

5 Komplexität

Algorithmen und Datenstrukturen 2 Sommer 2024

Prof. Dr. Sándor Fekete

5.3 Ein Beispiel mit Logik



Beispiel 5.5: Knapsack

Beispiel 5.5.

Wir betrachten die folgende Knapsack-Instanz mit $n=12, z_i=p_i, Z=111444$ und den folgenden Objekten:

Gibt es eine Menge $S \subseteq \{1,...12\}$ mit $\sum_{i \in S} z_i = \sum_{i \in S} p_i = Z$?

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	10101
$z_4 = p_4 =$	10010
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	1110
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1



1. Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	10101
$z_4 = p_4 =$	10010
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	1110
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1

- 1. Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	10101
$z_4 = p_4 =$	10010
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	1110
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1



- Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.
- 3. Ziffer: Man muss 5 oder 6 auswählen, aber nicht beide.

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	1 <mark>0</mark> 101
$z_4 = p_4 =$	10010
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	1110
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1



- Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.
- 3. Ziffer: Man muss 5 oder 6 auswählen, aber nicht beide.
- 4. Ziffer: Man muss 1, 3 oder 6 auswählen, dann kann man mit 7 und 8 den Wert 4 erzeugen.

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	101)1
$z_4 = p_4 =$	10 <mark>0</mark> 10
$z_5 = p_5 =$	1 <mark>0</mark> 01
$z_6 = p_6 =$	1 <mark>11</mark> 10
$z_7 = p_7 =$	200
$z_8 = p_8 =$	1 00
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1



- Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.
- 3. Ziffer: Man muss 5 oder 6 auswählen, aber nicht beide.
- 4. Ziffer: Man muss 1, 3 oder 6 auswählen, dann kann man mit 7 und 8 den Wert 4 erzeugen.
- 5. Ziffer: Man muss 1, 4 oder 6 auswählen, dann kann man mit 9 und 10 den Wert 4 erzeugen.

$\gamma_1 - \gamma_1 -$	100110
$z_1 = p_1 =$	
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	10101
$z_4 = p_4 =$	1001)
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	111)
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1



- Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.
- 3. Ziffer: Man muss 5 oder 6 auswählen, aber nicht beide.
- 4. Ziffer: Man muss 1, 3 oder 6 auswählen, dann kann man mit 7 und 8 den Wert 4 erzeugen.
- 5. Ziffer: Man muss 1, 4 oder 6 auswählen, dann kann man mit 9 und 10 den Wert 4 erzeugen.
- **6. Ziffer:** Man muss 2, 3 oder 5 auswählen, dann kann man mit 11 und 12 den Wert 4 erzeugen.

$z_1 = p_1 =$	10011	0
$z_2 = p_2 =$	10000	1
$z_3 = p_3 =$	1010	1
$z_4 = p_4 =$	1001	O
$z_5 = p_5 =$	100	1
$z_6 = p_6 =$	111	O
$z_7 = p_7 =$	20	0
$z_8 = p_8 =$	10	O
$z_9 = p_9 =$	2	O
$z_{10} = p_{10} =$	1	O
$z_{11} = p_{11} =$		2
$z_{12} = p_{12} =$		1



- Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.
- 3. Ziffer: Man muss 5 oder 6 auswählen, aber nicht beide.
- 4. Ziffer: Man muss 1, 3 oder 6 auswählen, dann kann man mit 7 und 8 den Wert 4 erzeugen.
- 5. Ziffer: Man muss 1, 4 oder 6 auswählen, dann kann man mit 9 und 10 den Wert 4 erzeugen.
- **6. Ziffer:** Man muss 2, 3 oder 5 auswählen, dann kann man mit 11 und 12 den Wert 4 erzeugen.

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	10101
$z_4 = p_4 =$	10010
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	1110
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$p_{10} = p_{10} = 0$	10
$p_{11} = p_{11} = 0$	2
$p_{12} = p_{12} = 0$	1



$$x_1 \vee \overline{x_1}$$

$$x_2 \vee \overline{x_2}$$

$$x_3 \vee \overline{x_3}$$

$$(x_1 \lor x_2 \lor \overline{x_3})$$

$$(x_1 \vee \overline{x_2} \vee \overline{x_3})$$

$$(\overline{x_1} \lor x_2 \lor x_3)$$

- Ziffer: Man muss 1 oder 2 auswählen, aber nicht beide.
- 2. Ziffer: Man muss 3 oder 4 auswählen, aber nicht beide.
- 3. Ziffer: Man muss 5 oder 6 auswählen, aber nicht beide.
- 4. Ziffer: Man muss 1, 3 oder 6 auswählen, dann kann man mit 7 und 8 den Wert 4 erzeugen.
- 5. Ziffer: Man muss 1, 4 oder 6 auswählen, dann kann man mit 9 und 10 den Wert 4 erzeugen.
- **6. Ziffer:** Man muss 2, 3 oder 5 auswählen, dann kann man mit 11 und 12 den Wert 4 erzeugen.

$z_1 = p_1 =$	100110
$z_2 = p_2 =$	100001
$z_3 = p_3 =$	10101
$z_4 = p_4 =$	10010
$z_5 = p_5 =$	1001
$z_6 = p_6 =$	1110
$z_7 = p_7 =$	200
$z_8 = p_8 =$	100
$z_9 = p_9 =$	20
$z_{10} = p_{10} =$	10
$z_{11} = p_{11} =$	2
$z_{12} = p_{12} =$	1

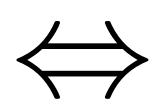
Beispiel 5.5: Äquivalenz

$$x_1 \vee \overline{x_1}$$

$$x_2 \vee \overline{x_2}$$

$$x_3 \vee \overline{x_3}$$

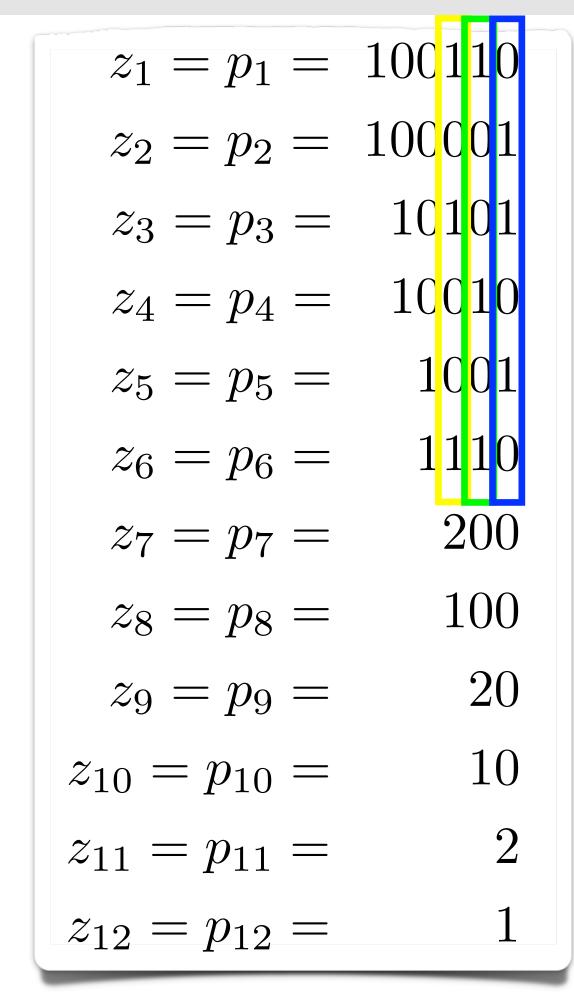
$$(x_1 \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor x_2 \lor x_3)$$



Konkret: Jede Lösung der logischen Formel entspricht einer Lösung der Instanz Subset Sum — und umgekehrt.

Allgemein: Für jede logische Formel dieser Art lässt sich schnell eine äquivalente Instanz von Subset Sum konstruieren.

Also: Wenn wir einen "perfekten" Algorithmus für Subset Sum haben, dann können wir auch schnell entscheiden, ob eine logische Formel lösbar ist.





5.4 3SAT



Das Logikproblem 3SAT

Definition 5.6. (3-Satisfiability (3SAT))

Gegeben: Eine Boolesche Formel, zusammengesetzt aus:

- n Boolesche Variablen $x_1, ..., x_n$, aus denen wir Literale ℓ_i der Form x_k oder \bar{x}_k bilden können.
- m Klauseln, jede zusammengesetzt aus genau drei Literalen $C_j = (\ell_{j,1} \vee \ell_{j,2} \vee \ell_{j,3})$ mit $1 \leq j \leq m$.

Gesucht: Eine alle m Klauseln erfüllende (engl: satisfying) Wahrheitsbelegung der n Variablen.

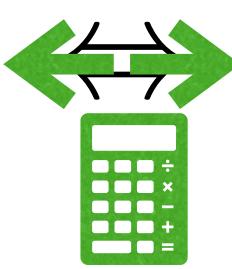
$$(x_1 \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor x_2 \lor x_3)$$



Wie schwer sind 3SAT und KNAPSACK?

3SAT

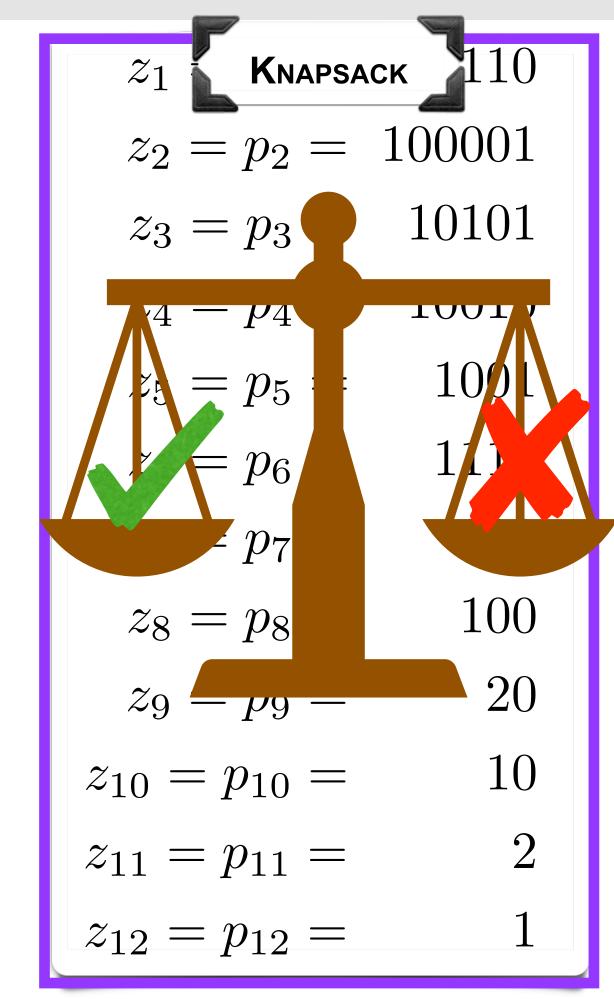
$$(x_1 \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor x_2 \lor x_3)$$



Satz 5.9. Wenn 0-1-Knapsack $\in P$ ist, dann ist auch $3SAT \in P$.

Satz 5.12 (Satz von Cook 1971). Wenn $3SAT \in P$, dann gilt P = NP.

Korollar 5.11. Wenn $Knapsack \in P$, $dann \ gilt \ P = NP$.





5.5 Konsequenzen



Algorithmen und Investmentbanker

Ziele: Ein "perfekter" Algorithmus liefert

1. immer

2. schnell

3. eine optimale Lösung.

Ziele: Ein "perfekter" Finanzberater ist ein

1. ehrlicher

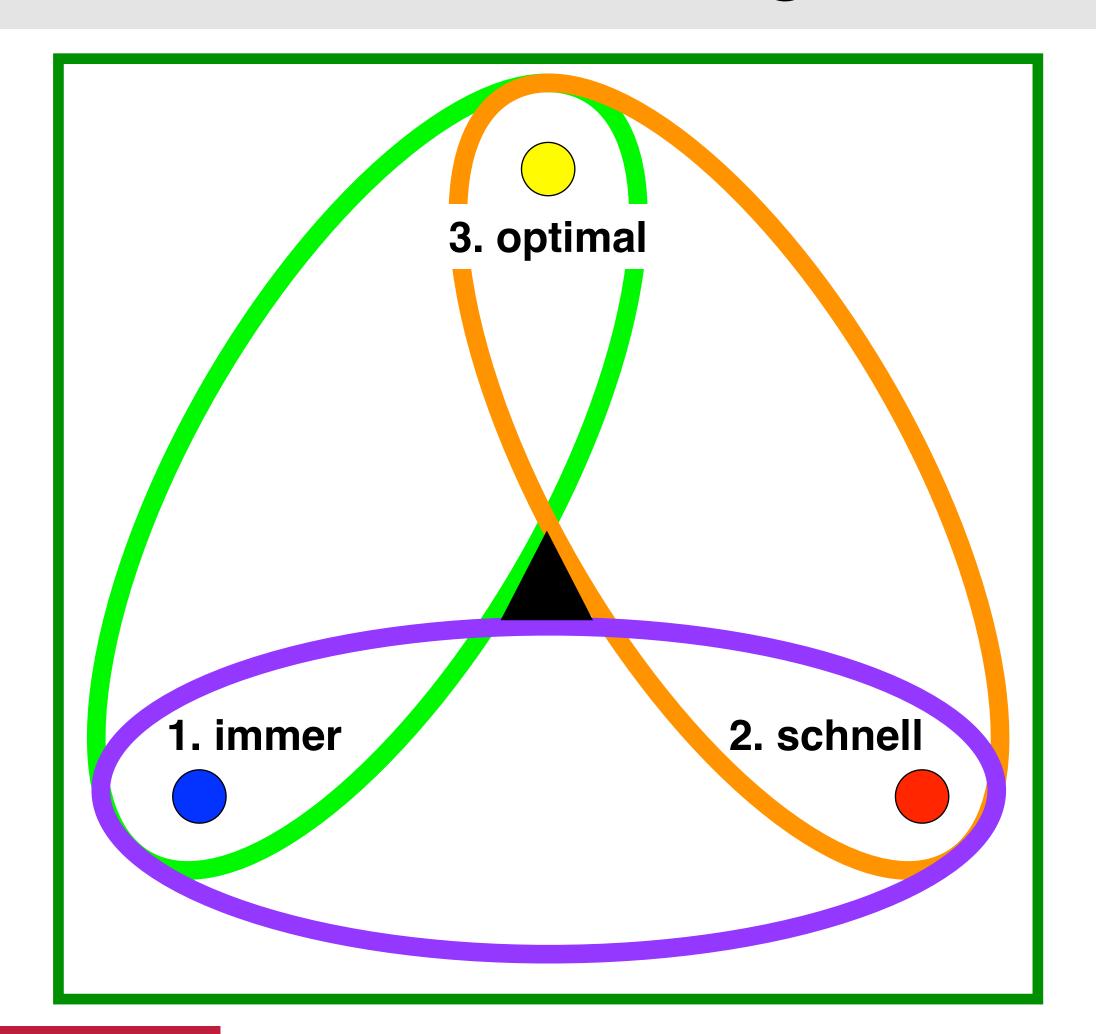
2. intelligenter

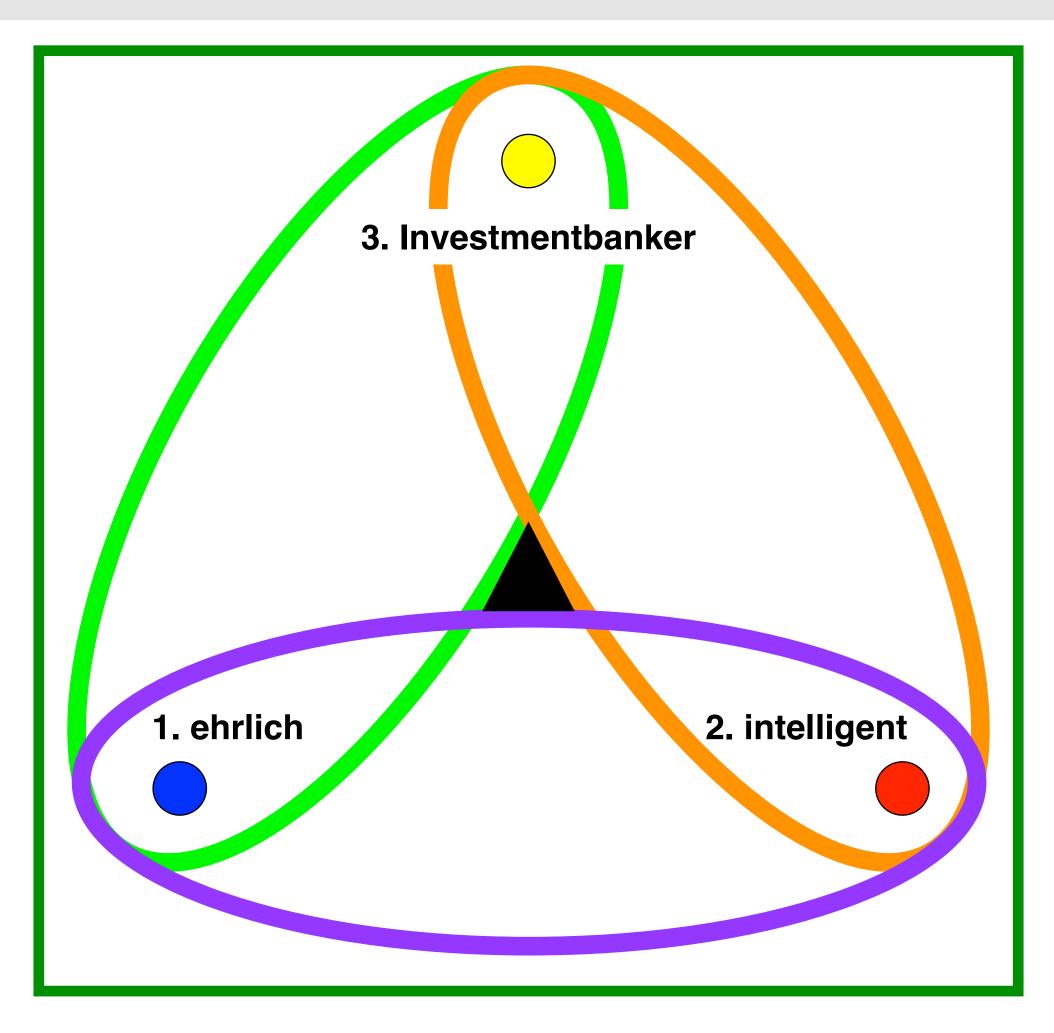
3. Investmentbanker.



NP-Vollständigkeit

Finanzkrise



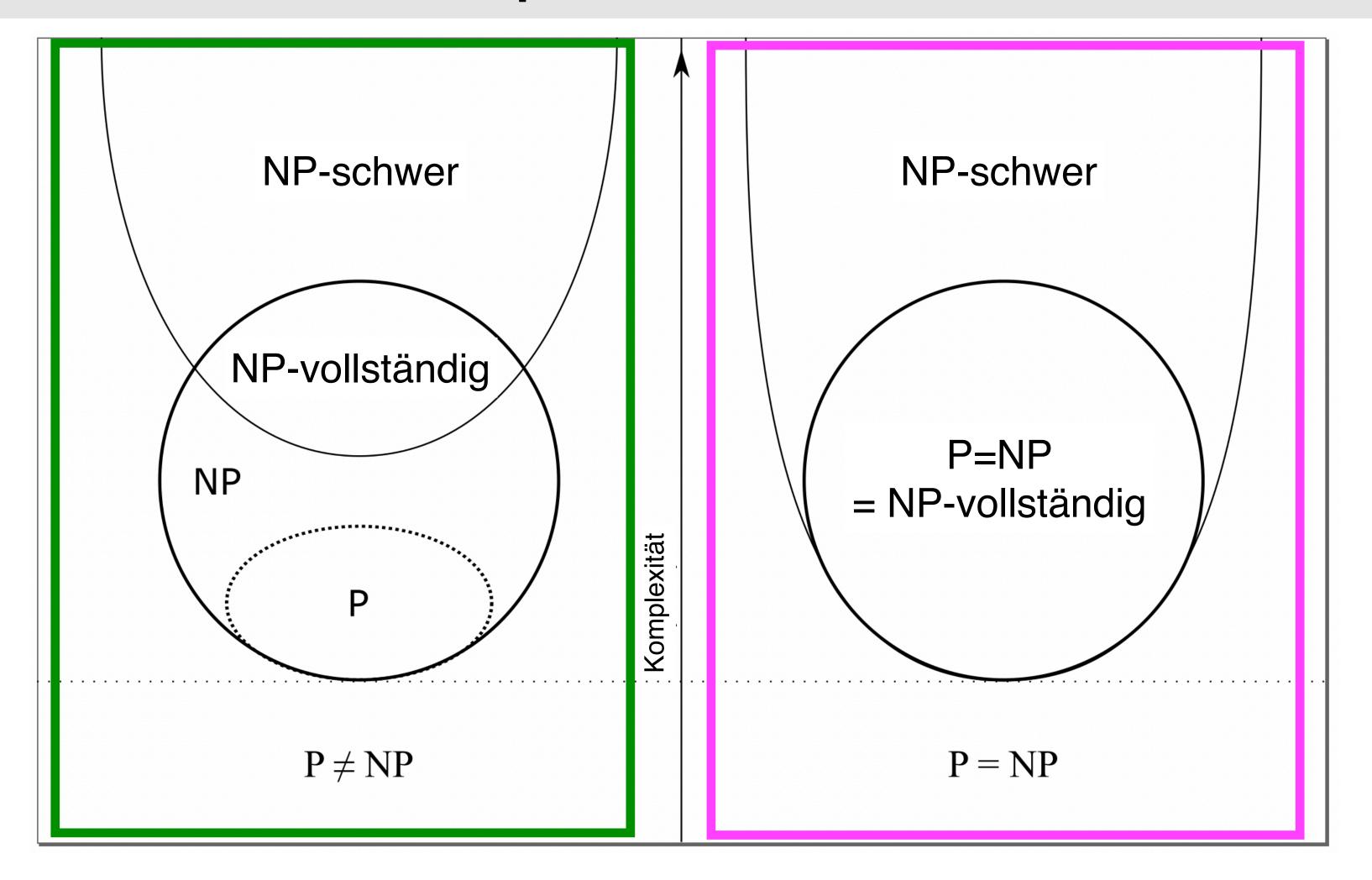






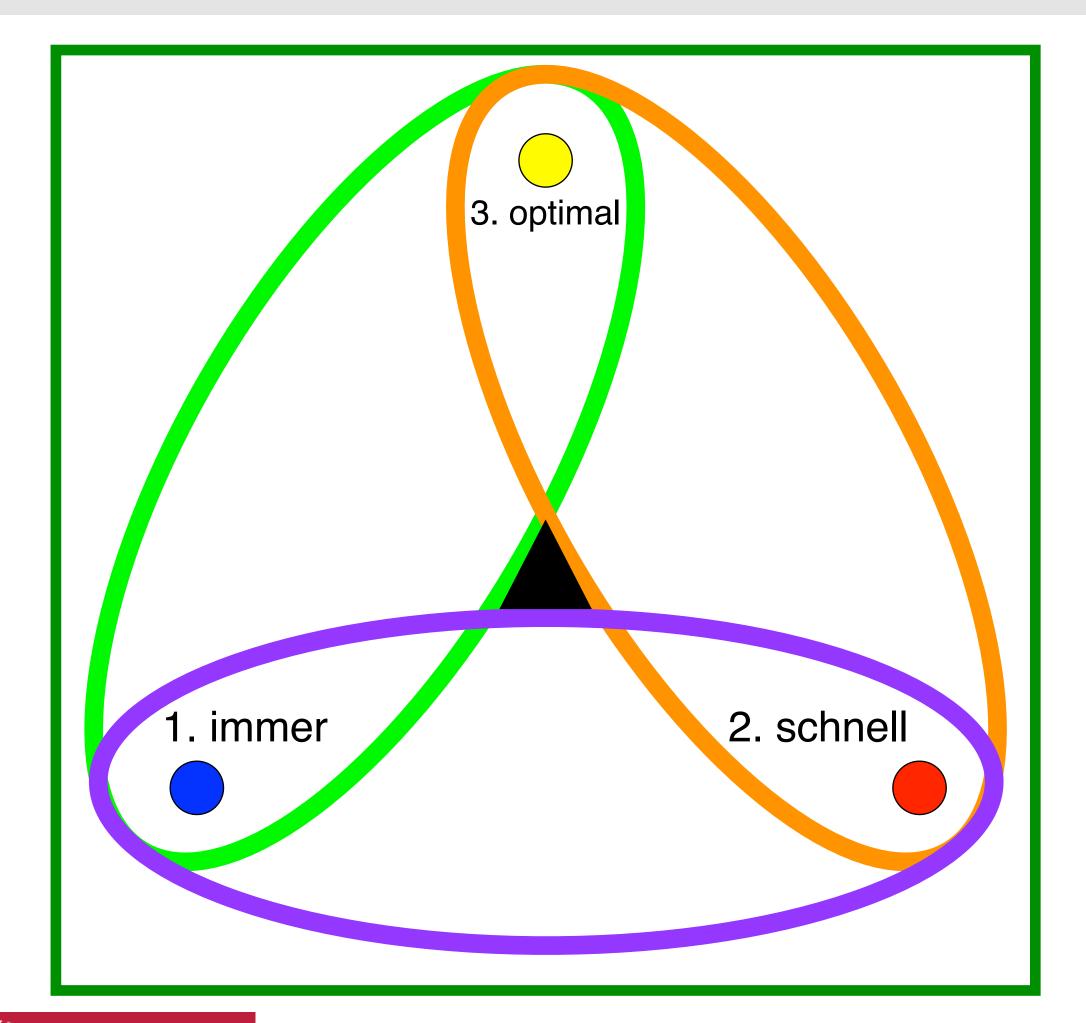


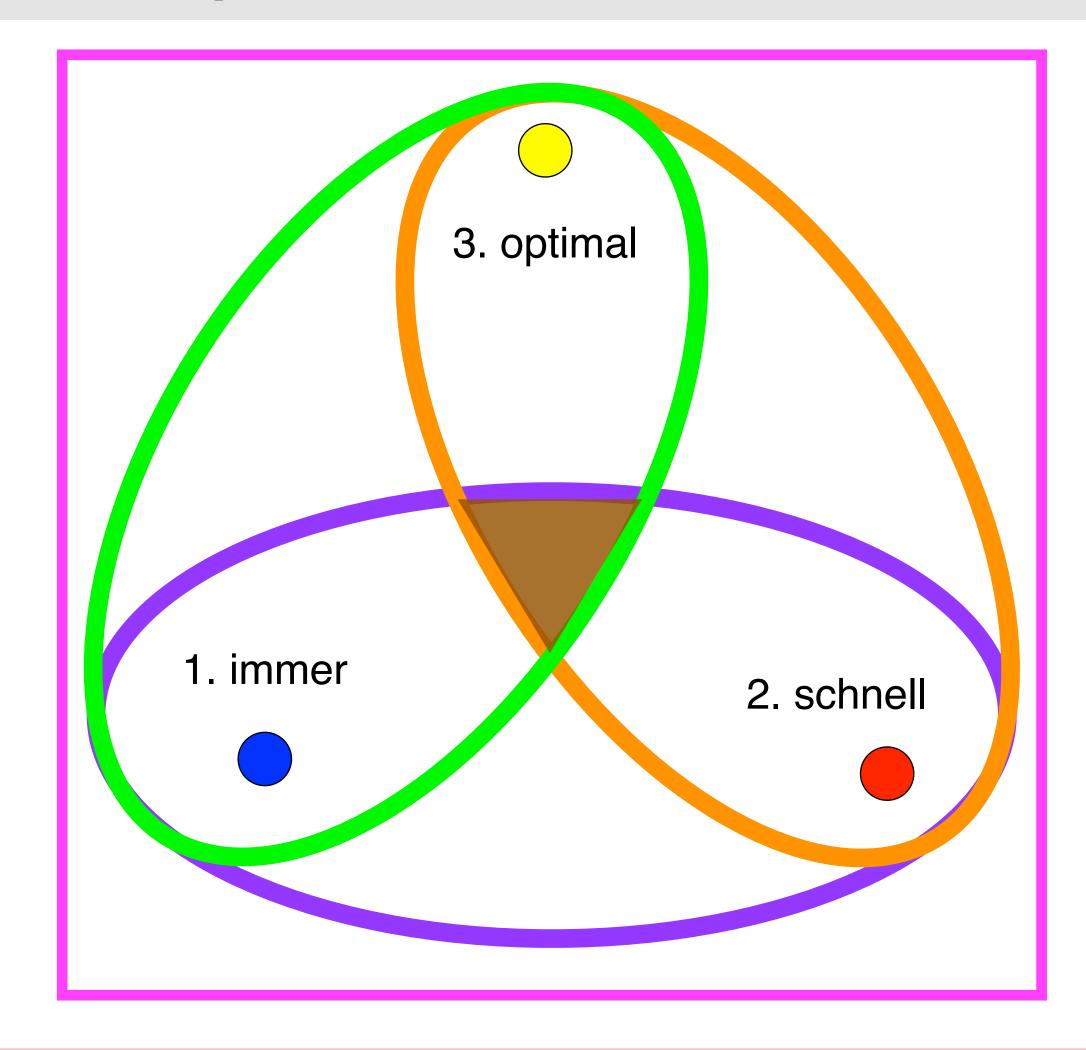
Die Komplexitätslandschaft





Das Bild für Knapsack







Ziel: Algorithmus für NP-schweres Problem

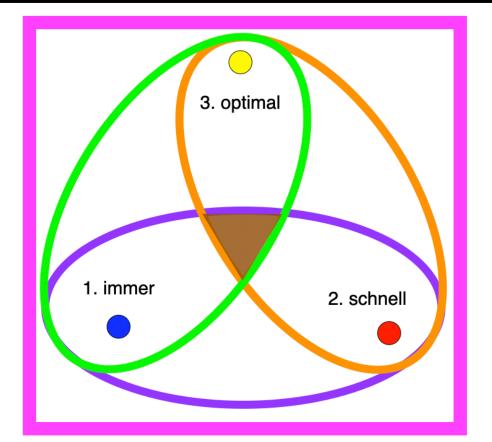
1. immer

2. schnell

3. eine optimale Lösung.

Persönlichkeitsprofile:

- (A) Mit dem Schicksal hadern und diskutieren
- (B) Auf Glück vertrauen
- (C) Hart arbeiten
- (D) Erwartungen zurückschrauben



(A) Komplexitätsanalyse: Spezialfall nicht NP-schwer?

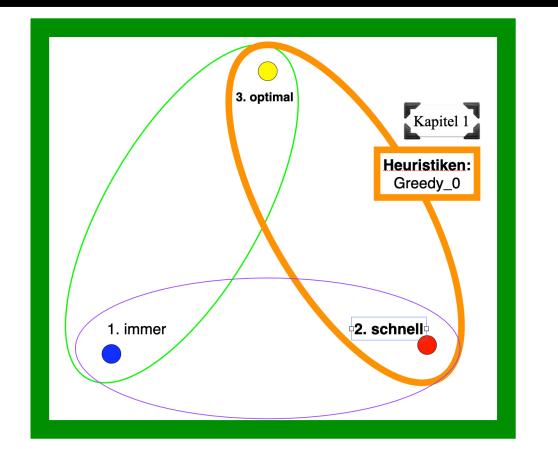


Ziel: Algorithmus für NP-schweres Problem

1. immer

2. schnell

3. eine optimale Lösung.





- (A) Mit dem Schicksal hadern und diskutieren
- (B) Auf Glück vertrauen
- (C) Hart arbeiten
- (D) Erwartungen zurückschrauben
- (A) Komplexitätsanalyse: Spezialfall nicht NP-schwer?
- (B) Heuristiken: raten und hoffen

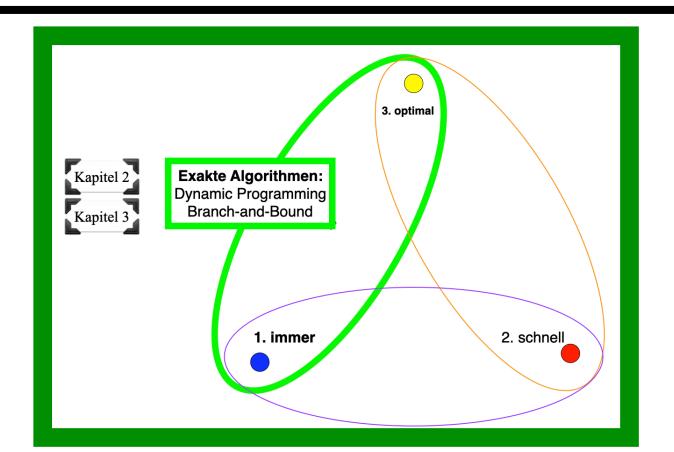


Ziel: Algorithmus für NP-schweres Problem

1. immer

2. schnell

3. eine optimale Lösung.





- (A) Mit dem Schicksal hadern und diskutieren
- (B) Auf Glück vertrauen
- (C) Hart arbeiten
- (D) Erwartungen zurückschrauben
- (A) Komplexitätsanalyse: Spezialfall nicht NP-schwer?
- (B) Heuristiken: raten und hoffen
- (C) Exakte Algorithmen

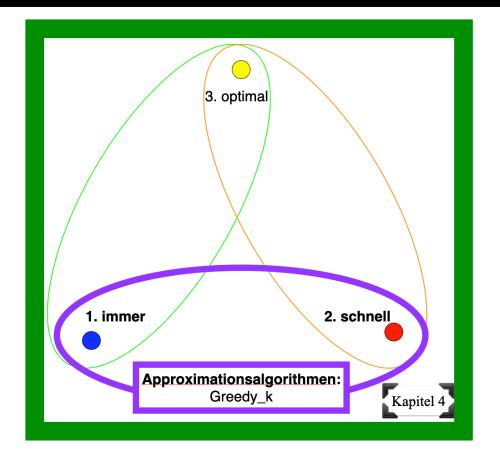


Ziel: Algorithmus für NP-schweres Problem

1. immer

2. schnell

3. eine optimale Lösung.





(B) Auf Glück vertrauen

(C) Hart arbeiten

(D) Erwartungen zurückschrauben

(A) Komplexitätsanalyse: Spezialfall nicht NP-schwer?

(B) Heuristiken: raten und hoffen

(C) Exakte Algorithmen

(D) Approximationsalgorithmen: "gut" statt "optimal"

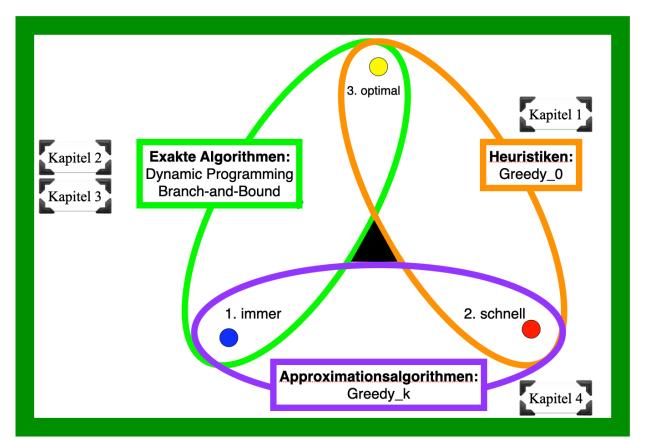


Ziel: Algorithmus für NP-schweres Problem

1. immer

2. schnell

3. eine optimale Lösung.







- (A) Mit dem Schicksal hadern und diskutieren
- (B) Auf Glück vertrauen
- (C) Hart arbeiten
- (D) Erwartungen zurückschrauben
- (A) Komplexitätsanalyse: Spezialfall nicht NP-schwer?
- (B) Heuristiken: raten und hoffen
- (C) Exakte Algorithmen
- (D) Approximationsalgorithmen: "gut" statt "optimal"

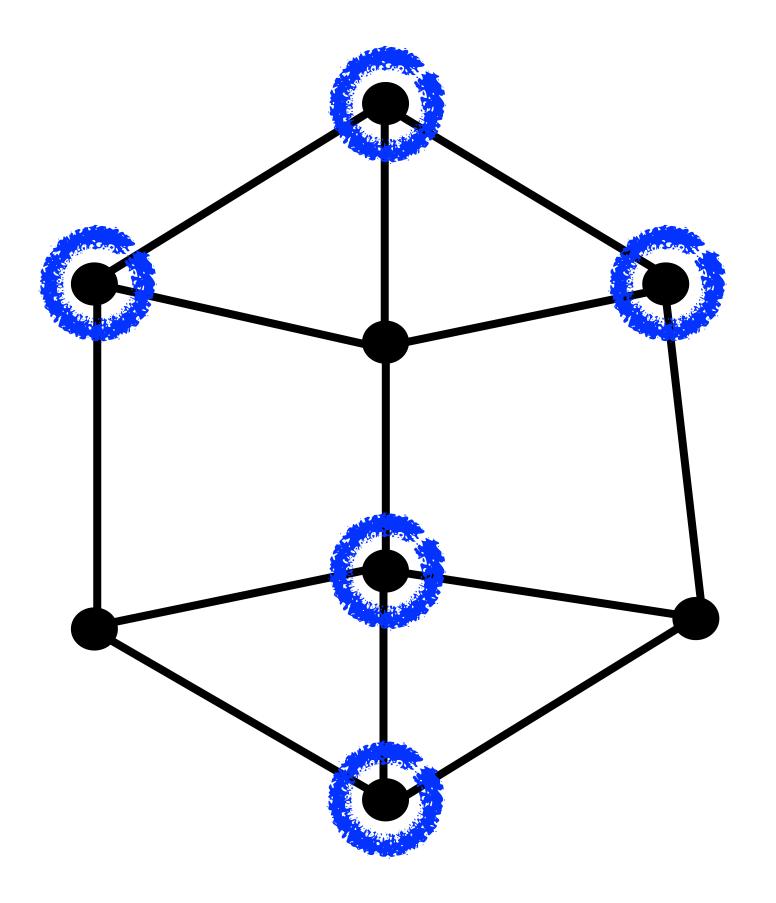
5.6 Ausblick



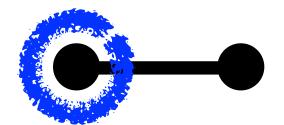
Vertex Cover

Gegeben: Ein Graph G=(V,E)

Gesucht: Ein kleinstmögliches Vertex Cover S in G: eine möglichst kleine Menge von Knoten, die alle Kanten überdecken

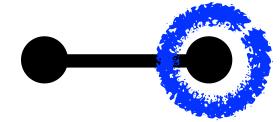






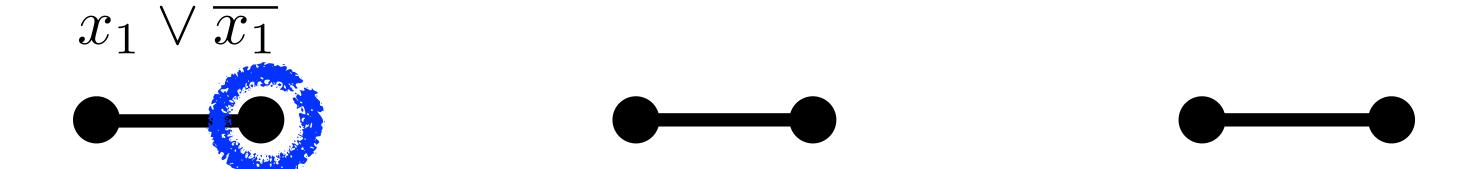


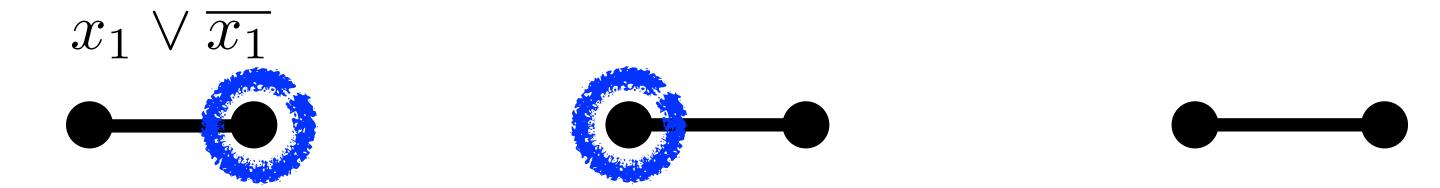


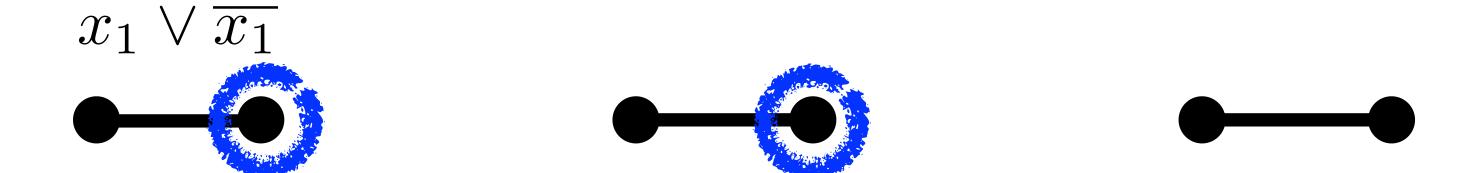




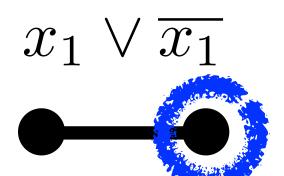


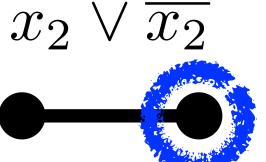




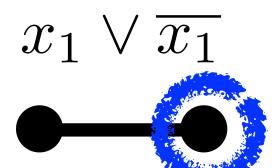


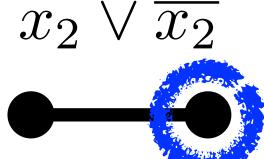


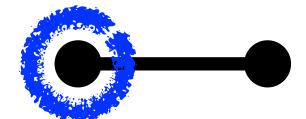


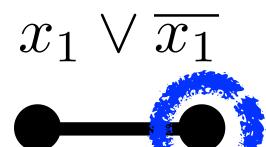


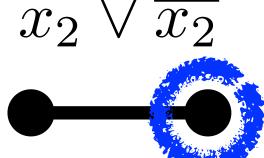


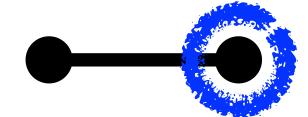


















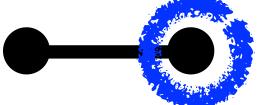
$$x_2 \vee \overline{x_2}$$



$$x_3 \vee \overline{x_3}$$











$$x_3 \vee \overline{x_3}$$



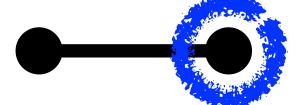


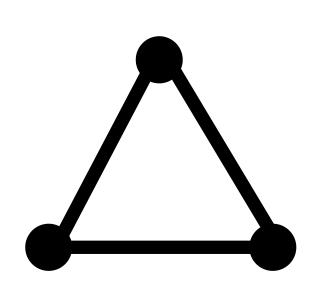






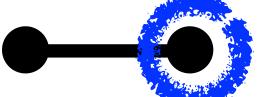
$$x_3 \vee \overline{x_3}$$









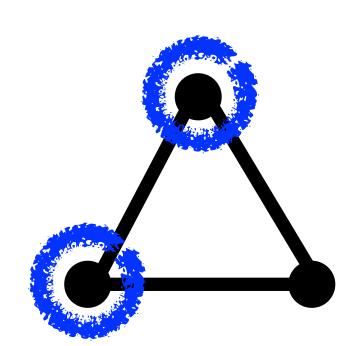


$$x_2 \vee \overline{x_2}$$



$$x_3 \vee \overline{x_3}$$







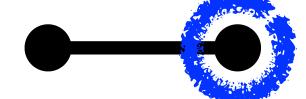


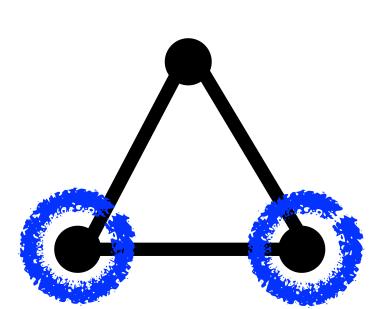


$$x_2 \vee \overline{x_2}$$



$$x_3 \vee \overline{x_3}$$









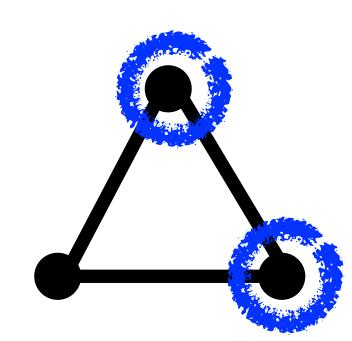






$$x_3 \vee \overline{x_3}$$









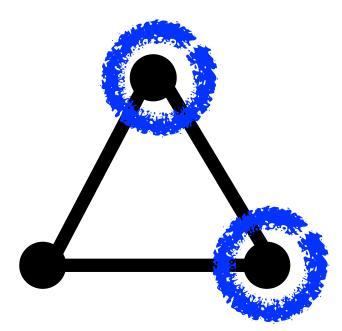


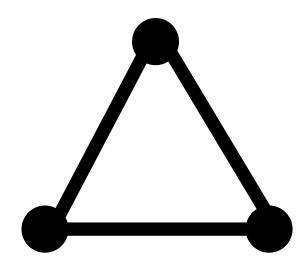




$$x_3 \vee \overline{x_3}$$









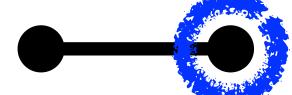


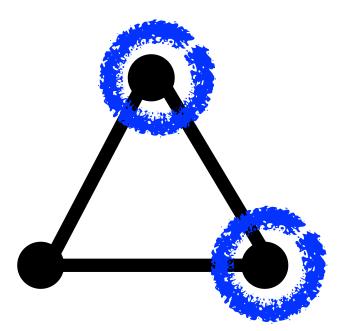


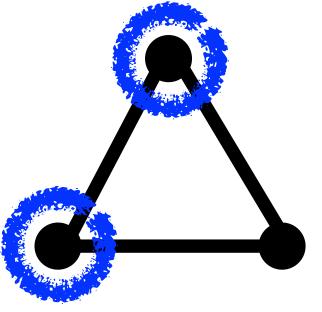




$$x_3 \vee \overline{x_3}$$











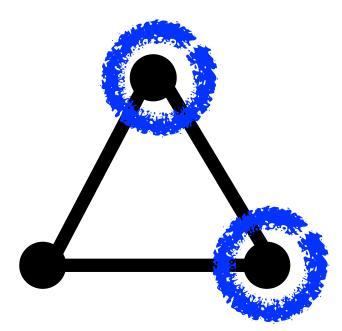


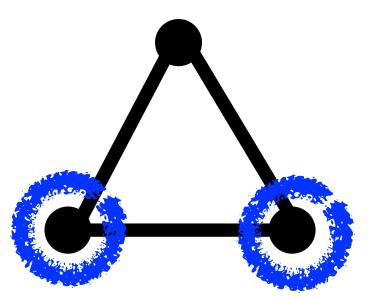
$$x_2 \vee \overline{x_2}$$



$$x_3 \vee \overline{x_3}$$









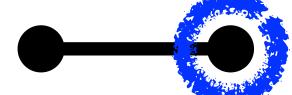


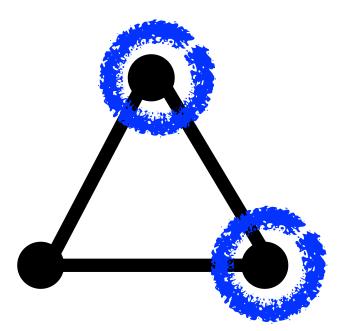


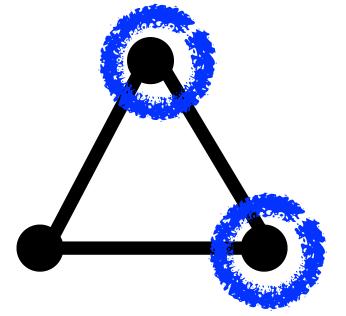
$$x_2 \vee \overline{x_2}$$



$$x_3 \vee \overline{x_3}$$









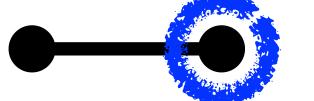


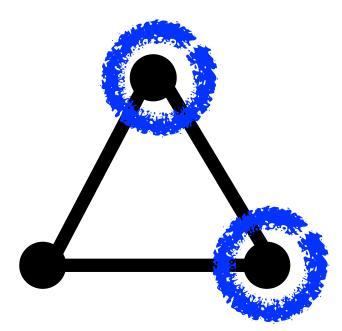


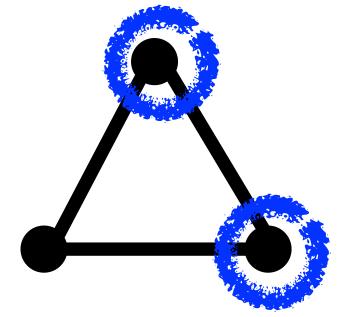
$$x_2 \vee \overline{x_2}$$

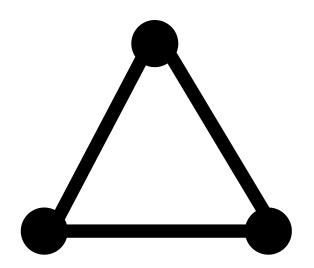


$$x_3 \vee \overline{x_3}$$









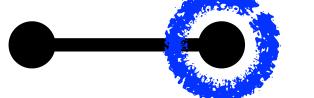


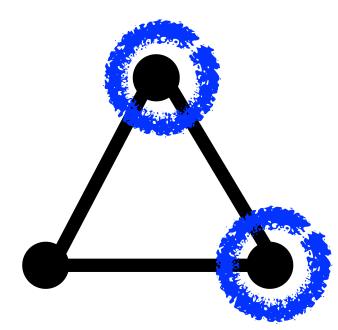


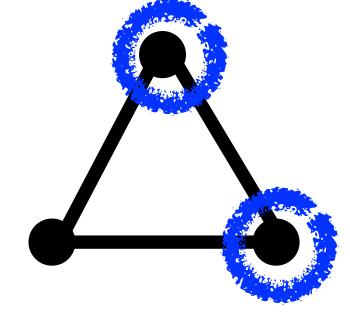


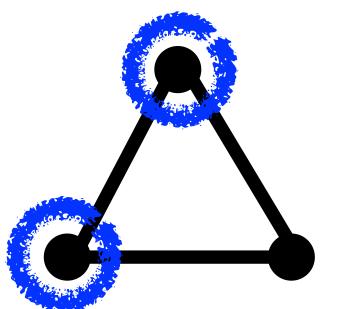


$$x_3 \vee \overline{x_3}$$











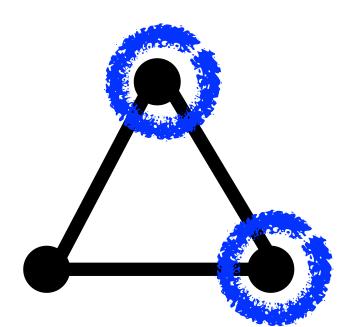


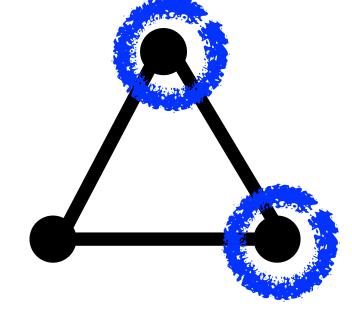
$$x_2 \vee \overline{x_2}$$

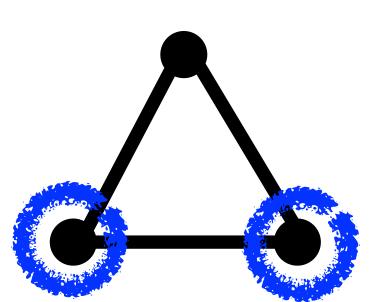


$$x_3 \vee \overline{x_3}$$













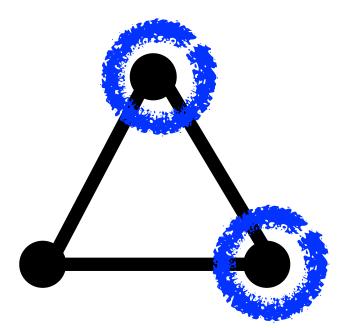


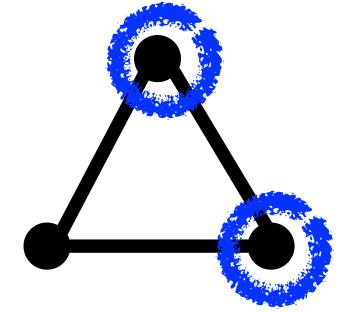


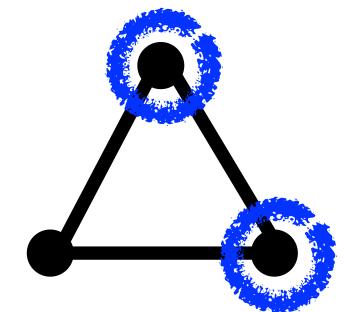


$$x_3 \vee \overline{x_3}$$













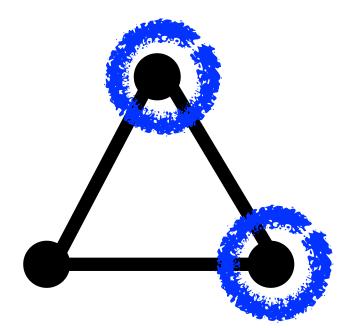


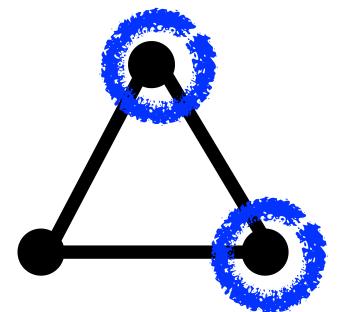
$$x_2 \vee \overline{x_2}$$

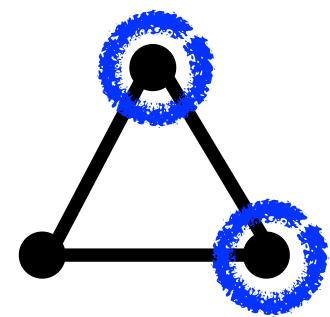


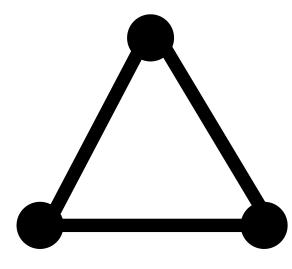
$$x_3 \vee \overline{x_3}$$













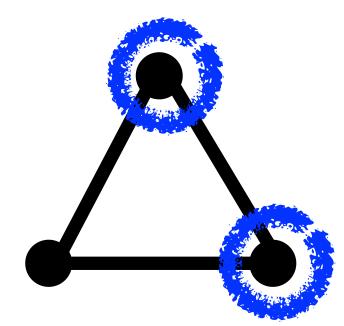


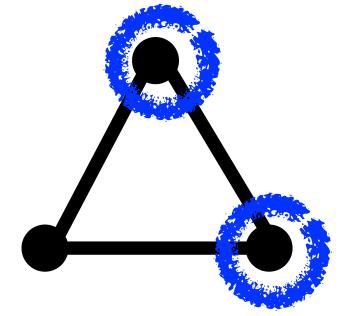


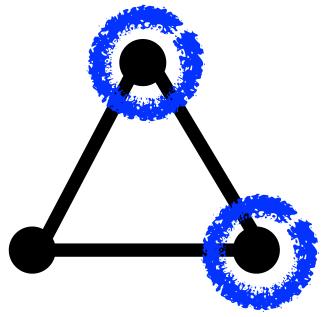


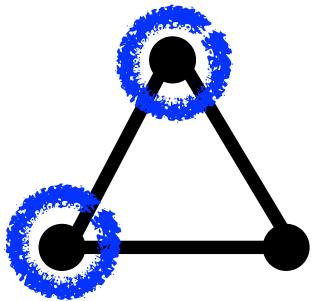
$$x_3 \vee \overline{x_3}$$











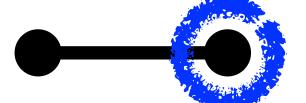


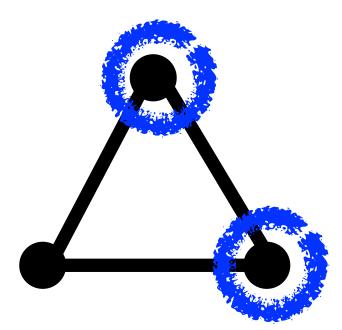


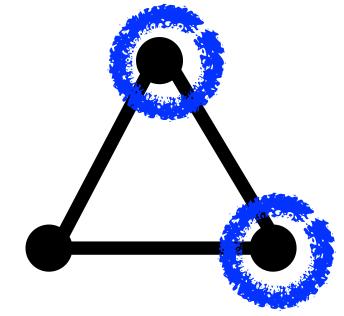
$$x_2 \vee \overline{x_2}$$

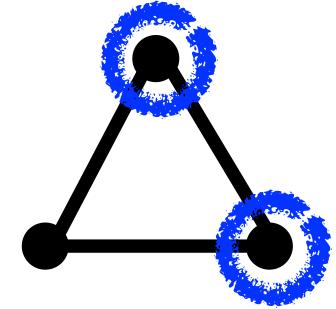


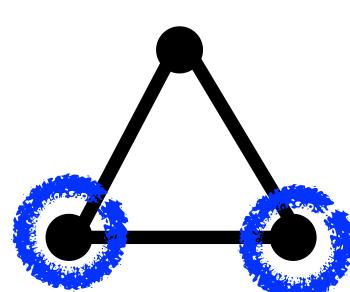
$$x_3 \vee \overline{x_3}$$











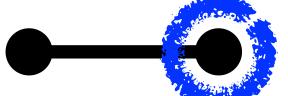


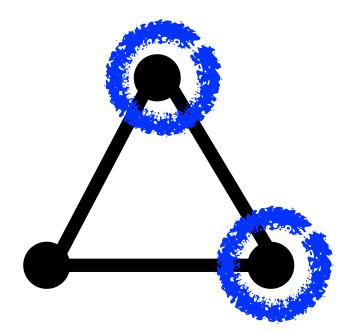


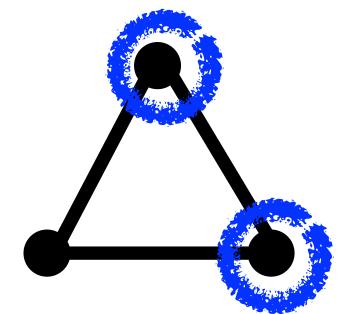
$$x_2 \vee \overline{x_2}$$

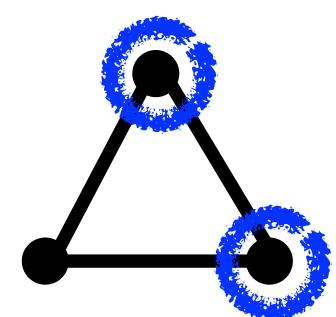


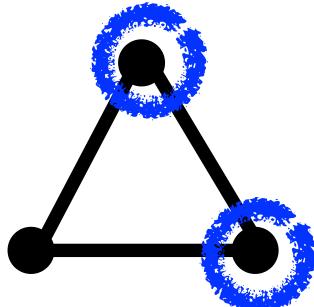
$$x_3 \vee \overline{x_3}$$











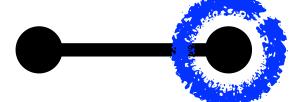




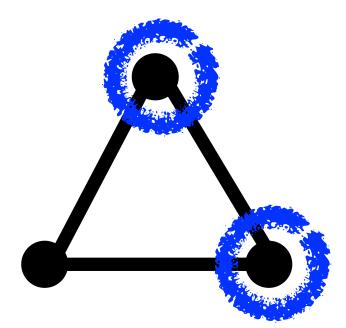


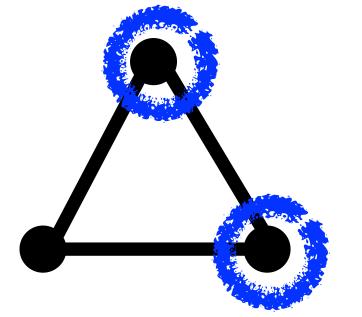


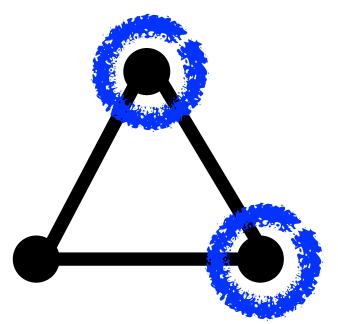
$$x_3 \vee \overline{x_3}$$

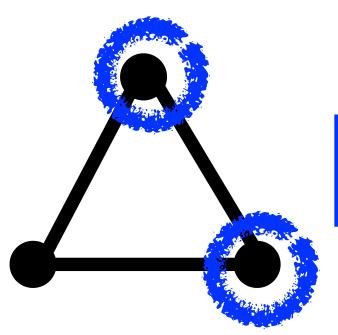


n Knoten

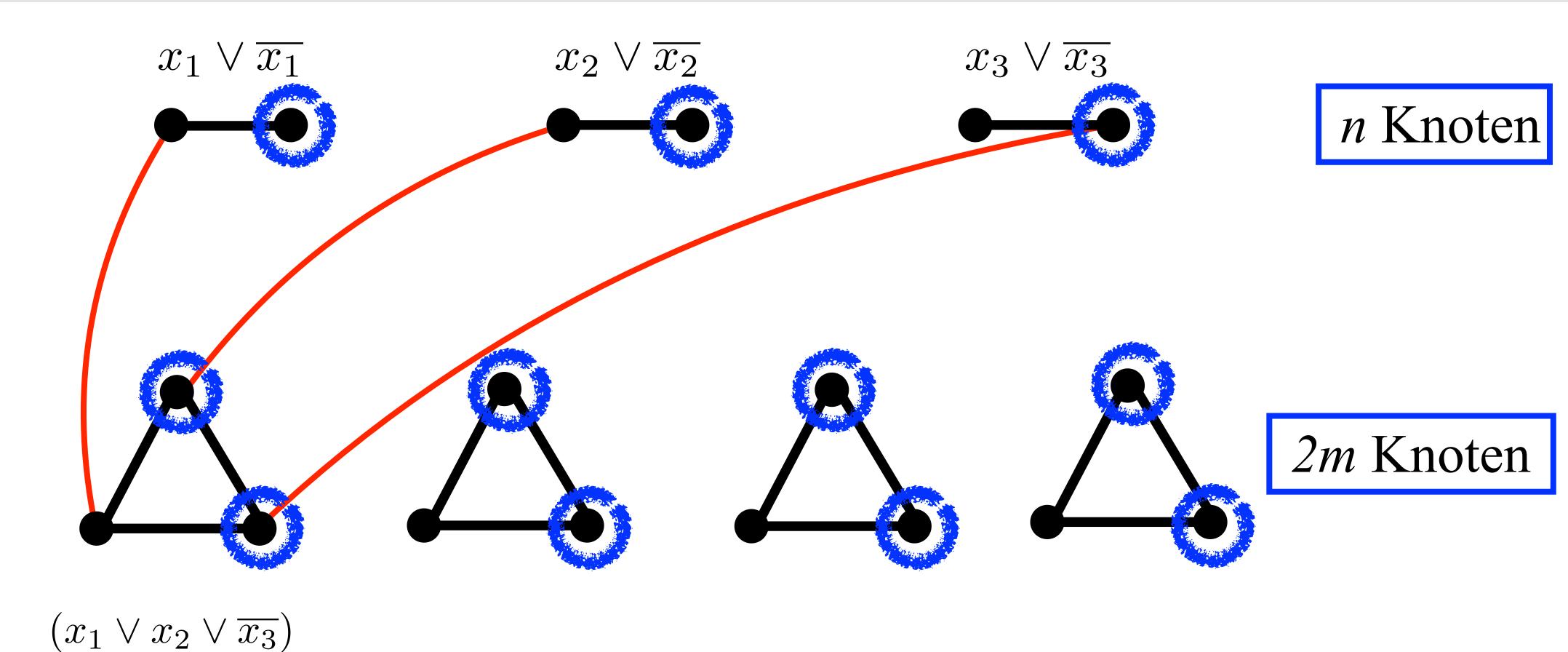




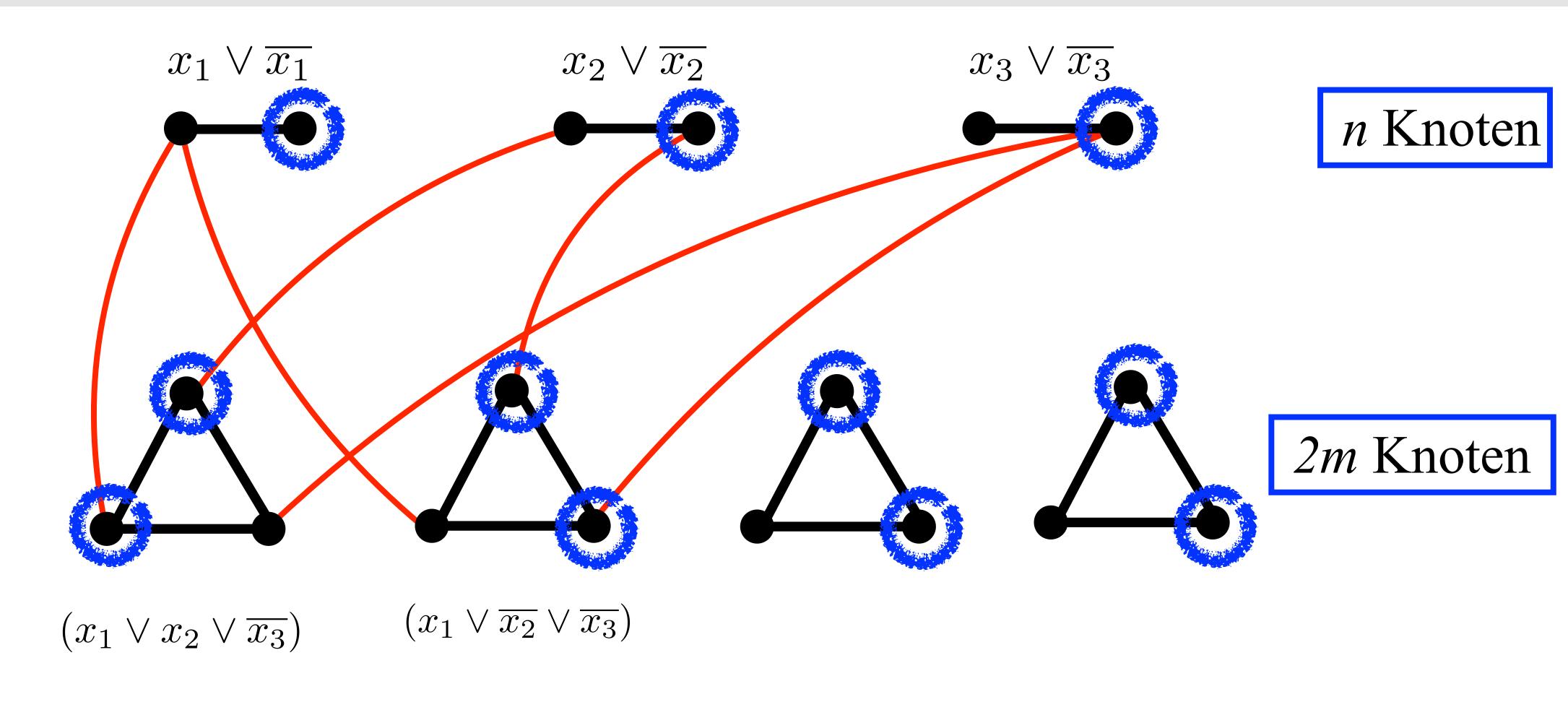




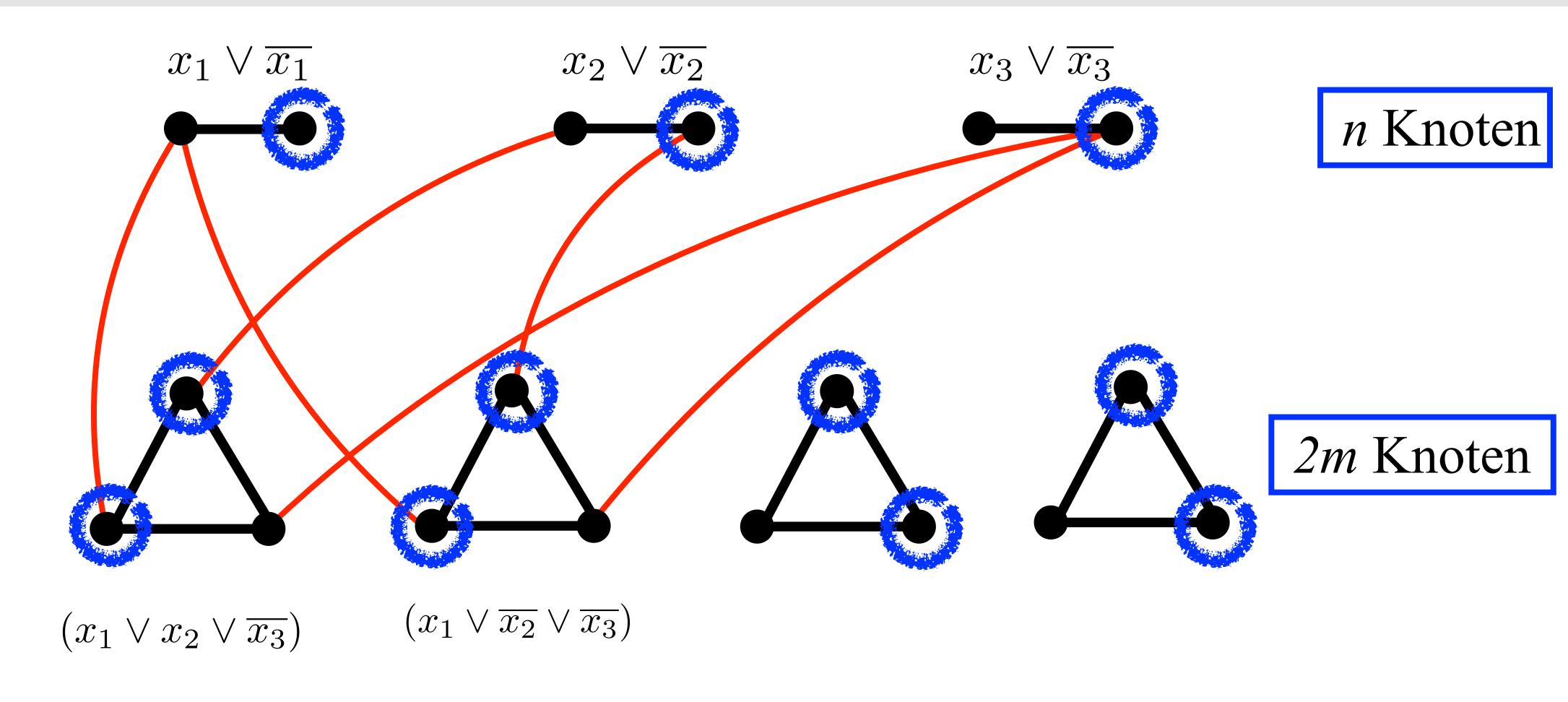
2m Knoten



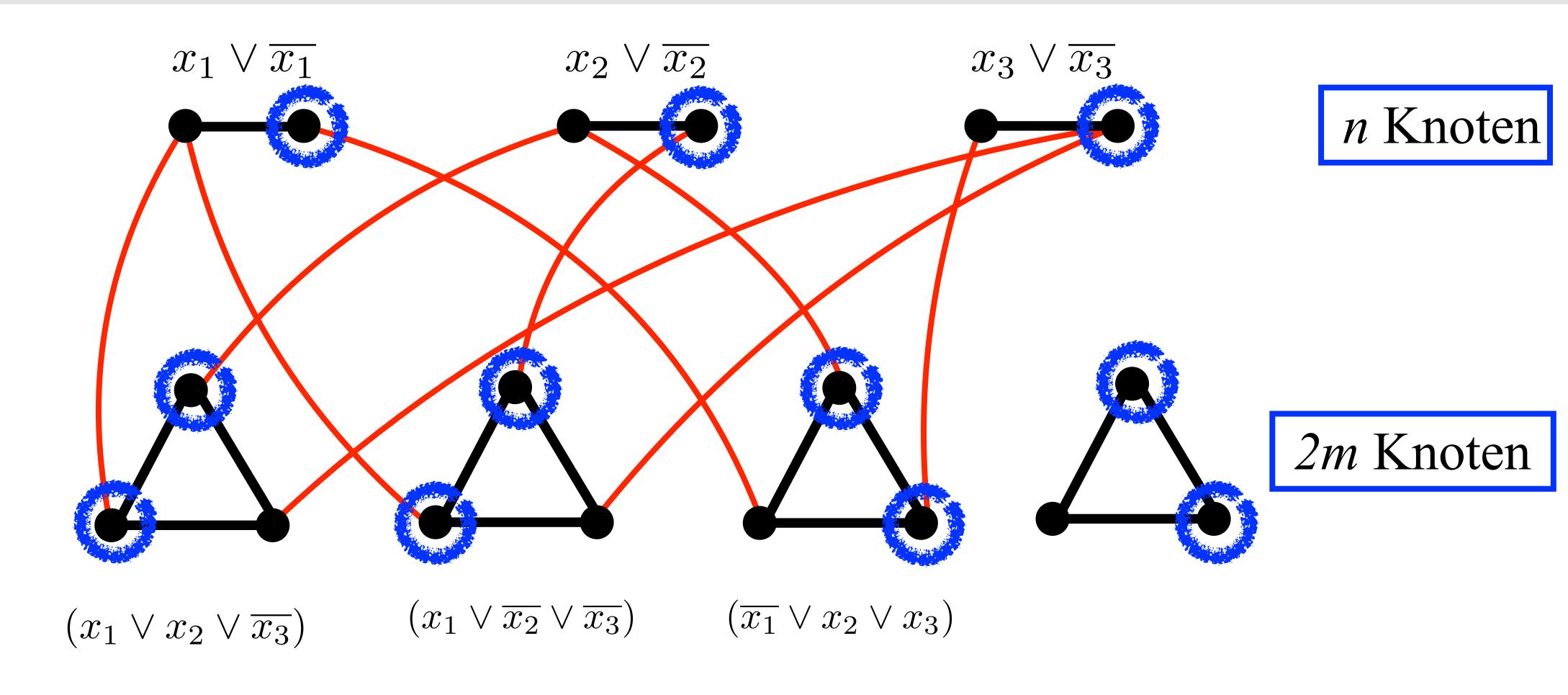




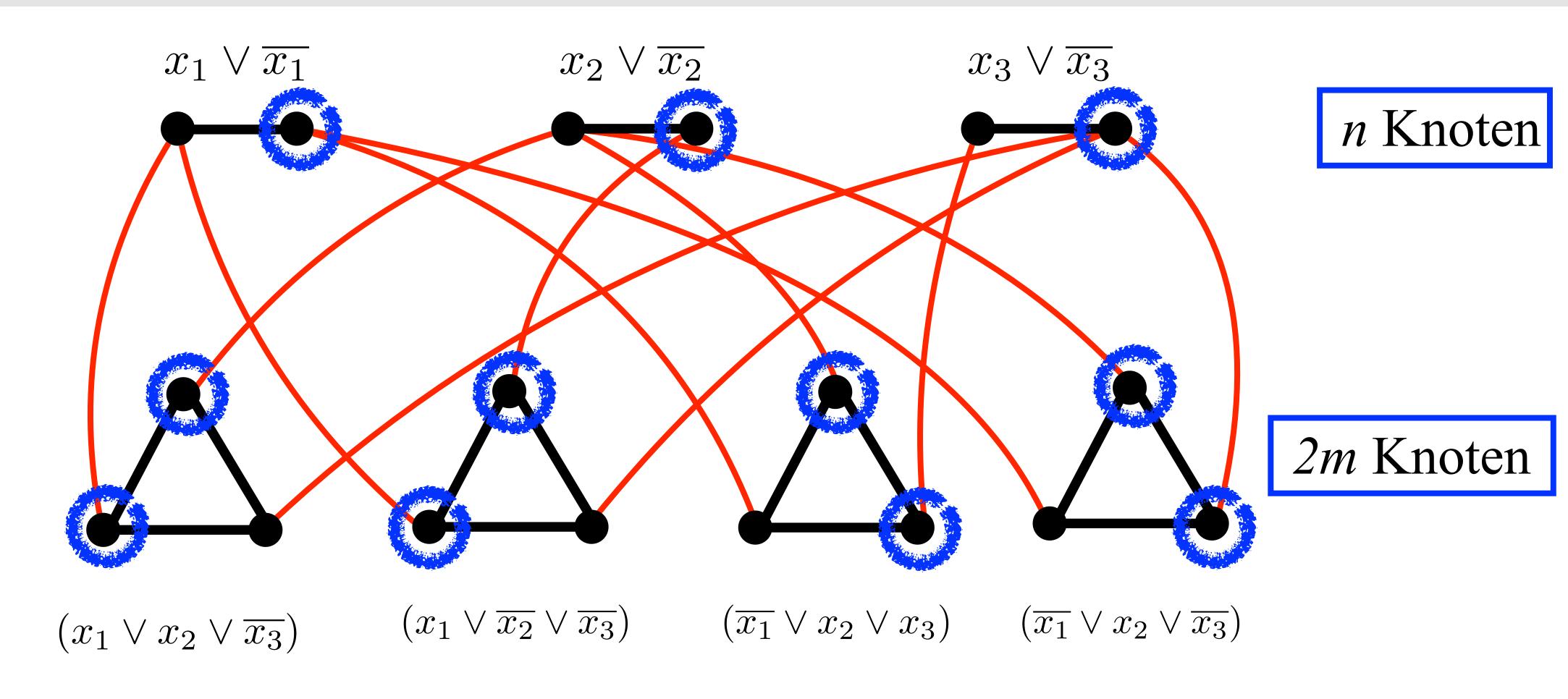




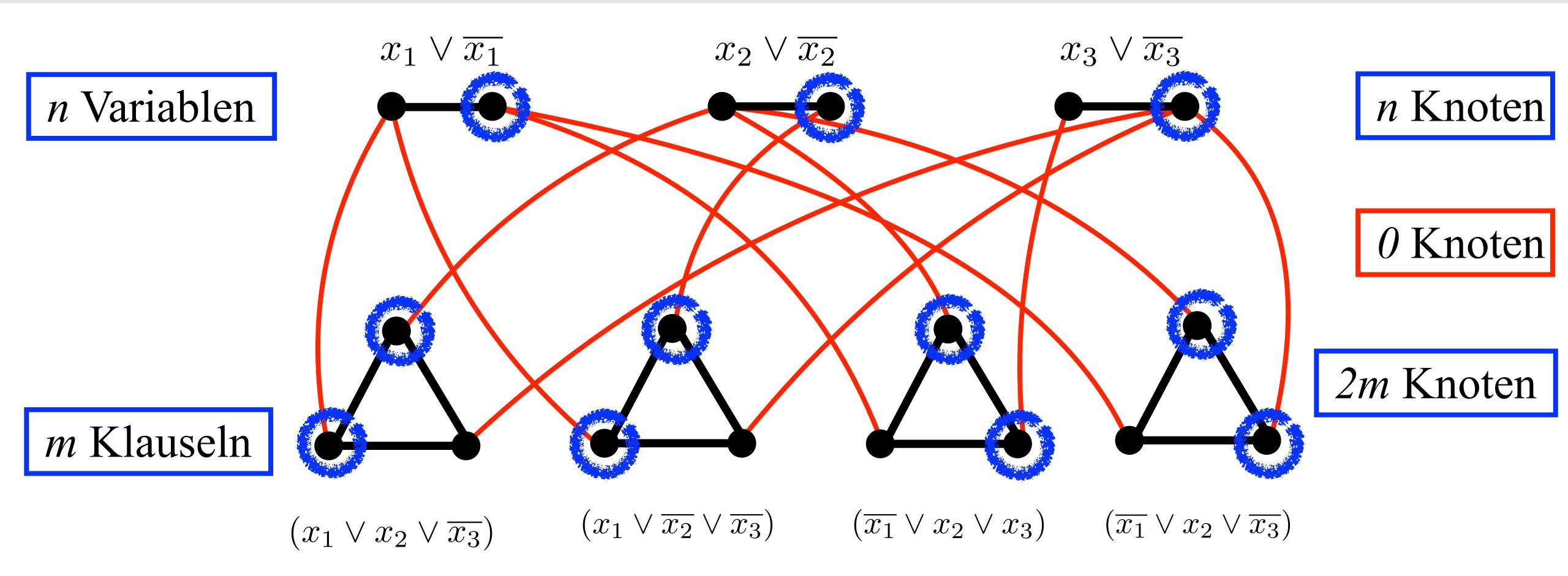






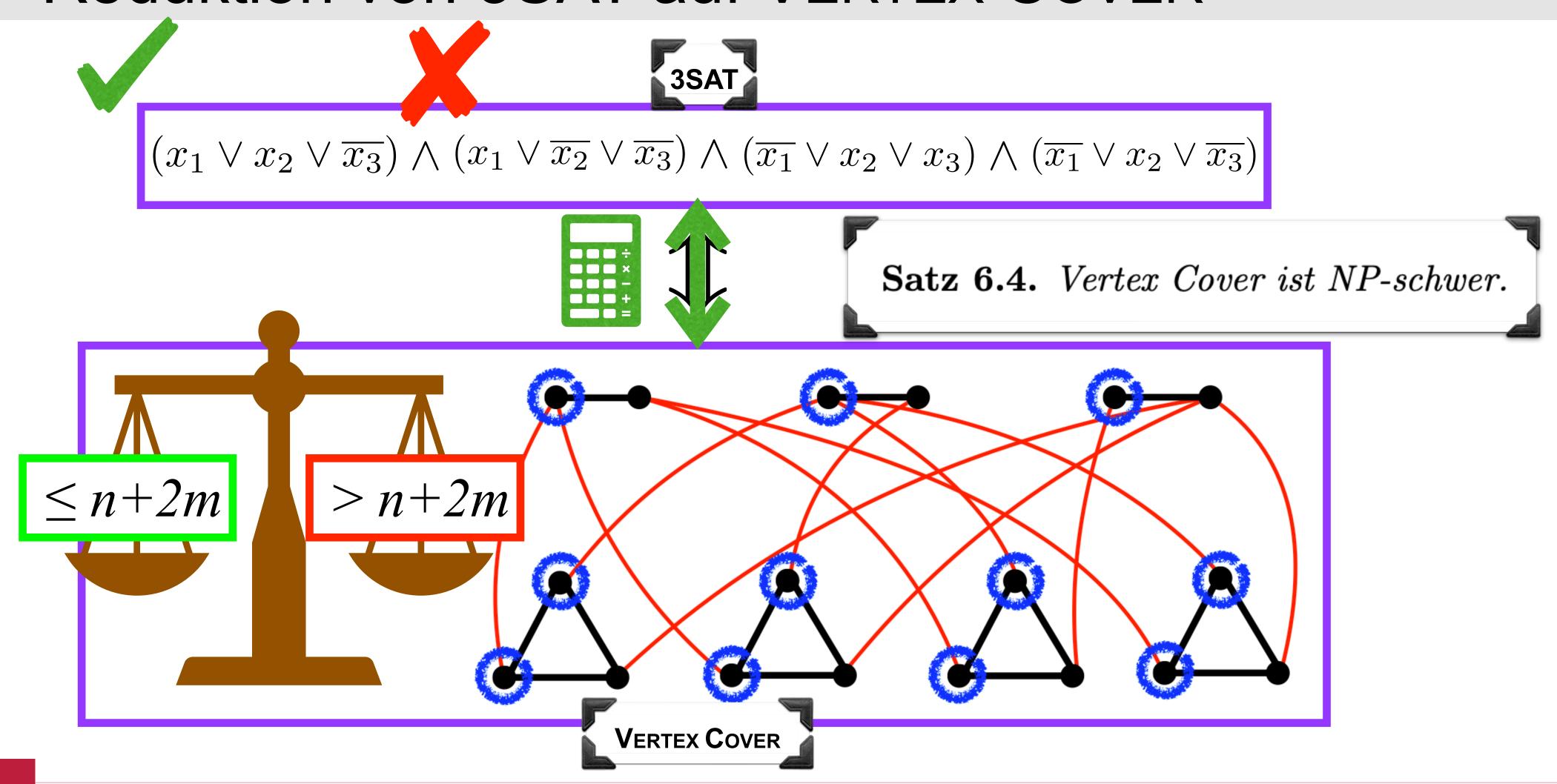








Reduktion von 3SAT auf VERTEX COVER





Vielen Dank!

s.fekete@tu-bs.de