

# 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

## Fokus auf Extremen bei Kernreduktionen

Bsp.: Buss' Regel bei VC

Wie wäre diese Regel für MMVC umzuformulieren?!

## MMVC-Regeln

**Reduktionsregel 1** *Lösche isolierte Knoten (Beibehaltung des Parameters)*

**Reduktionsregel 2** *Ist  $v$  ein Knoten vom Grad größer gleich  $k$ , dann JA.*

**Satz 1** *MMVC gestattet einen Kern der Größe  $k^2$  (gemessen in der Knotenzahl).*

Wie sieht “vollständiges Regelsystem” aus ?

## Zurück zu Nonblocker: Knoten vom Grad 1

### Erweiterte Instanz (mit Annotation):

Graph mit ausgezeichnetem Knoten in der dominierenden Menge (Dominator  $d$ ).  
Wegen seiner Funktion heißt  $d$  auch **Katalysator**.

**(Hinter-)Grund:** Ist  $x$  Nachbarknoten eines Blattes, so kann  $x$  mit Dominator  $d$  zu neuem Dominator  $[x, d]$  verschmolzen (identifiziert) und das Blatt entfernt werden.

## Mehr Mathematik...

**Satz 2 (Blank 1973; McCuaig / Shepherd 1989)** *Hat ein zusammenhängender Graph  $G = (V, E)$  Minimalgrad zwei und gehört er nicht zu einem von sieben "Ausnahme-Graphen" (jeweils mit maximal sieben Knoten), so besitzt  $G$  eine dominierende Menge mit höchstens  $2/5 \cdot |V|$  vielen Knoten.*

Wir können diesen Satz ausnutzen für einen Problemerkernalgorithmus durch Einführen von **Katalysator**regeln und **Dekatalysator**regeln.

## Die “alten” Regeln (verallgemeinert)

Sei  $(G, c, k_d)$  eine Instanz von NB (annotiert mit Katalysator  $c$ ).

**Reduktionsregel 3** *Ist  $C$  eine vollständige Graphkomponente ohne  $c$ , reduziere zu  $(G - C, c, k_d - (|C| - 1))$ .*

**Reduktionsregel 4** *Gibt es  $x \in V(G)$  mit  $U \subseteq N(x)$  sodass  $N(U) \subseteq U \cup \{x\}$  und  $c \notin U$ , dann reduziere zu  $(G[c = x], [c, x], k_d - |U|)$ .*

## (De-)Katalysatorregeln

**Katalysatorregel** Ist  $(G, \kappa_d)$  eine Nonblocker-Instanz  $G = (V, E)$ , dann ist  $(G', c, \kappa_d)$  eine äquivalente annotierte Instanz, wobei  $c \notin V$  ein neuer Knoten ist und  $G' = (V \cup \{c\}, E)$ .

**Dekatalysatorregel** Sei  $(G, c, \kappa_d)$  eine annotierte Nonblocker-Instanz. Die neue nicht-annotierte Nonblocker-Instanz  $(G', \kappa'_d)$  entsteht wie folgt:

Verbinde  $c$  mit drei neuen Knoten  $u, v$  und  $w$ . Ferner führe neue Kanten  $uv$  und  $vw$  ein. Alle andere Knoten und Kanten bleiben unverändert. Dies beschreibt den neuen Graphen  $G'$ . Setze  $\kappa'_d = \kappa_d + 3$ .

**Achtung:** Keine echte Kernreduktion!



## Eine alternative Idee

katalytisches Verzweigen anstelle der Katalysatorregel.

Die folgenden Folien zeigen eine “falsche Wahl”.

Es wird eine weitere Regel benutzt, um sämtliche Ausnahmegrphen abzuhandeln (insgesamt werden damit alle Graphen mit zwei benachbarten Knoten vom Grad zwei reduziert):

**Reduktionsregel 5**  $\deg(v) = \deg(v') = 2, v \neq c, v' \neq c, N(v) = \{u, v'\},$   
 $N(v') = \{u', v\}, u \neq u'$   
 $\rightsquigarrow (G[u = u'] - \{v, v'\}, c, k_d - 2)$

---

**Algorithm 1** A kernelization algorithm for NB

---

**Input(s):** an instance  $(G, k_d)$  of NB

**Output(s):** an equivalent instance  $(G', k'_d)$  of NB with  $V(G') \subseteq V(G)$ ,  $|V(G')| \leq 5/3 \cdot k'_d$  and  $k'_d \leq k_d$  OR ✓

**if**  $G$  has more than seven vertices **then**

    Apply the catalyzation rule.

    Exhaustively apply Rules 3 and 4 for neighborhoods  $U$  up to size two.

    Apply the decatalyzation rule.

    {This leaves us with a reduced instance  $(G', k'_d)$ .}

**if**  $|V(G')| > 5/3 \cdot k'_d$  **then**

        return ✓

**else**

        return  $(G', k'_d)$

**end if**

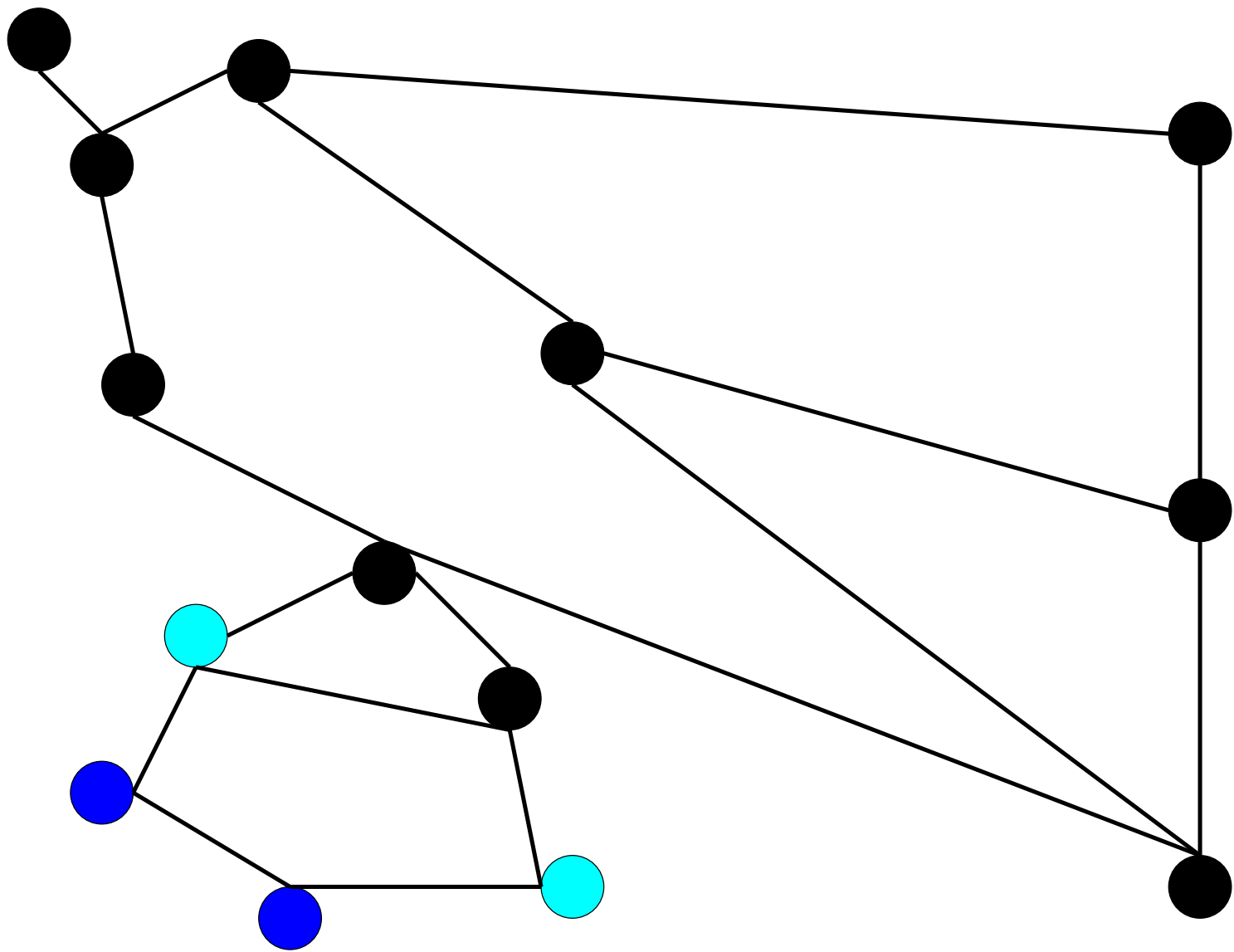
**else**

    Solve by table look-up and answer accordingly.

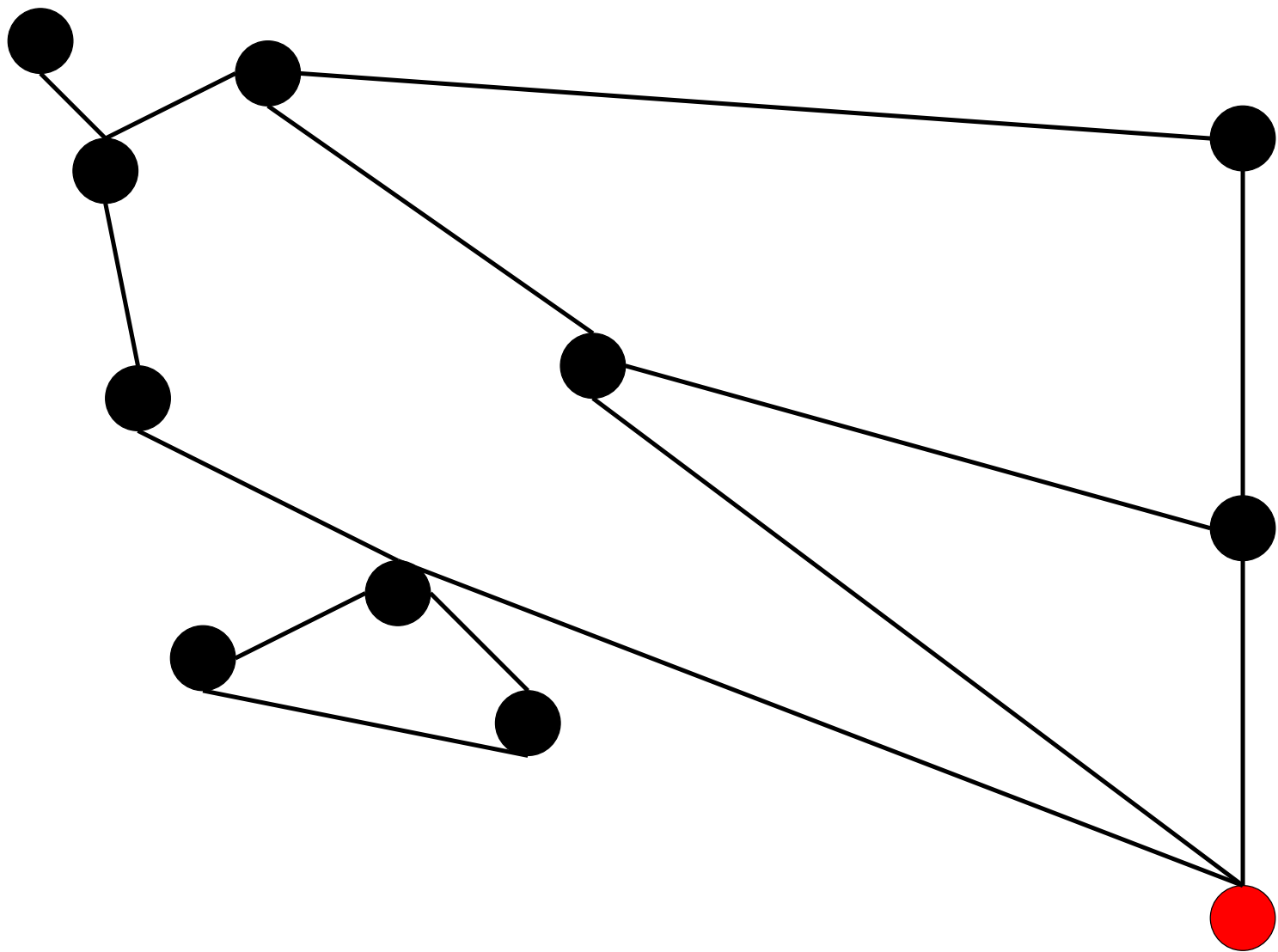
**end if**

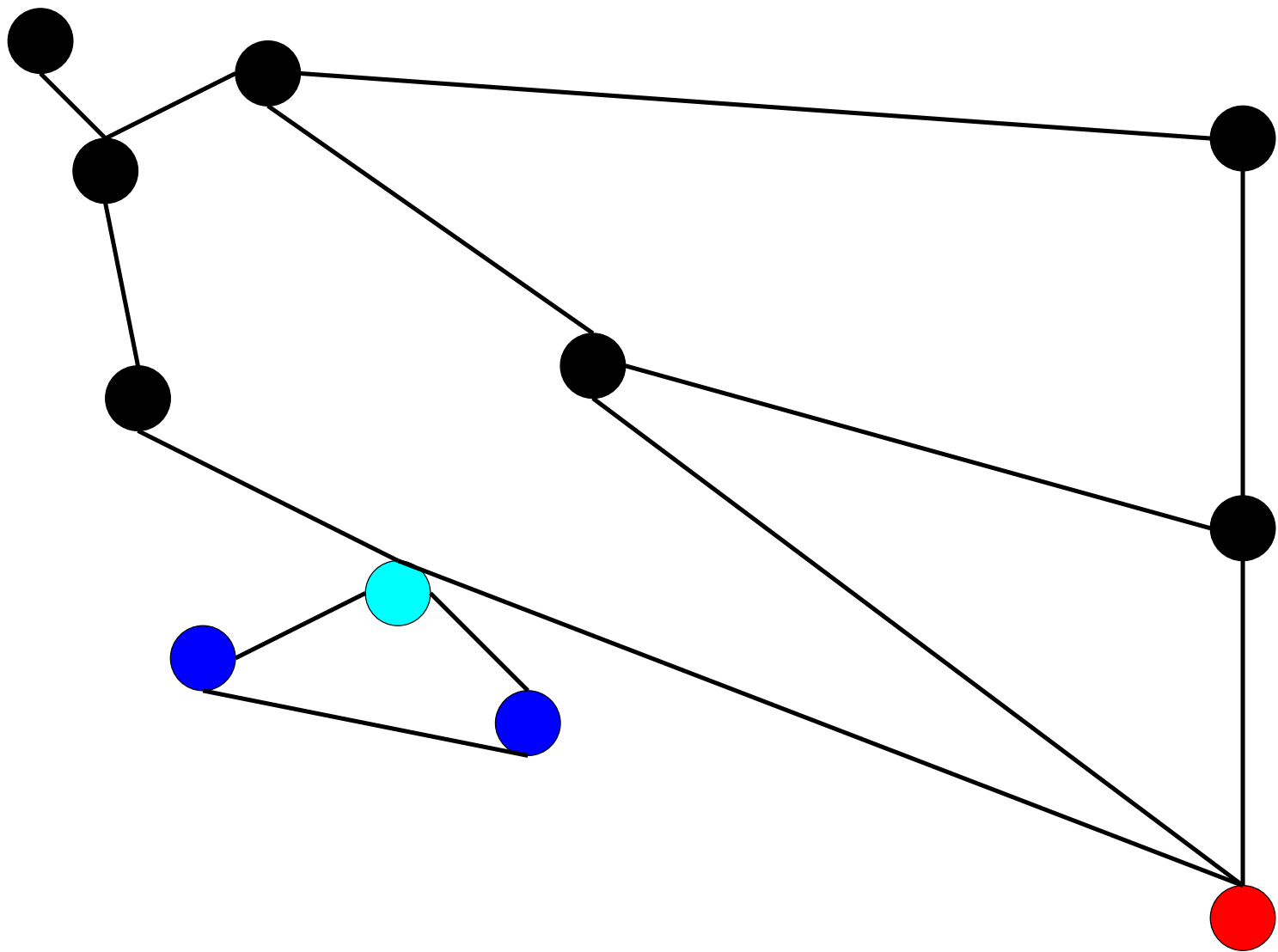
---

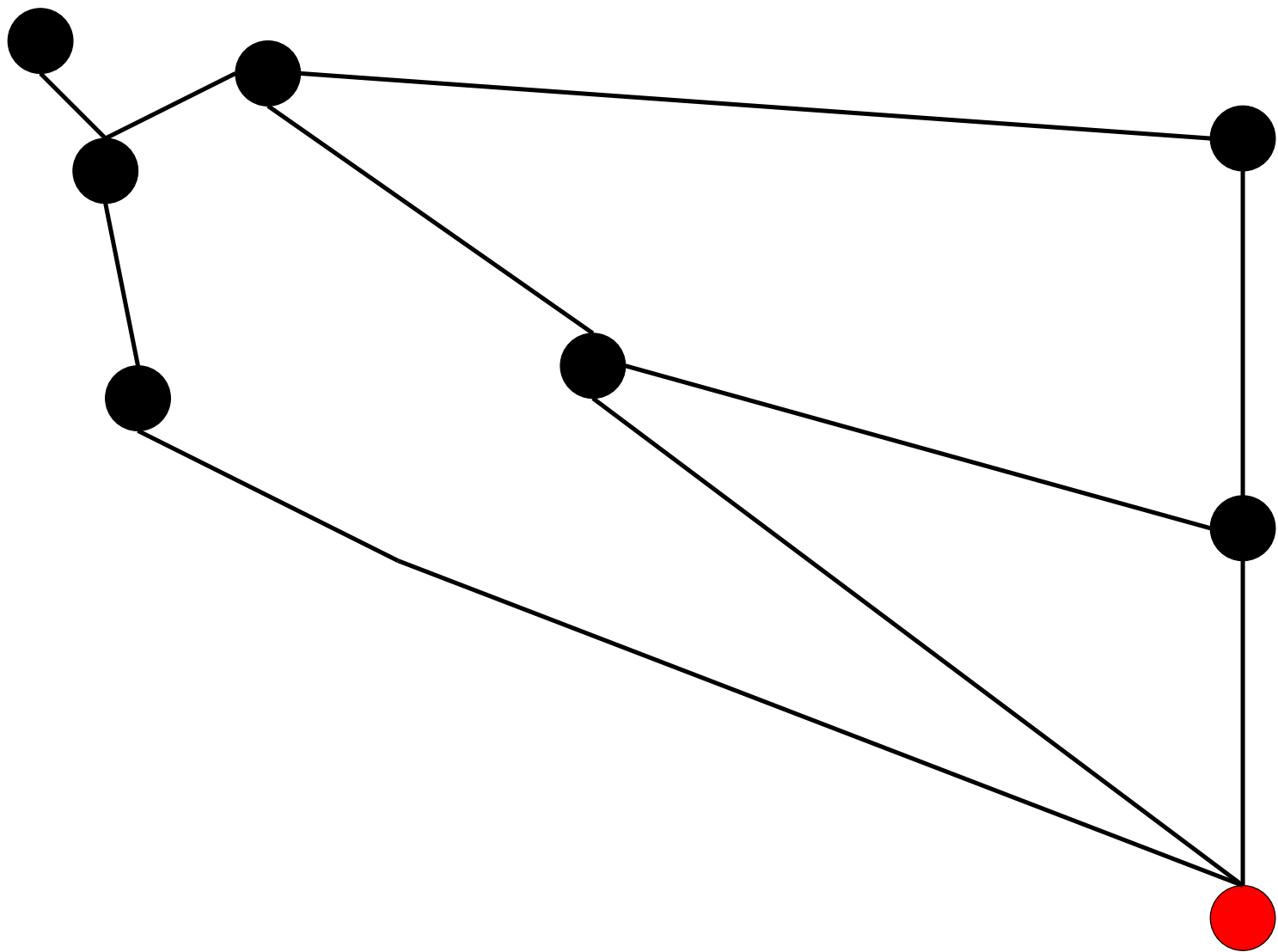




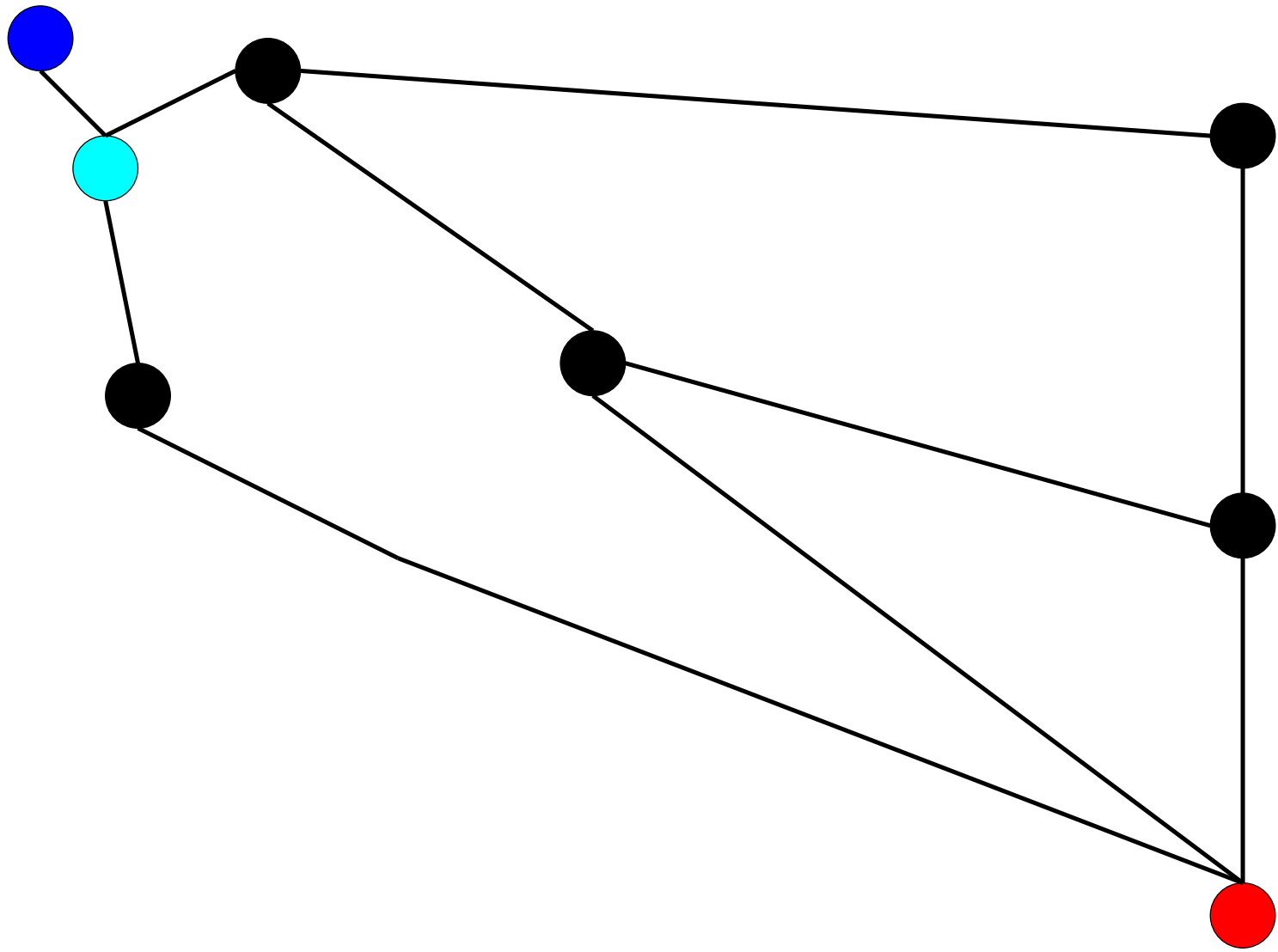


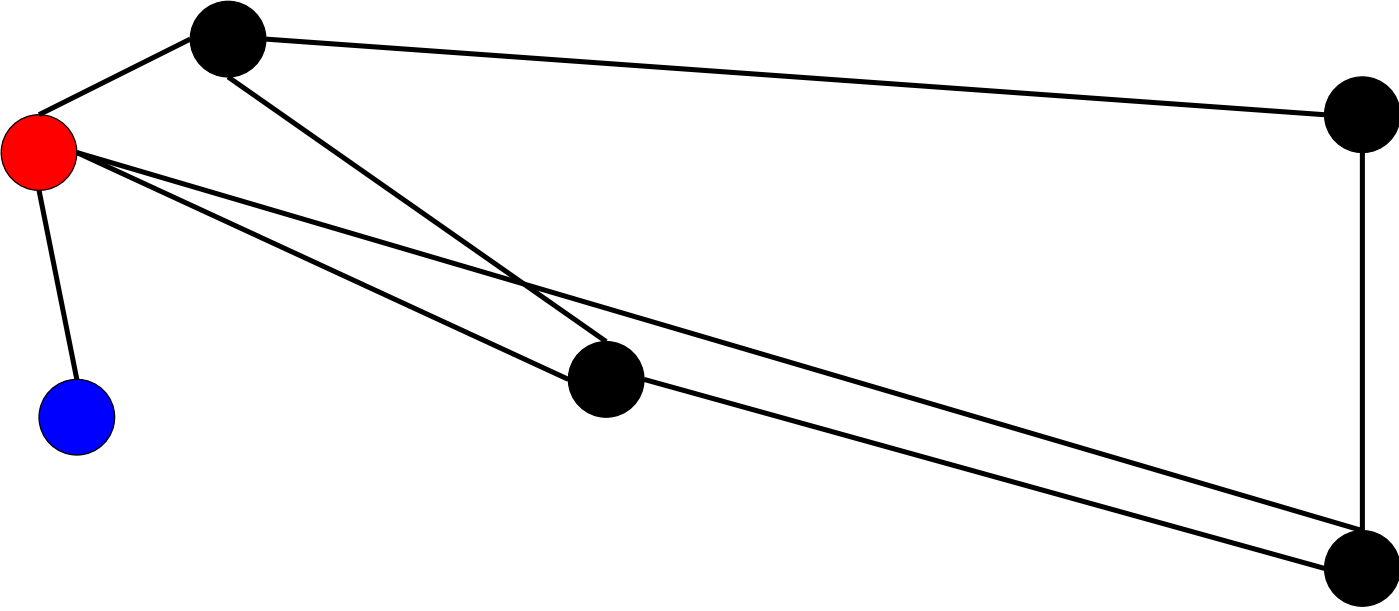


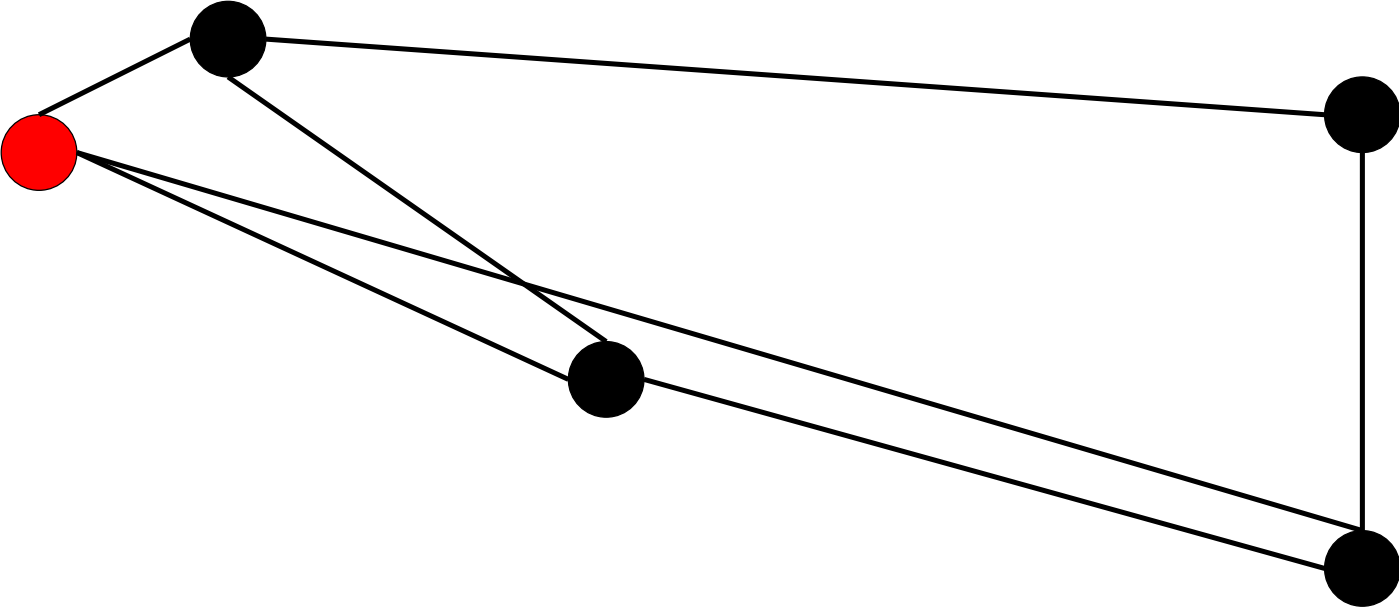


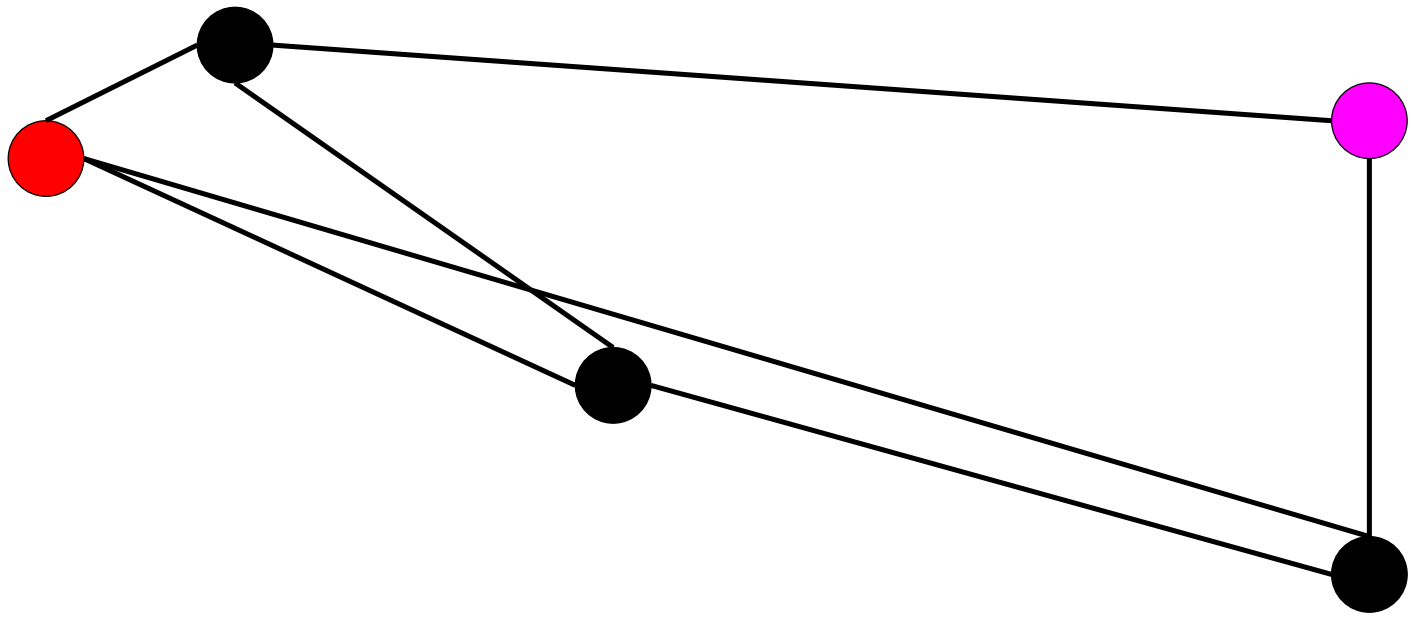


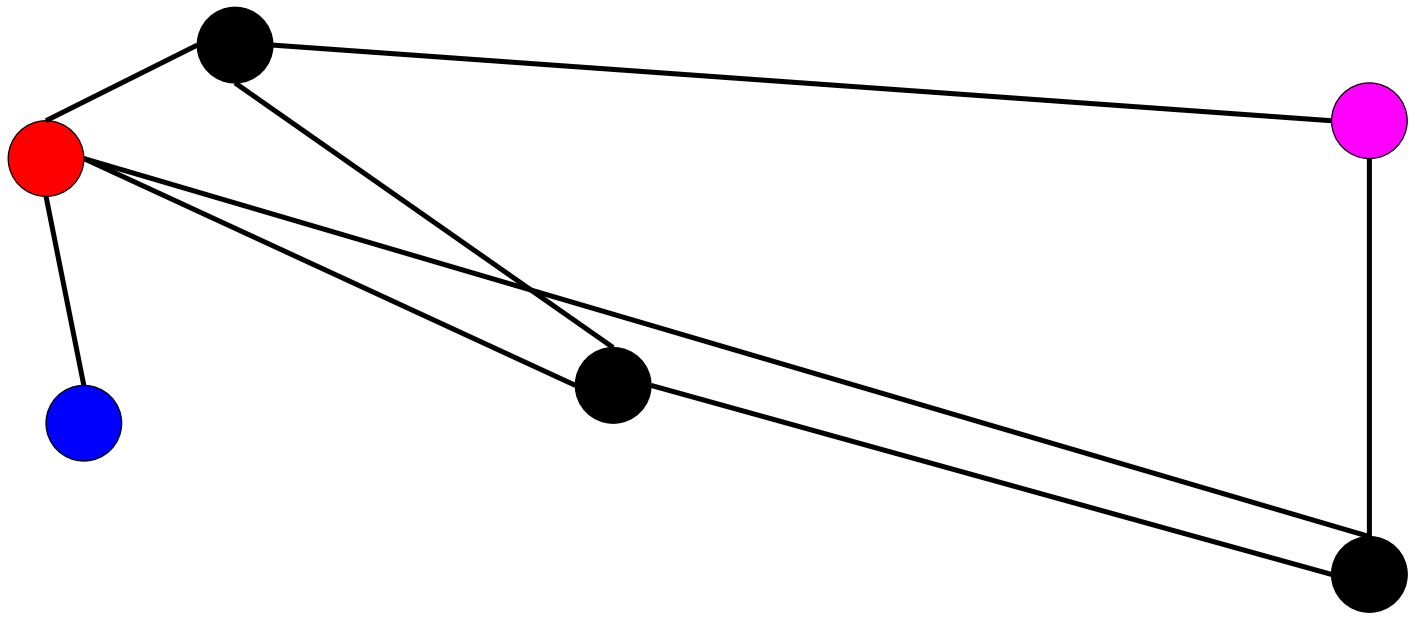


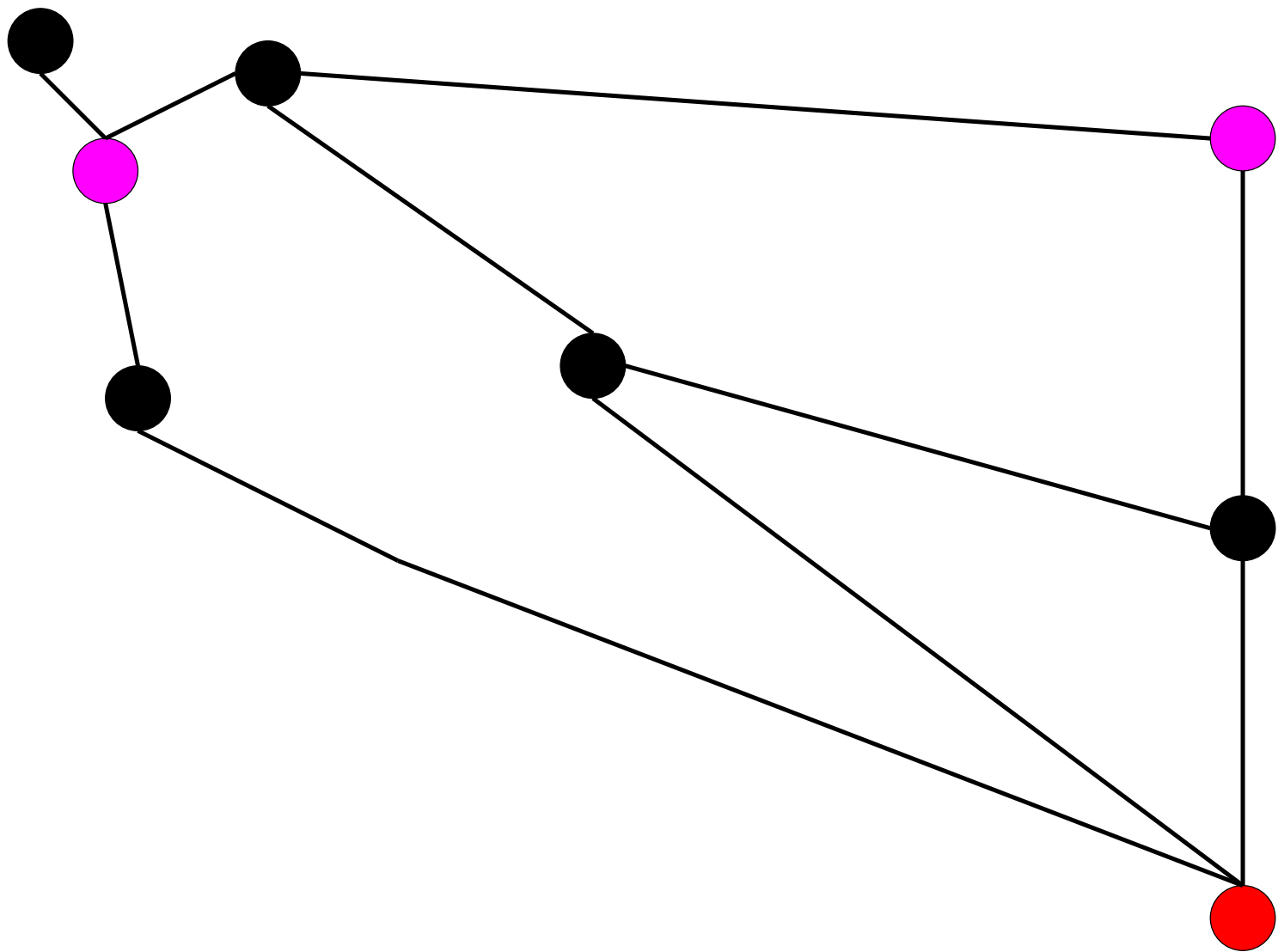


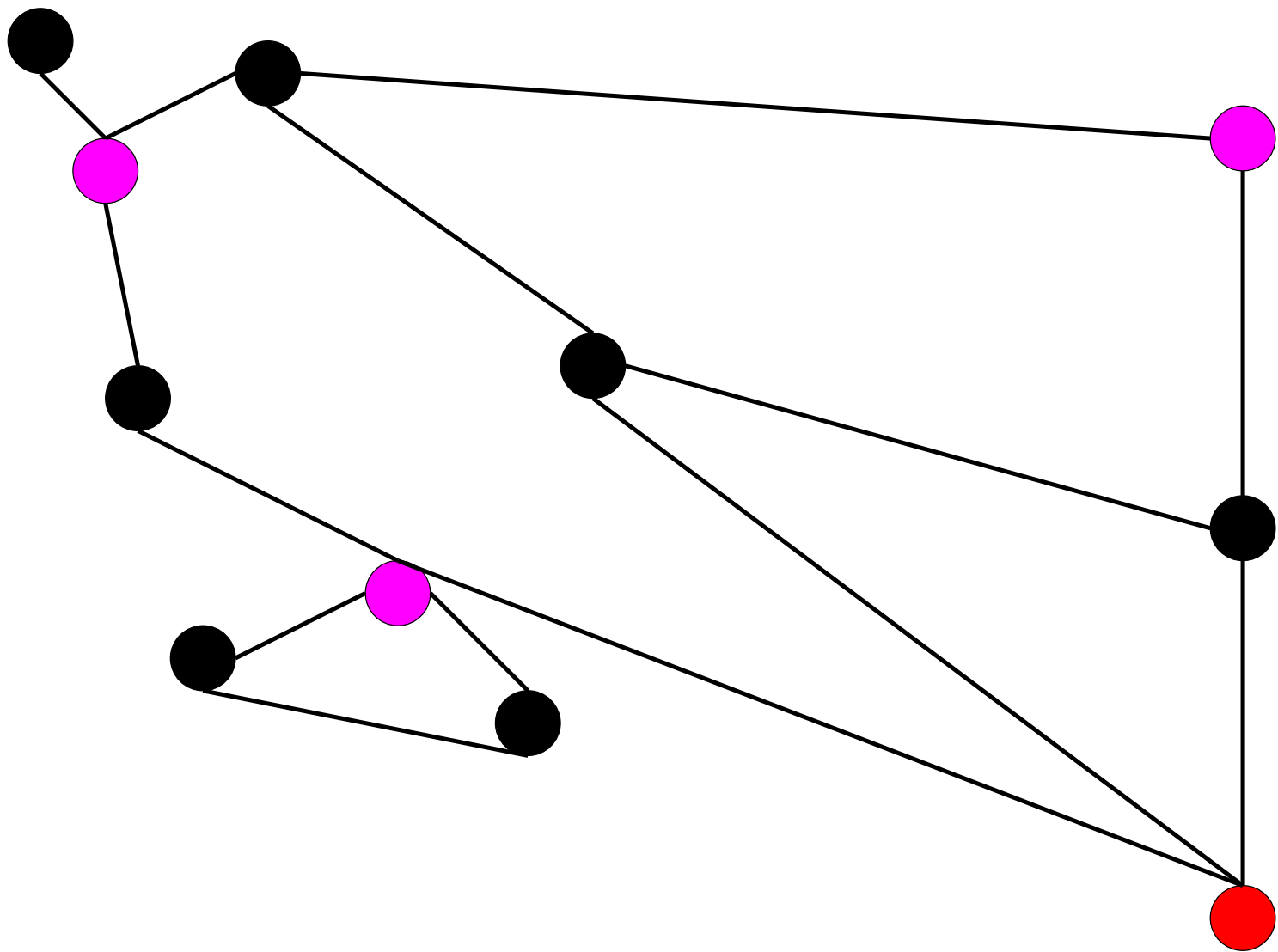


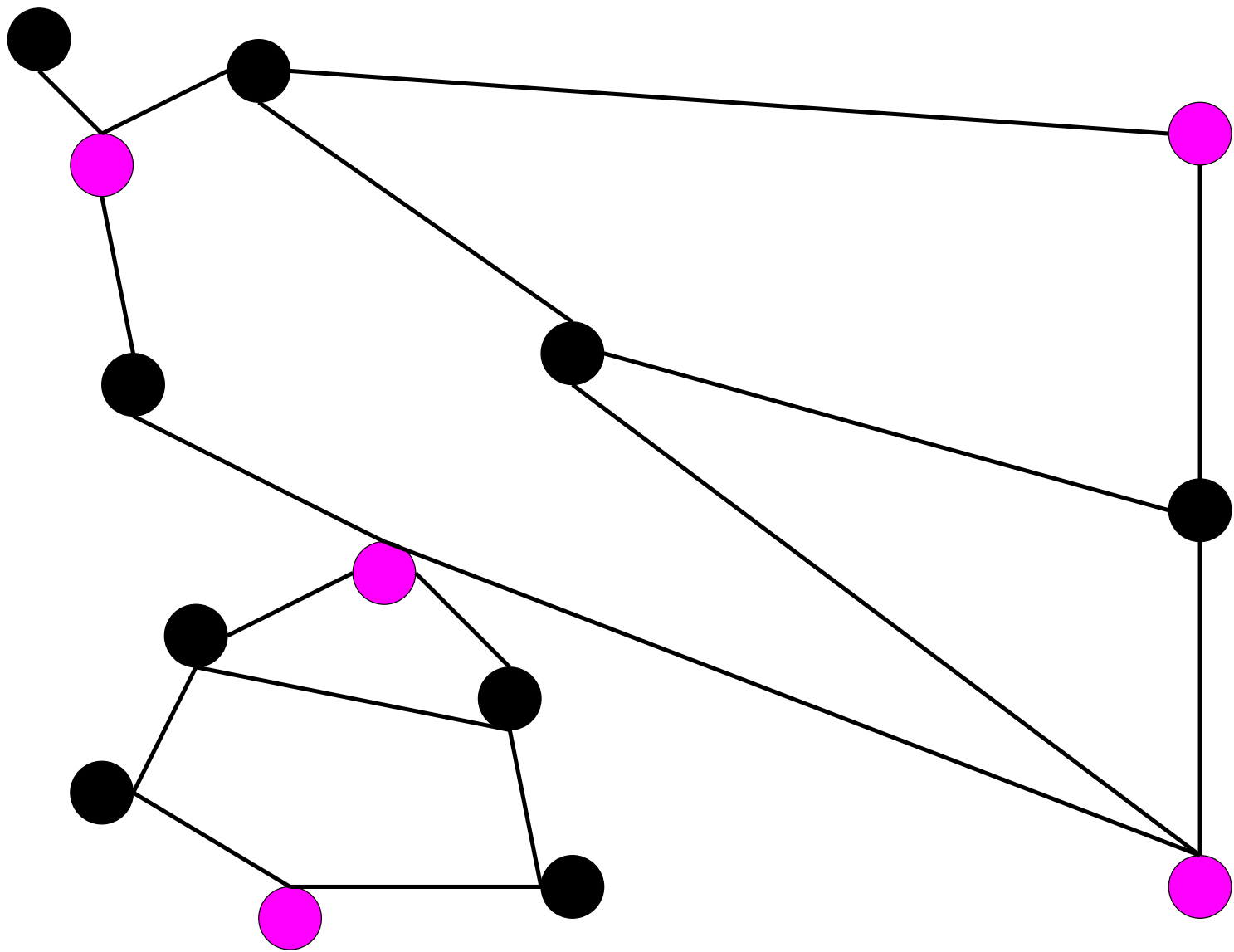




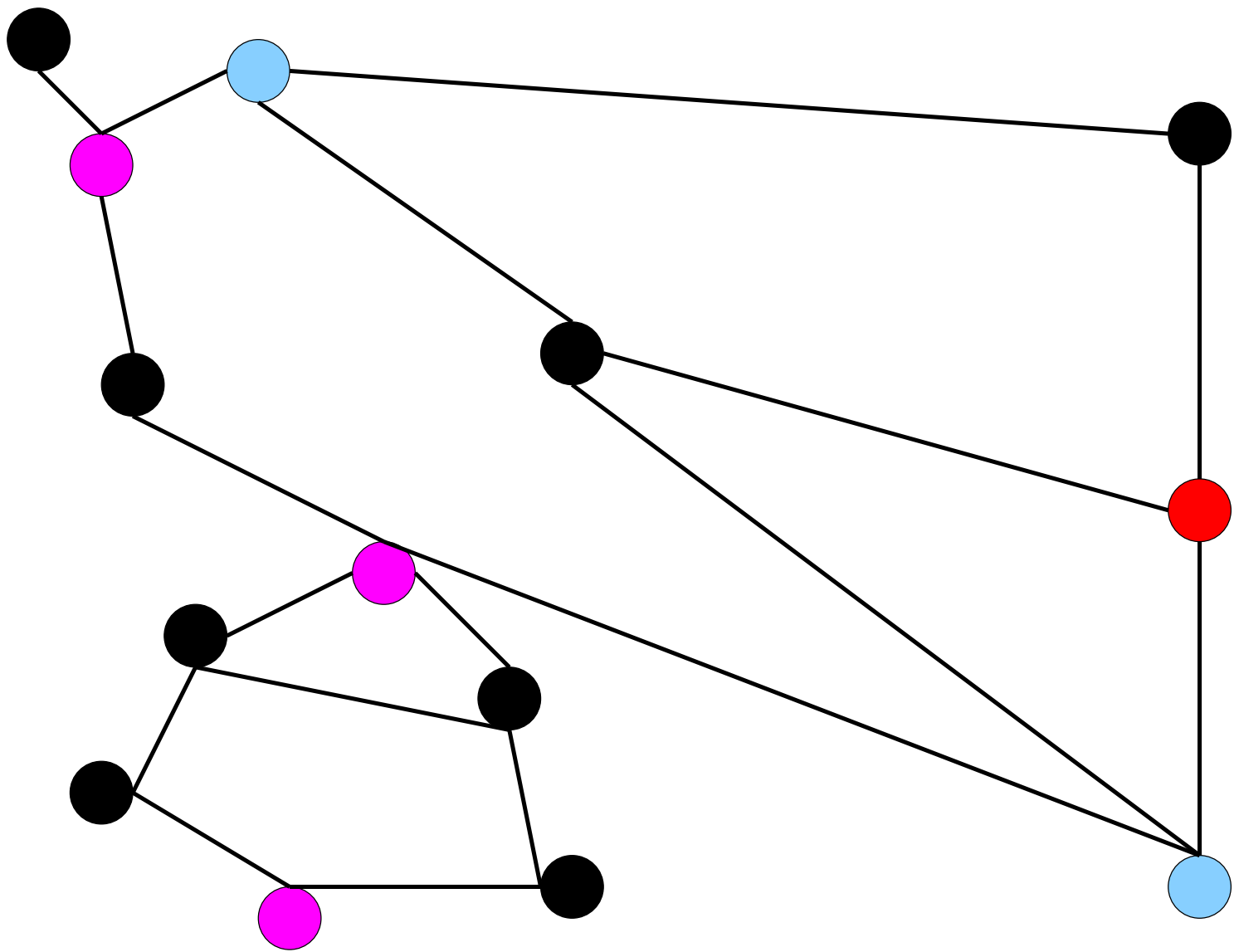












Die Stirling-Formel liefert:

**Folgerung 3** *Nonblocker kann in Zeit  $\mathcal{O}^*(3.0701^{k_d})$  gelöst werden.*

**Hinweis:** Die  $\mathcal{O}^*$ -Notation “vernachlässigt” sowohl konstante Faktoren als auch polynomielle Anteile und ist daher sehr geeignet für FPT.

## Eine weitere Reduktionsidee für VC: Kronen

Es sei  $G = (V, E)$  ein Graph. Eine unabhängige Menge  $I$  von  $G$  heißt **Krone** gdw. für alle nicht-leere Mengen  $U \subseteq N(I)$ :  $|U| \leq |N(U) \cap I|$ .

Eine Krone  $I$  hat die folgende Eigenschaft (**Heiratssatz von Hall**):

Sei  $G = (V_1, V_2; E)$  ein bipartiter Graph mit  $|V_1| \leq |V_2|$ .

$G$  besitzt ein Matching der Größe  $|V_1|$  gdw. für alle  $U \subseteq V_1$  ist  $|U| \leq |N(U)|$ .

$H := N(I)$  wird in  $I$  (gepaart) gematched. ( $H$  heißt auch **Kopf** der Krone.)

**Reduktionsregel 6** *Ist  $(G = (V, E), k)$  eine VC Instanz und  $I \subseteq V$  eine Krone mit Kopf  $H = N(I)$ , so reduziere zu  $(G - (I \cup H), k - |H|)$ .*

**Problem:** Wie findet man (große) Kronen?

**Lemma 4** *Alg. 3 kombiniert mit Regel 6, angesetzt auf  $(G, k)$ , liefert entweder korrekt NEIN oder liefert eine reduzierte Instanz  $(G', k')$  von VC mit  $|V(G')| \leq 3k$ .*

Beobachte hierzu:

- Finden wir ein Matching  $M$  (i.e.,  $M_1$  oder  $M_2$ ) in  $G$  mit  $|M| > k$ , so ist  $(G, k)$  eine  $\times$ -Instanz wegen dem Lemma von König und Egervary.
- Enthalten weder  $M_1$  noch  $M_2$  mehr als  $k$  Kanten, so hat  $G[M_1 \cup M_2]$  höchstens  $3k$  Knoten.

---

**Algorithm 2** An efficient algorithm to find a crown in a graph, called GETCROWN

---

**Input(s):** an graph  $G$

**Output(s):** a crown  $I \subseteq V(G)$

Greedily find a **maximal** matching  $M_1$  of  $G$ .

Let  $O$  (outsiders) be the set of vertices from  $G$  that are not in the matching.

**if**  $O \neq \emptyset$  **then**

    Let  $I' := \emptyset$ .

    Find a **maximum** matching  $M_2$  in the bipartite graph  $B$  with vertex set  $O \cup N(O)$  and edge set  $E_B = \{uv \mid u \in O \wedge v \in N(O)\}$ .

    Let  $I$  collect all vertices in  $O$  that are not matched by  $M_2$ .

**while**  $I' \neq I$  **do**

        Let  $I' := I$ . Let  $H := N(I)$ .

        Let  $I := I \cup \{u \in O \mid \exists v \in H (uv \in M_2)\}$ .

**end while**

**else**

$I := \emptyset$ .

**end if**

---

---

**Algorithm 3** Using crowns for kernels in the case of VC

---

**Input(s):** an instance  $(G, k)$  of VC

**Output(s):**  $\times$  if the VC instance has no solution OR a crown  $I \subseteq V(G)$  such that

$$|V(G - N[I])| \leq 3k$$

Greedy find a maximal matching  $M_1$  of  $G$ .

**if**  $|M_1| > k$  **then**

    return  $\times$

**else**

    Let  $O$  (outsiders) be the set of vertices from  $G$  that are not in  $M_1$ .

    ...

    return  $I$

**end if**

---

Notation: Größe einer kleinstmöglichen Knotenüberdeckung von  $G$ :  $vc(G)$

### Zur Korrektheit der Kronenregel

Ist  $I$  Krone in  $G = (V, E)$  mit Kopf  $H = N(I)$ , so gilt:

$$vc(G) = vc(G - N[I]) + |H|.$$

Da  $I$  Krone, ist  $H$  kleinstmögliche Knotenüberdeckung von  $G[N[I]]$ .

Ist  $C^*$  kleinstmögliche Knotenüberdeckung (MVC) von  $G$ , so ist  $C^* \cap N[I]$  VC von  $G[N[I]]$  und  $C^* \setminus N[I]$  ist VC von  $G - N[I]$ .

$$\leadsto vc(G) = |C^*| = |C^* \setminus N[I]| + |C^* \cap N[I]| \geq vc(G') + |H|.$$

Ist  $C'$  MVC on  $G'$ , setze  $\tilde{C} = C' \cup H$ .  $\tilde{C}$  ist VC von  $G$ .

$$\leadsto vc(G) \leq |\tilde{C}| = vc(G') + |H|.$$

Bem.: "O.E."  $\tilde{C}$  statt  $C^*$ , also o.E.:  $H \subseteq C^*$ .

## **Bemerkungen zum Kronenalgorithmus** (Alternative ?!), falls keine $\times$ -Instanz:

O.E.: Graph  $G$  hat keine isolierten Knoten.

Maximales Matching  $M_1$  erfüllt  $|M_1| \leq k$ .

$O = V \setminus V(M_1)$  ist unabhängige Menge.

Gilt  $|O| \leq k$ , so haben wir Kern:  $|V| = 2|M_1| + |O| \leq 3k$ .

Maximum Matching  $M_2$  in bipartitem Graph  $B = (O, N(O), E_B)$  erfüllt  $|M_2| \leq k$ .

Gilt  $|M_2| = |N(O)|$ , so ist  $O$  Krone in  $G$ .

$\leadsto$  "Neustart" nach Anwendung der Kronenregel.

Im übrigen Fall:  $|M_2| \leq k$  und  $|M_2| < |N(O)|$ .

$I = O \setminus V(M_2)$  ist unabhängige Menge, nicht-leer wegen  $|O| > k$ .

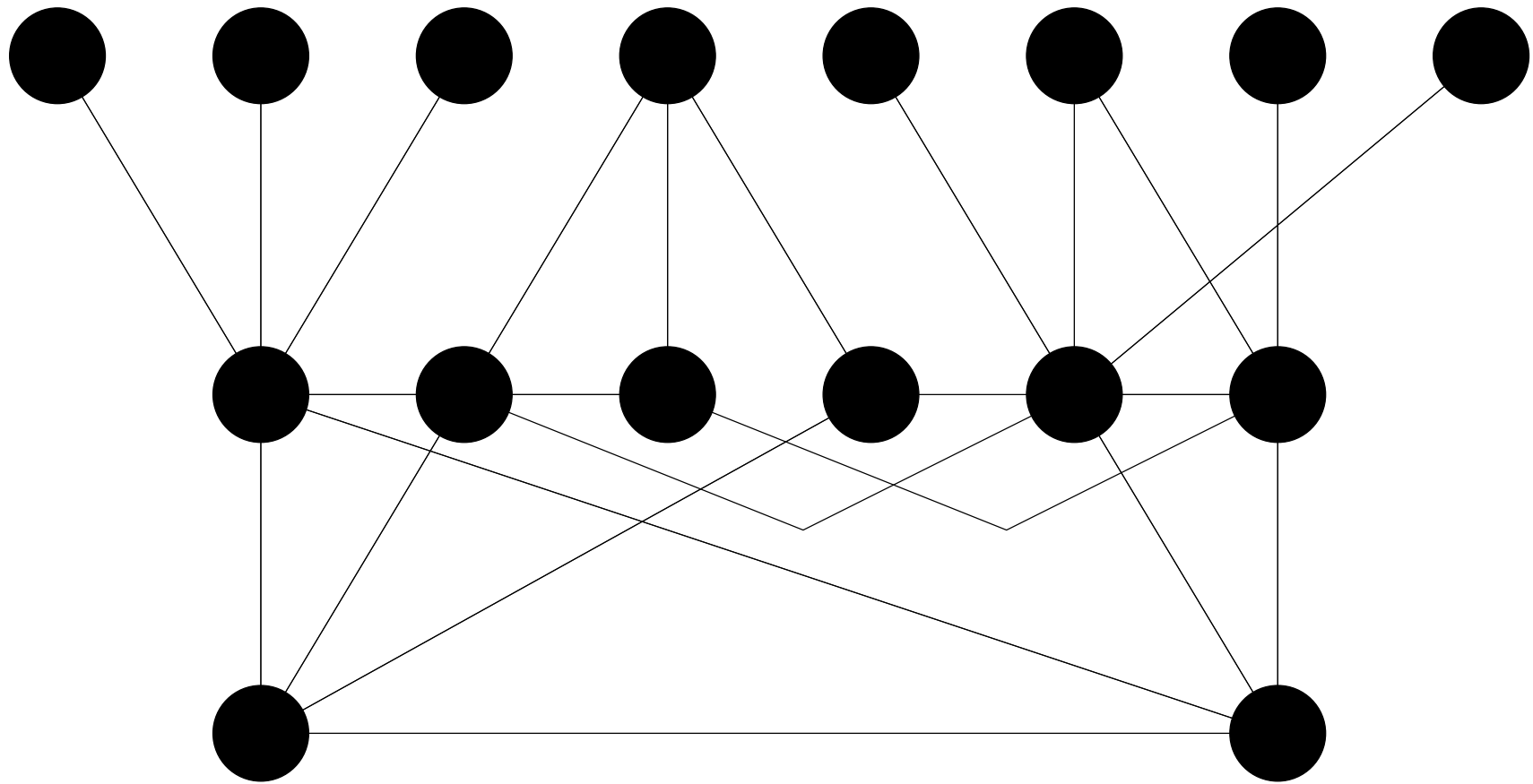
Wähle  $x \in I$  beliebig und konstruiere von  $x$  startenden  $M_2$ -alternierenden Pfad  $P$ ;

$P$  induziert in  $B$  (somit in  $G$ ) eine Krone.

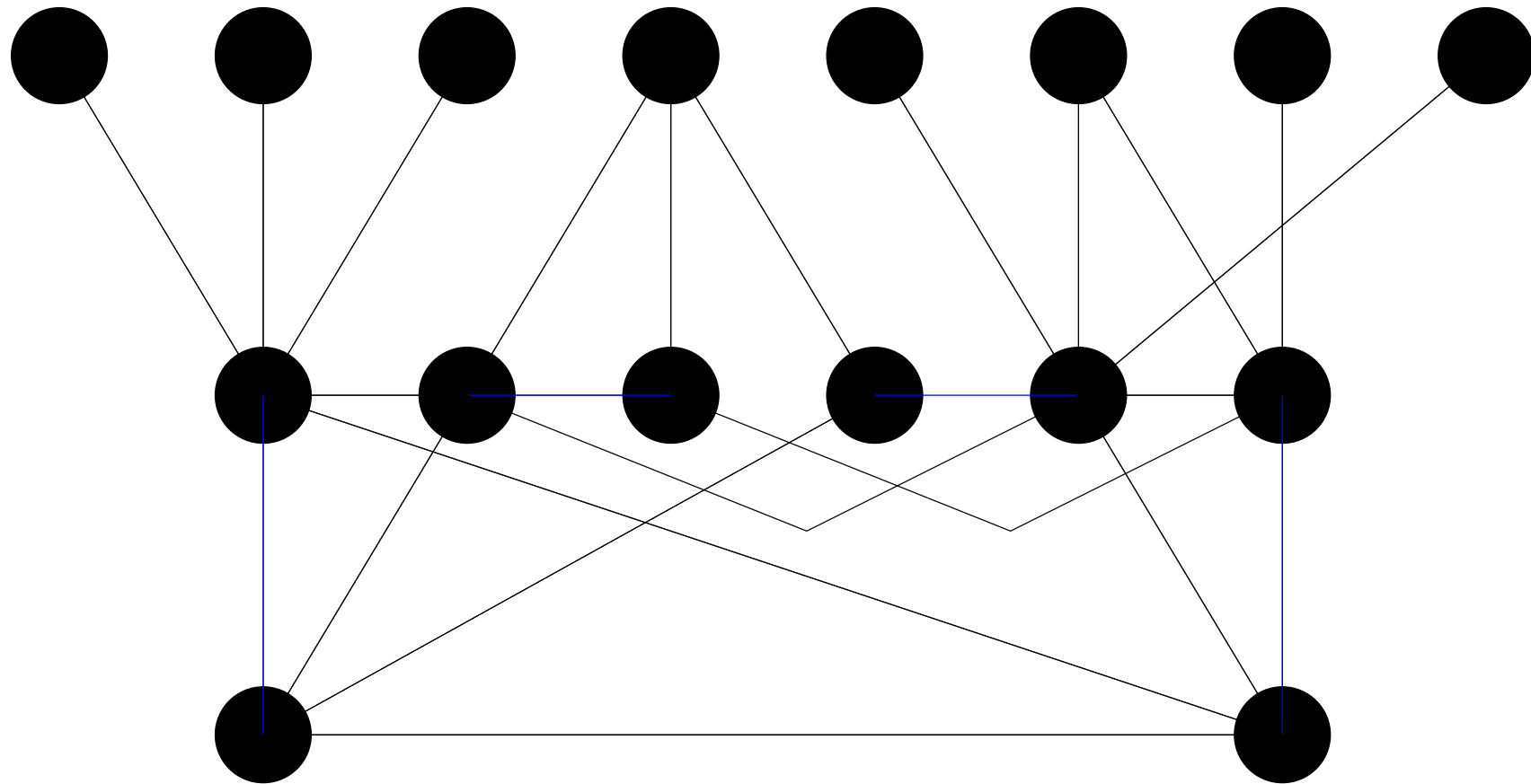
$\leadsto$  "Neustart" nach Anwendung der Kronenregel.



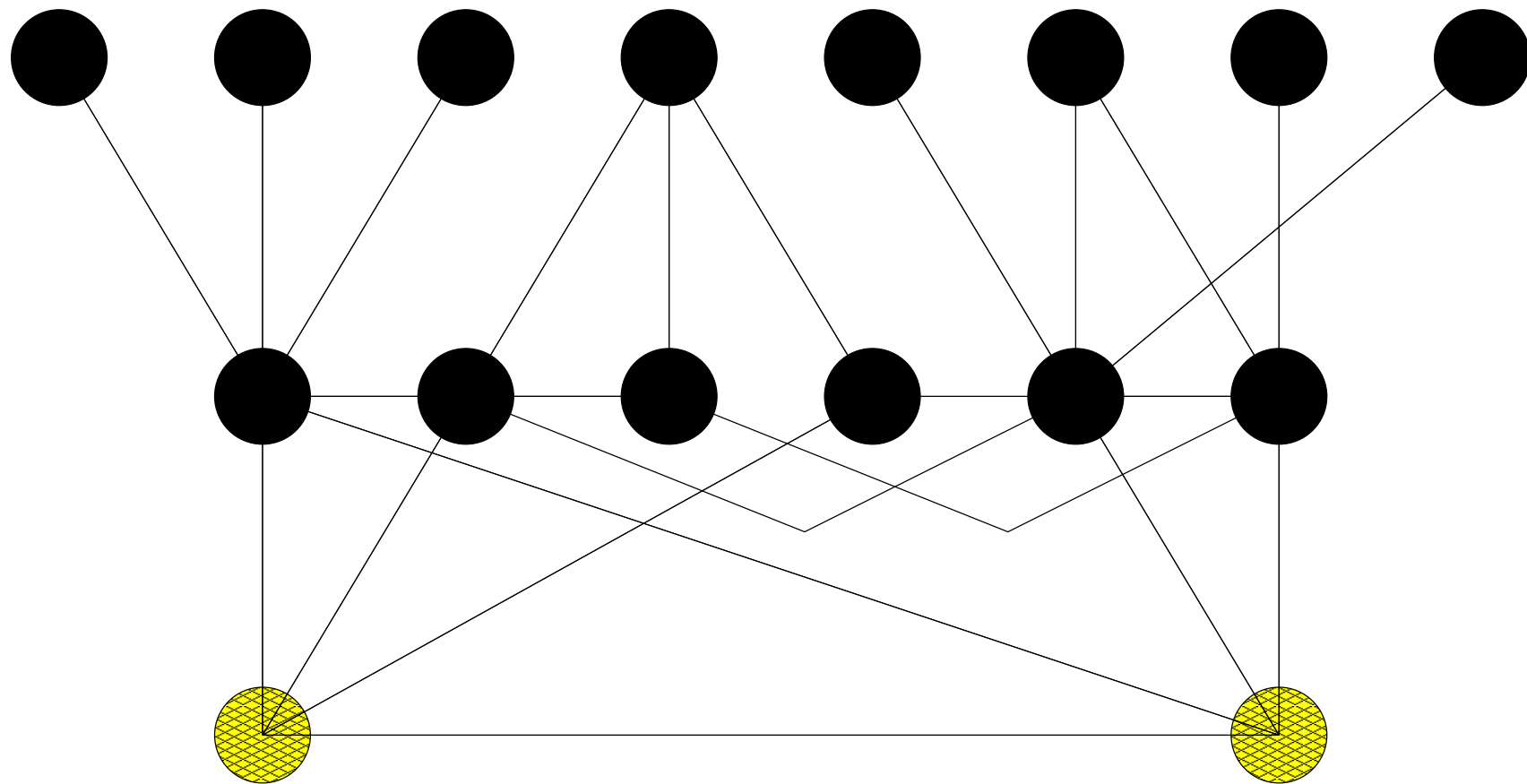
## Die Kronenregel an einem Beispiel



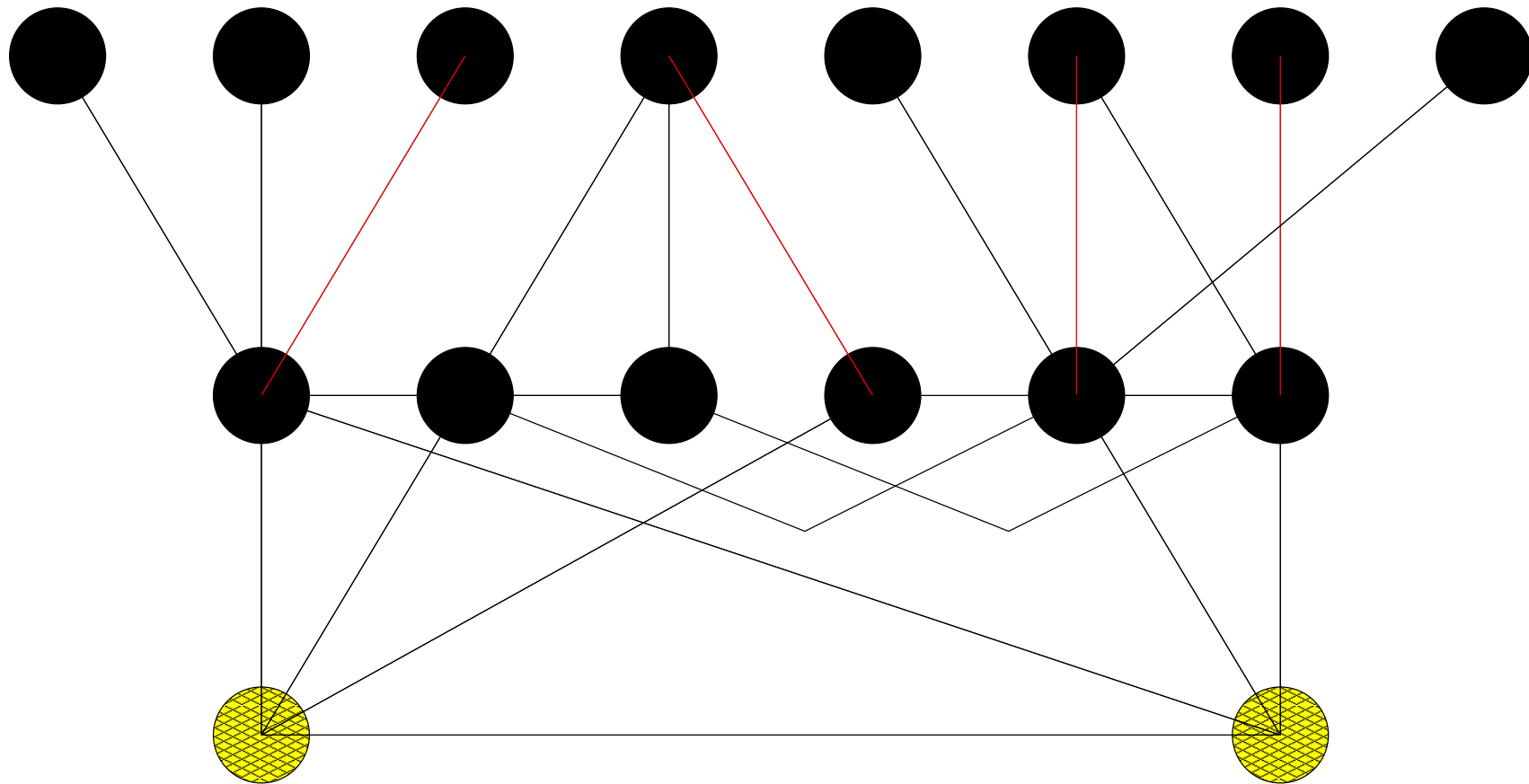
**Die Kronenregel an einem Beispiel:** Ein maximales Matching  $M_1$



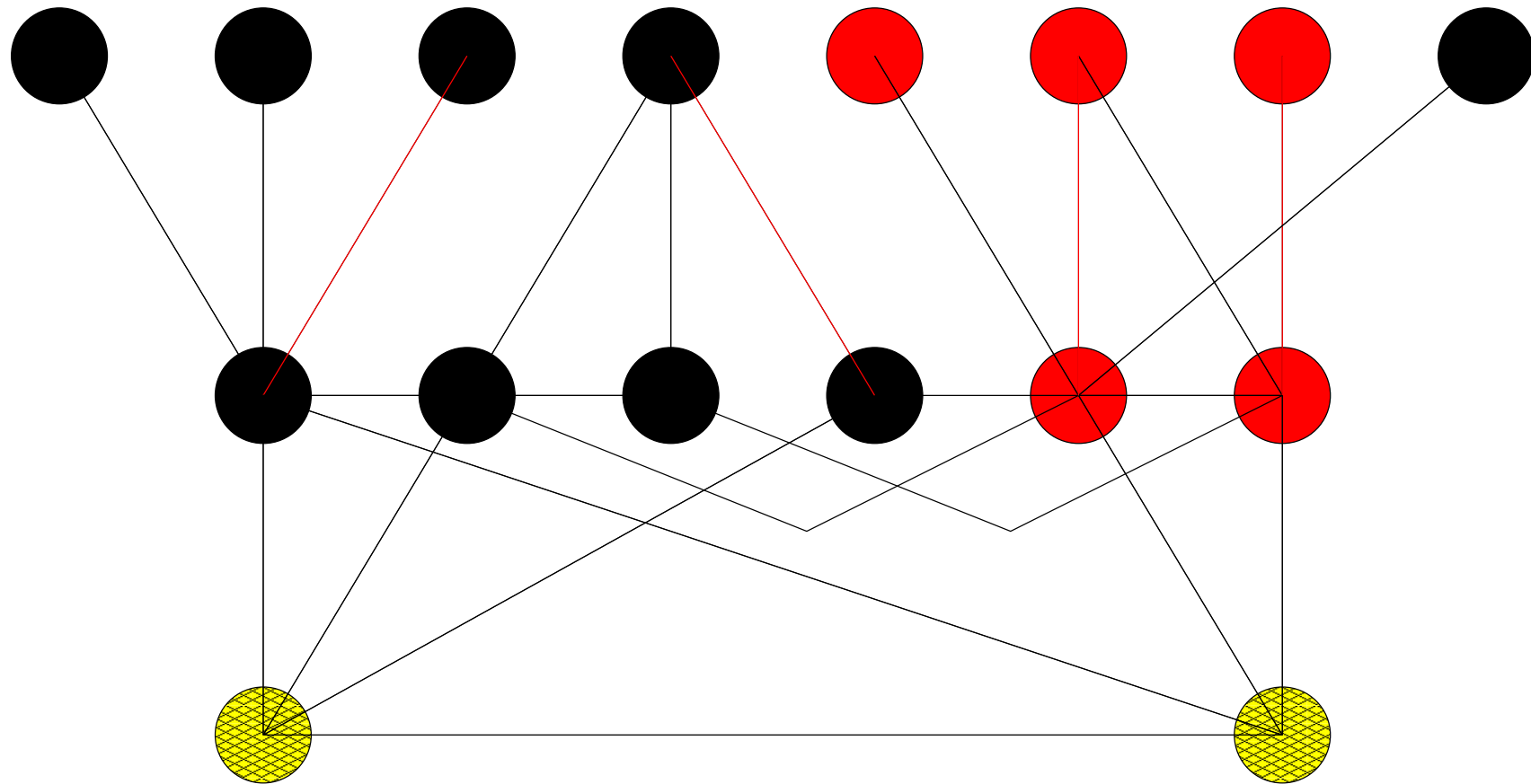
**Die Kronenregel an einem Beispiel:** Der übrige Graph ist nicht gelb



**Die Kronenregel an einem Beispiel:** Ein rotes Maximum Matching

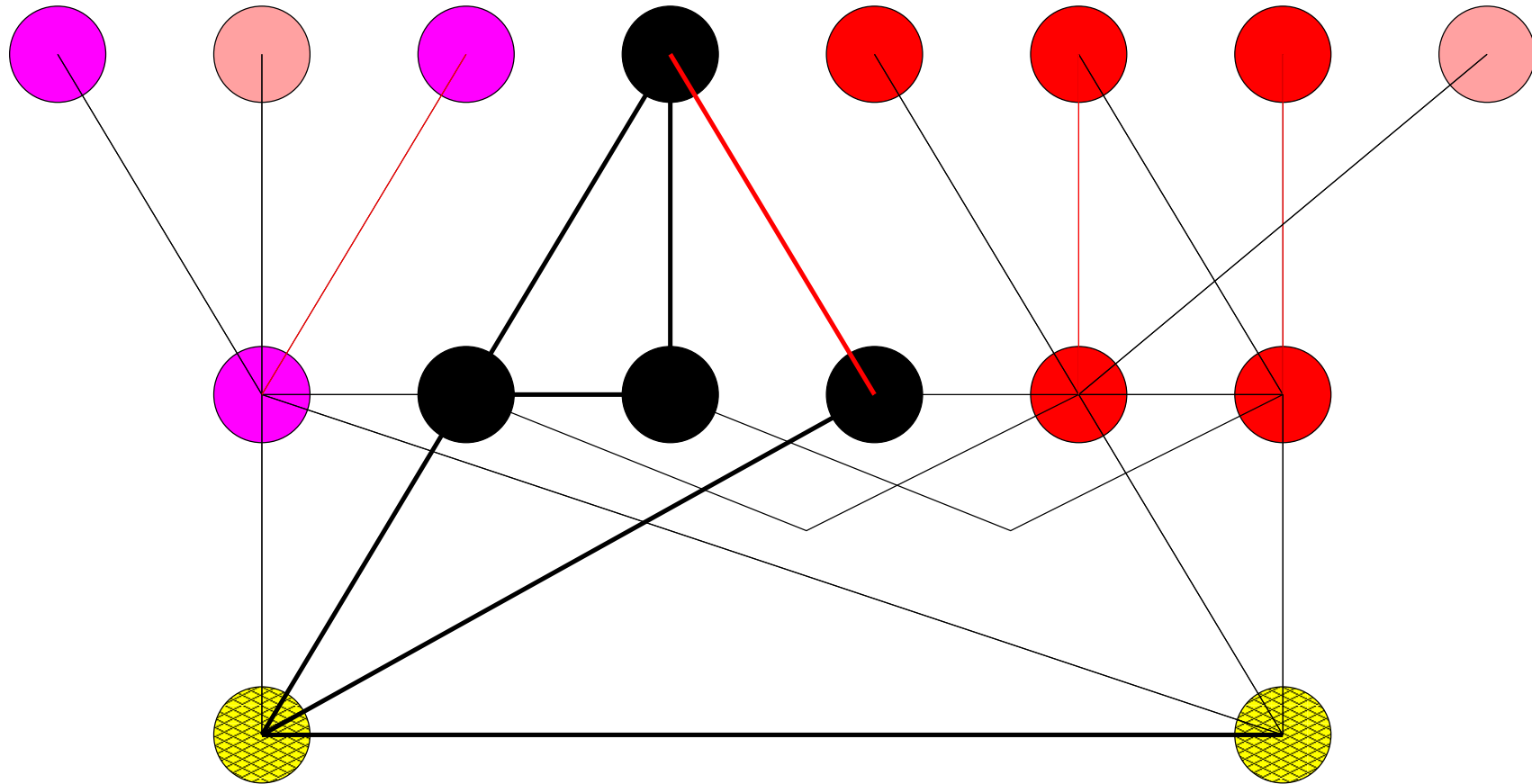


**Die Kronenregel an einem Beispiel:** Die roten Knoten sind eine Krone





**Die Kronenregel an einem Beispiel:** Die rosa Knoten werden entfernt (isoliert)



Kern: fette Kanten; Achtung: Regeln werden nochmals ausgelöst

