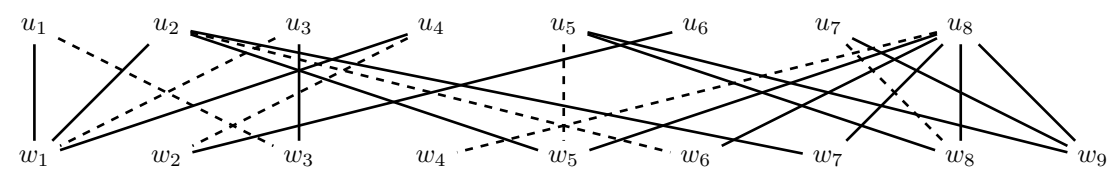
Pfad: $u_4 w_2$



Pfad: $u_7 w_8$



Pfad: $u_8 w_4$



Keine weiteren M-erweiternden Pfade vorhanden.

$$\Rightarrow M = \{(1, 3), (2, 6), (3, 1), (4, 2), (5, 5), (7, 8), (8, 4)\}$$

Listing 1: Algorithmus zum Berechnen des Vertex Covers

```

1  $T' \leftarrow U \setminus \bigcup_{e \in M} e$ 
2 Wiederhole:
3    $T \leftarrow T'$ 
4    $T' \leftarrow T \cup (\{w \in W \mid \exists e \in E \setminus M : e = \{w, v\}, v \in T\})$ 
5    $T' \leftarrow T' \cup (\{u \in U \mid \exists e \in M : e = \{u, v\}, v \in T\})$ 
6 Solange  $T' \neq T$ 
7 Vertex Cover  $C = (U \setminus T) \cup (W \cap T)$ 

```

Ausführen des Algorithmus:

$$T' = \{u_6\}$$

$$T = T'$$

$$T' = \{u_6, w_2\}$$

$$T' = \{u_6, w_2, u_4\}$$

$$T = T'$$

$$T' = \{u_6, w_2, u_4, w_1\}$$

$$T' = \{u_6, w_2, u_4, w_1, u_3\}$$

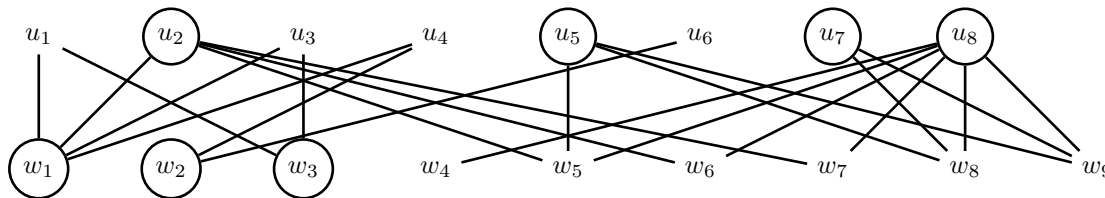
$$T = T'$$

$$T' = \{u_6, w_2, u_4, w_1, u_3, w_3\}$$

$$T' = \{u_6, w_2, u_4, w_1, u_3, w_3, u_1\}$$

$$T = T'$$

$$(U \setminus T) \cup (W \cap T) = \{u_2, u_5, u_7, u_8, w_1, w_2, w_3\}$$



Aufgabe 3

Aufgabe 4

Zum Beweis betrachten wir ein Verfahren, ein perfektes Matching auf einem beliebigen Baum zu konstruieren, falls eines existiert. Das Verfahren wird eindeutig sein. Das erzeugte Matching ist somit auch eindeutig (oder existiert nicht) und jeder Baum hat somit maximal ein perfektes Matching.

Betrachte folgenden Algorithmus auf dem Baum $B = (V, E)$:

1. Falls $|V| \bmod 2 \neq 0$: Gebe Fehler aus
2. OBdA: wähle einen beliebigen Knoten des Baumes als Wurzelknoten. So ergibt sich ein natürlicher Begriff von Blattknoten und dem Wurzelknoten.
3. Solange Knoten in V existieren:
 - (a) Wähle den Blattknoten v mit maximaler Distanz zum Wurzelknoten
 - (b) Füge die von v ausgehende Kante $e = \{v, w\}$ in das Matching ein
 - (c) Falls w weitere Kindknoten hat: Breche mit Fehler ab
 - (d) Entferne v und w aus V sowie alle zu v und w inzidenten Kanten aus E

Behauptung: Der Baum B hat ein perfektes Matching \Leftrightarrow der Algorithmus findet ein perfektes Matching auf B

Beweis:

\Rightarrow : Es existiert ein perfektes Matching auf B , für jeden Knoten ist also eine inzidente Kante in M enthalten. Insbesondere ist also für den von der Wurzel am weitesten entfernten Blattknoten v eine inzidente Kante in M enthalten. Ein Blattknoten hat nur eine inzidente Kante $e = \{v, w\}$, also muss genau diese in M enthalten sein.

Mit e ist nun auch bereits eine zu w inzidente Kante in M enthalten, es darf also keine weitere zu w inzidente Kante in M enthalten sein. Sie können daher im folgenden völlig ignoriert werden, bzw aus dem Baum gelöscht werden.

Hierbei besteht prinzipiell die Möglichkeit, dass w weitere Kindknoten besitzt und der Baum somit in Teilbäume aufgeteilt wird. Da v jedoch der Blattknoten mit der größten Distanz zur Wurzel ist, kann w nur weitere einzelne Kindknoten, nicht jedoch Kindknoten mit weiteren Kindern haben. Ist dies der Fall, so liegt ein Fehler vor, da auch dieser andere Kindknoten lediglich eine Kante mit w gemeinsam hat, w jedoch keine zwei inzidenten Kanten in M haben darf.

Der Baum wird so iterativ verkleinert und immer weitere Kanten in M eingefügt. Der Restbaum besteht zu jedem Zeitpunkt aus den Knoten und Kanten, die noch für weitere Matchingkanten zur Verfügung stehen. Sind keine Knoten mehr vorhanden, wurden alle Knoten durch Matchingkanten abgedeckt und ein perfektes Matching gefunden. Dieses Verfahren entspricht genau dem Algorithmus, der daher ein perfektes Matching findet, falls eines existiert.

Die Terminierung ist sichergestellt, da der Algorithmus abbricht, sobald der Graph keine Knoten mehr besitzt und in jeder Iteration zwei Knoten aus dem Graphen entfernt werden.

\Leftarrow : Der Algorithmus findet ein Matching auf B . Da der Algorithmus erst terminiert, wenn alle Knoten durch das Matching abgedeckt sind, muss das Matching perfekt sein. Die Matchingeigenschaften werden durch das Löschen der bereits abgedeckten Knoten sichergestellt.

Aufgabe 5

Sei M ein Matching und M^* ein Maximales Matching auf dem Graphen $G = (V, E)$.

Betrachte $H = (V, M \cup M^*)$. Da für jeden Knoten v gilt, dass $\deg(v) \leq 2$ besteht H lediglich aus Pfaden und Kreisen. Wir werden zeigen, dass M maximal $|M^*|$ Kanten mehr als M^* beinhalten kann. Dazu versuchen wir $\Delta = \frac{|M| - |M^*|}{|M^*|}$ zu maximieren. (Maximale Matchings mit 0 Kanten können wir ignorieren, da hier ohnehin $|M| = |M^*| = 0$ gelten muss.)

Da M und M^* Matchings, müssen Kreise abwechselnd Kanten auf M und M^* enthalten, die Anzahl der Kanten aus M und aus M^* sind also für jeden Kreis identisch. Kreise erhöhen Δ also nicht.

Wir betrachten nun alle Pfade und vergleichen die Anzahl von Kanten dieser Pfade aus M und M^* . Pfade der Länge 1 können nur in M^* oder in M und M^* enthalten sein. Sie erhöhen Δ daher nicht. Pfade gerader Länge müssen, da M und M^* Matchings, gleich viele Kanten aus M und M^* enthalten. Sie erhöhen Δ ebenfalls nicht. Pfade ungerader Länge können entweder k Kanten aus M und $k + 1$ Kanten aus M^* (Δ wird verringert) enthalten, oder umgekehrt k Kanten aus M und $k - 1$ Kanten aus M^* (Δ wird erhöht).

Um Δ zu maximieren kommen also lediglich Pfade ungerader Länge mit Länge mindestens 3 mit mehr Kanten aus M als aus M^* . Offensichtlich trägt ein solcher Pfad der Länge k eine Differenz von 1 zu Δ bei, wobei M^* um $(k - 1)/2$ vergrößert wird.

Das maximale Delta ergibt sich bei solchen Pfaden der Länge 3: $\Delta = \frac{2-1}{1} = 1$.

Es gilt nach Definition von Δ

$$|M^*| \cdot \frac{1}{\Delta} + |M^*| = |M|$$

Da $\Delta \leq 1$ gilt also für beliebige M : $2 \cdot |M^*| \geq |M|$. □