

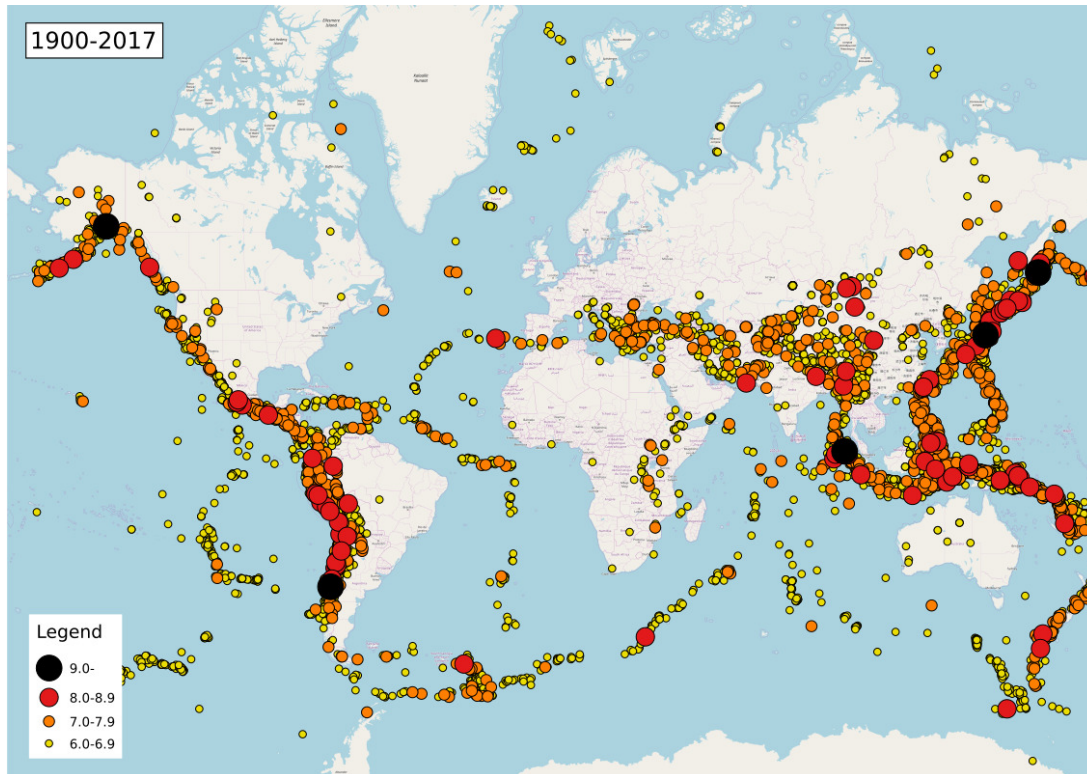
# Algorithmische Geometrie

## Schwere Probleme



# Proportional Symbol Maps

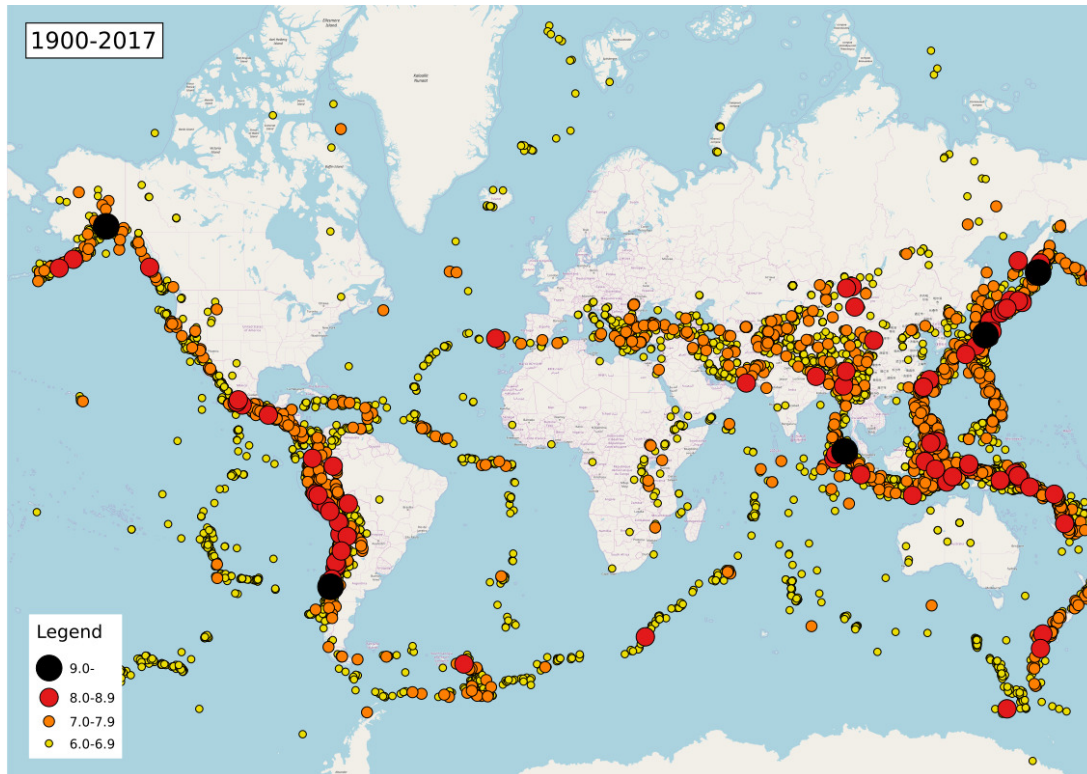
## Proportional Symbol Map (Beispiel: Erdbeben)



- Darstellung gewichteter Punkte auf einer Karte
- Gewicht wird durch Diskgröße repräsentiert

# Proportional Symbol Maps

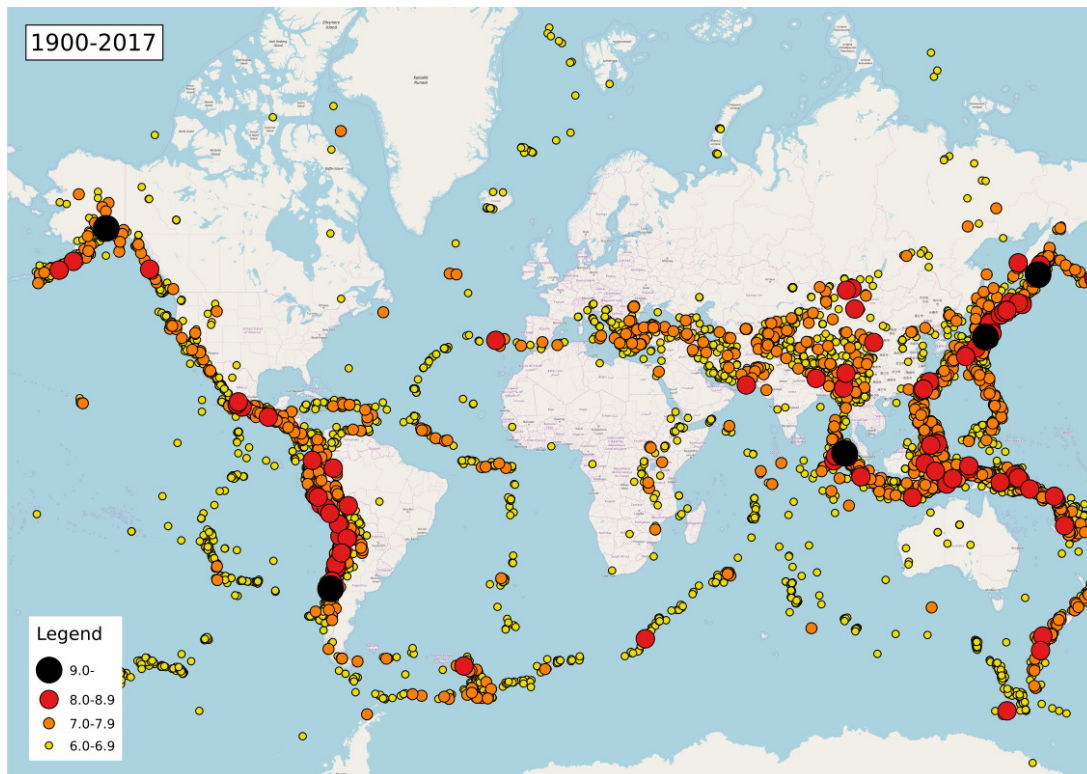
## Proportional Symbol Map (Beispiel: Erdbeben)



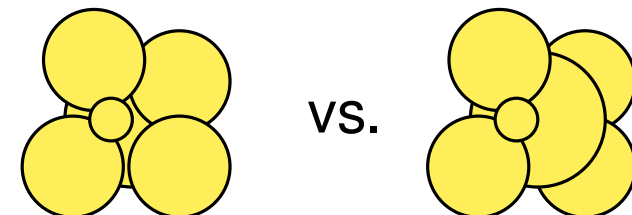
- Darstellung gewichteter Punkte auf einer Karte
- Gewicht wird durch Diskgröße repräsentiert
- Freiheitsgrad: Ordnung überlappender Disks

# Proportional Symbol Maps

## Proportional Symbol Map (Beispiel: Erdbeben)

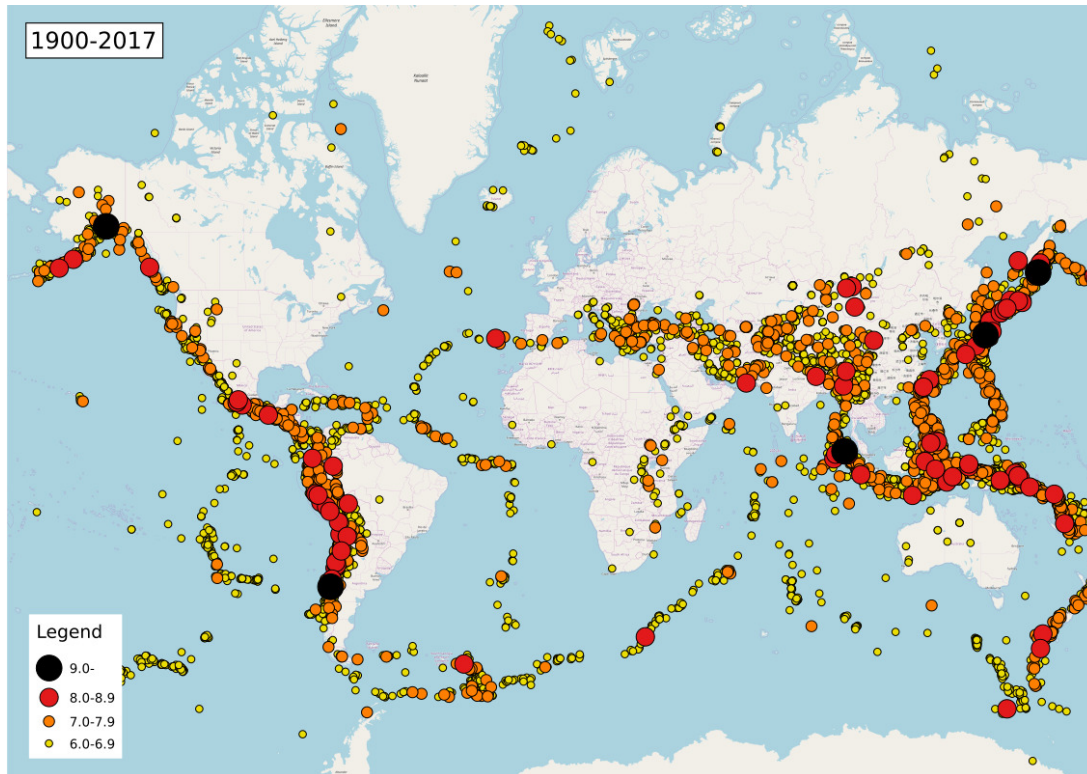


- Darstellung gewichteter Punkte auf einer Karte
- Gewicht wird durch Diskgröße repräsentiert
- Freiheitsgrad: Ordnung überlappender Disks
- Lesbarkeit hängt von der Ordnung ab

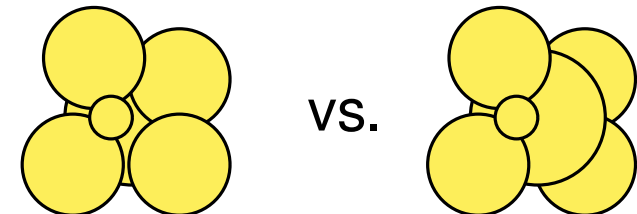


# Proportional Symbol Maps

## Proportional Symbol Map (Beispiel: Erdbeben)



- Darstellung gewichteter Punkte auf einer Karte
- Gewicht wird durch Diskgröße repräsentiert
- Freiheitsgrad: Ordnung überlappender Disks
- Lesbarkeit hängt von der Ordnung ab

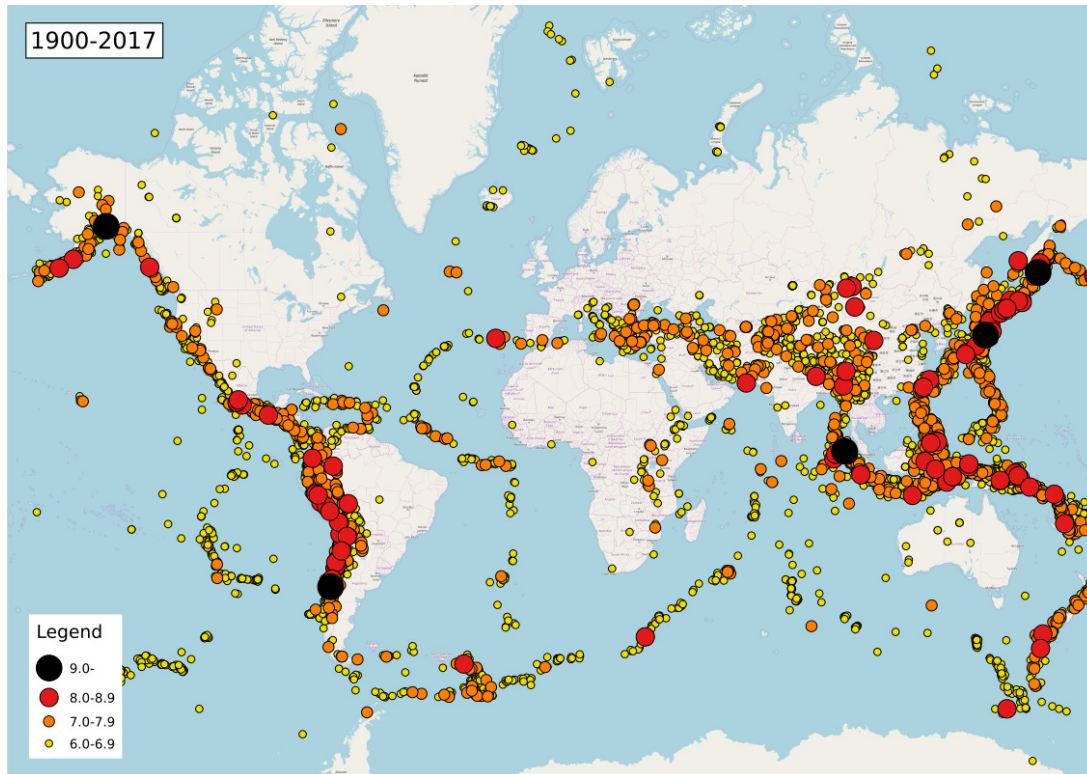


## Problemstellung

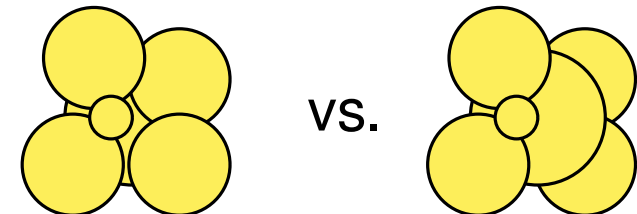
- gegeben: Menge von Disks mit potentiell unterschiedlichen Radien

# Proportional Symbol Maps

## Proportional Symbol Map (Beispiel: Erdbeben)



- Darstellung gewichteter Punkte auf einer Karte
- Gewicht wird durch Diskgröße repräsentiert
- Freiheitsgrad: Ordnung überlappender Disks
- Lesbarkeit hängt von der Ordnung ab

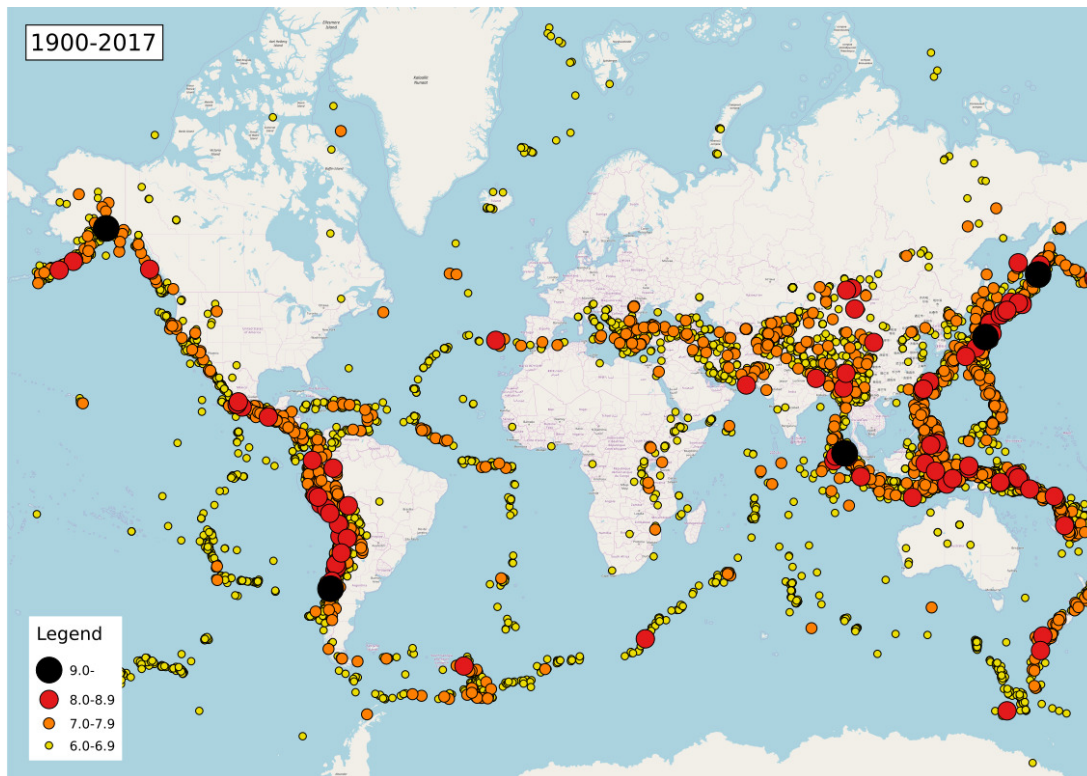


## Problemstellung

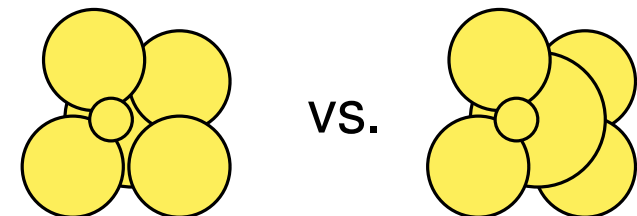
- gegeben: Menge von Disks mit potentiell unterschiedlichen Radien
- gesucht: Zeichnung, bei der für jede Disk möglichst viel Rand sichtbar ist

# Proportional Symbol Maps

## Proportional Symbol Map (Beispiel: Erdbeben)



- Darstellung gewichteter Punkte auf einer Karte
- Gewicht wird durch Diskgröße repräsentiert
- Freiheitsgrad: Ordnung überlappender Disks
- Lesbarkeit hängt von der Ordnung ab



## Problemstellung

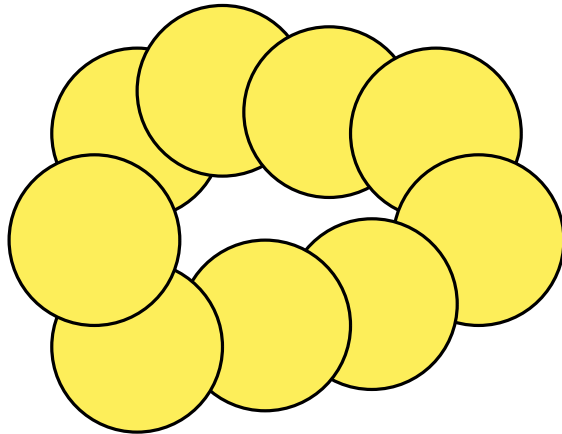
- gegeben: Menge von Disks mit potentiell unterschiedlichen Radien
- gesucht: Zeichnung, bei der für jede Disk möglichst viel Rand sichtbar ist
- Was ist eine gültige Zeichnung? Was genau heißt möglichst viel Rand?

# Was genau ist das Problem?

## Zwei Typen gültiger Zeichnungen

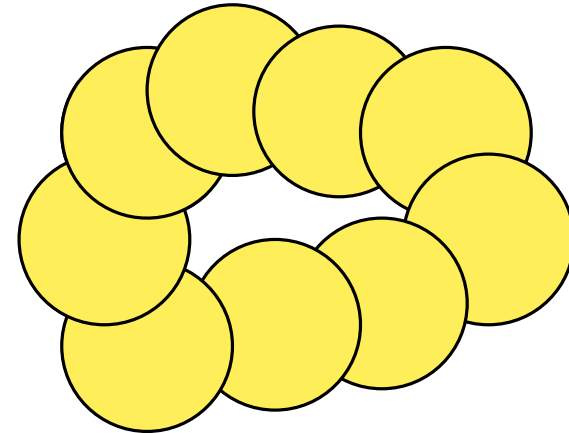
### Stacking

- totale z-Ordnung auf allen Disks



### Physikalisch realisierbare Zeichnung

- mit dünnen Münzen nachbaubar



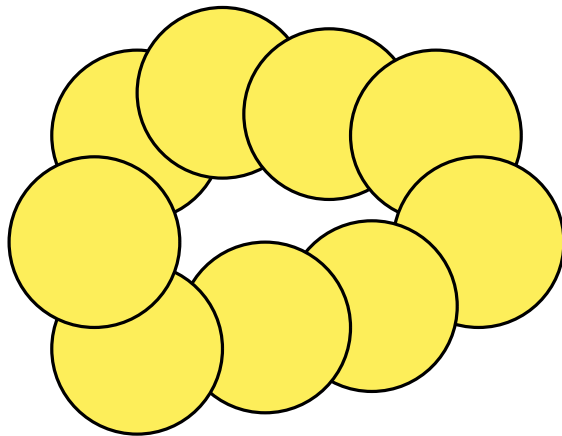


# Was genau ist das Problem?

## Zwei Typen gültiger Zeichnungen

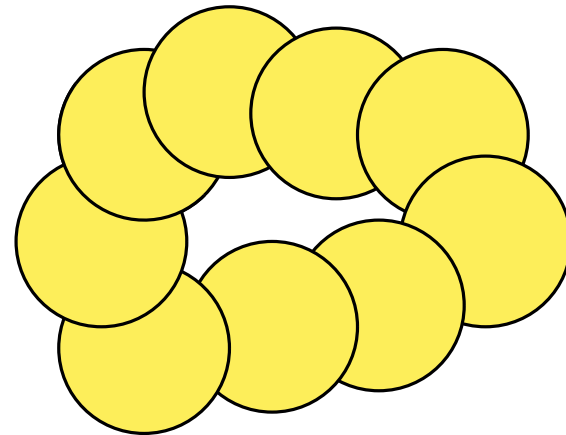
### Stacking

- totale z-Ordnung auf allen Disks



### Physikalisch realisierbare Zeichnung

- mit dünnen Münzen nachbaubar



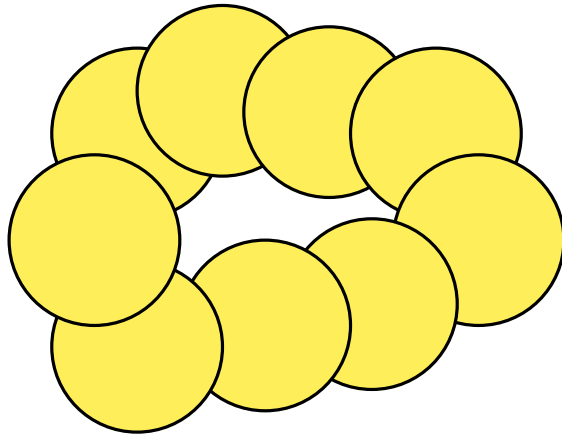
- jedes Stacking ist physikalisch realisierbar aber nicht umgekehrt

# Was genau ist das Problem?

## Zwei Typen gültiger Zeichnungen

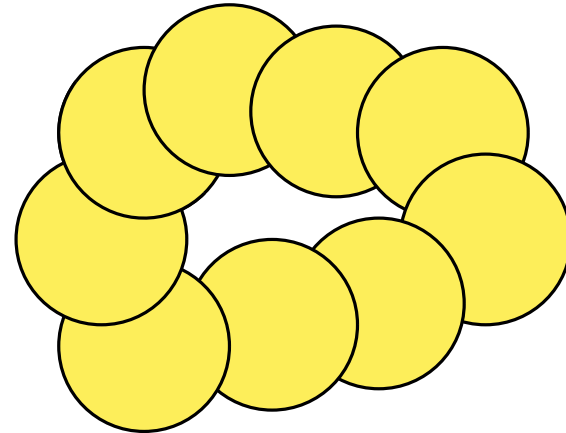
### Stacking

- totale z-Ordnung auf allen Disks



### Physikalisch realisierbare Zeichnung

- mit dünnen Münzen nachbaubar



- jedes Stacking ist physikalisch realisierbar aber nicht umgekehrt

## Zwei Optimierungsprobleme

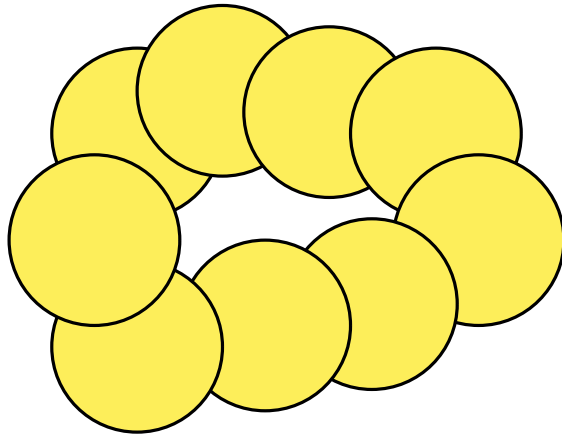
- Max-Min: maximiere den minimal sichtbaren Rand unter allen Disks
- Max-Total: maximiere den insgesamt sichtbaren Rand

# Was genau ist das Problem?

## Zwei Typen gültiger Zeichnungen

### Stacking

- totale z-Ordnung auf allen Disks



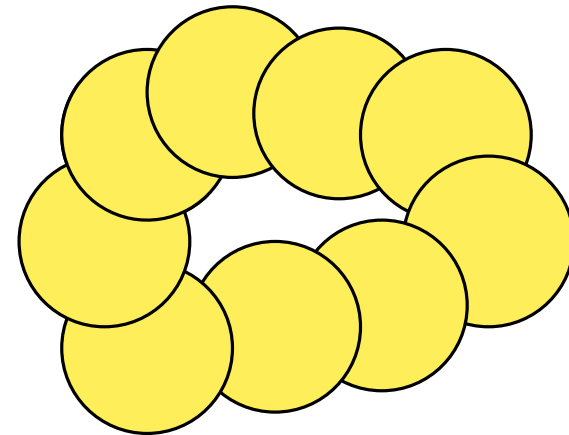
- jedes Stacking ist physikalisch realisierbar aber nicht umgekehrt

## Zwei Optimierungsprobleme

- Max-Min: maximiere den minimal sichtbaren Rand unter allen Disks
- Max-Total: maximiere den insgesamt sichtbaren Rand

### Physikalisch realisierbare Zeichnung

- mit dünnen Münzen nachbaubar



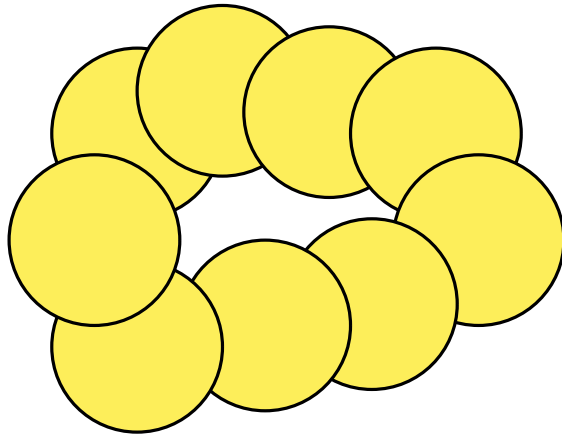
	Max-Min	Max-Total
Stacking	P	?
physikalisch realisierbar	NP-Schwer	NP-Schwer

# Was genau ist das Problem?

## Zwei Typen gültiger Zeichnungen

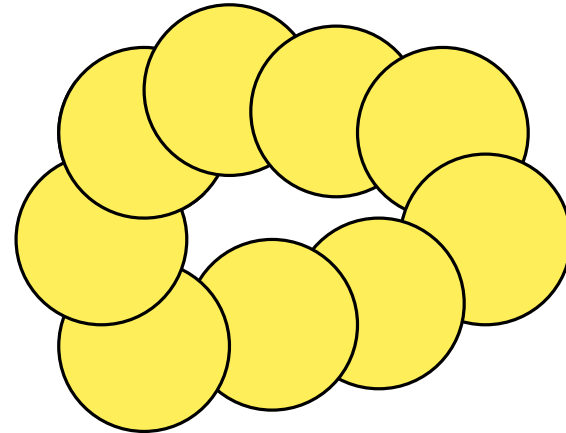
### Stacking

- totale z-Ordnung auf allen Disks



### Physikalisch realisierbare Zeichnung

- mit dünnen Münzen nachbaubar



- jedes Stacking ist physikalisch realisierbar aber nicht umgekehrt

## Zwei Optimierungsprobleme

- Max-Min: maximiere den minimal sichtbaren Rand unter allen Disks
- Max-Total: maximiere den insgesamt sichtbaren Rand

	Max-Min	Max-Total
Stacking	P	?
physikalisch realisierbar	NP-Schwer	NP-Schwer

↑  
heute

# Nützliche NP-schwere Sat-Varianten

## Problem: 3-Sat

Seien  $x_1, \dots, x_n$  Boolesche Variablen und sei  $\mathcal{C} = C_1 \wedge \dots \wedge C_m$  eine Formel in CNF, wobei jede Klausel  $C_i$  maximal drei Literale enthält. Gibt es eine Variablenbelegung, die  $\mathcal{C}$  zu WAHR auswertet?

$$\begin{aligned} &(x_1 \vee \neg x_2 \vee \neg x_3) \\ &\wedge (x_2 \vee \neg x_4 \vee x_5) \\ &\wedge (\neg x_1 \vee x_4) \end{aligned}$$

# Nützliche NP-schwere Sat-Varianten

## Problem: 3-Sat

Seien  $x_1, \dots, x_n$  Boolesche Variablen und sei  $\mathcal{C} = C_1 \wedge \dots \wedge C_m$  eine Formel in CNF, wobei jede Klausel  $C_i$  maximal drei Literale enthält. Gibt es eine Variablenbelegung, die  $\mathcal{C}$  zu WAHR auswertet?

$$\begin{aligned}
 &(x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee \neg x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee x_4)
 \end{aligned}$$

## Problem: monotone 3-Sat

Jede Klausel enthält nur positive oder nur negative Literale.

$$\begin{aligned}
 &(\neg x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee \neg x_4)
 \end{aligned}$$

# Nützliche NP-schwere Sat-Varianten

## Problem: 3-Sat

Seien  $x_1, \dots, x_n$  Boolesche Variablen und sei  $\mathcal{C} = C_1 \wedge \dots \wedge C_m$  eine Formel in CNF, wobei jede Klausel  $C_i$  maximal drei Literale enthält. Gibt es eine Variablenbelegung, die  $\mathcal{C}$  zu WAHR auswertet?

$$\begin{aligned}
 &(x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee \neg x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee x_4)
 \end{aligned}$$

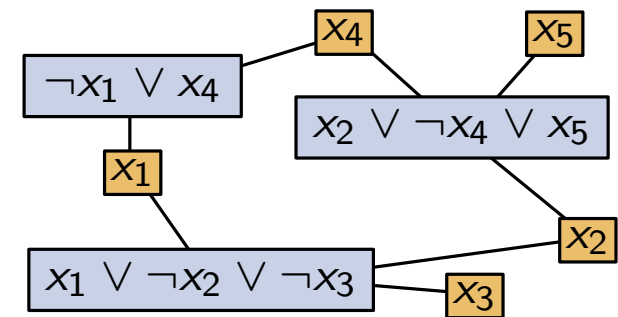
## Problem: monotone 3-Sat

Jede Klausel enthält nur positive oder nur negative Literale.

$$\begin{aligned}
 &(\neg x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee \neg x_4)
 \end{aligned}$$

## Problem: planar 3-Sat

Der Variablen-Klausel-Graph ist planar.



# Nützliche NP-schwere Sat-Varianten

## Problem: 3-Sat

Seien  $x_1, \dots, x_n$  Boolesche Variablen und sei  $\mathcal{C} = C_1 \wedge \dots \wedge C_m$  eine Formel in CNF, wobei jede Klausel  $C_i$  maximal drei Literale enthält. Gibt es eine Variablenbelegung, die  $\mathcal{C}$  zu WAHR auswertet?

$$\begin{aligned}
 &(x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee \neg x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee x_4)
 \end{aligned}$$

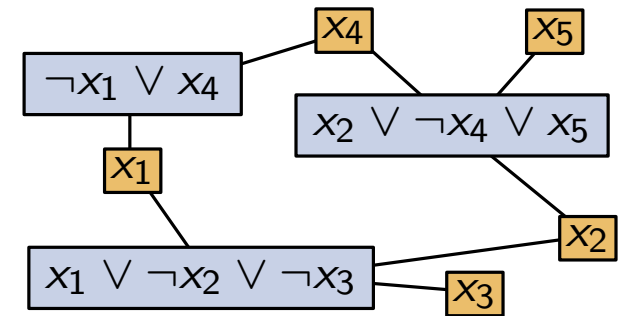
## Problem: monotone 3-Sat

Jede Klausel enthält nur positive oder nur negative Literale.

$$\begin{aligned}
 &(\neg x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee \neg x_4)
 \end{aligned}$$

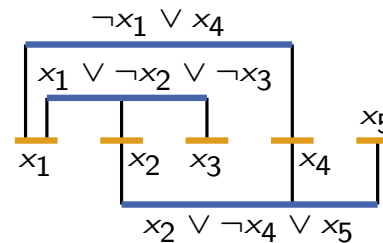
## Problem: planar 3-Sat

Der Variablen-Klausel-Graph ist planar.



## Problem: rectilinear planar 3-Sat

Der Variablen-Klausel-Graph hat eine *rechtwinklige planare* Zeichnung.



## rectw. plan. Zeichnung

- Knoten/Kanten: horizontale/vertikale Strecken
- alle Variablen-Knoten liegen auf einer Geraden



# Nützliche NP-schwere Sat-Varianten

## Problem: 3-Sat

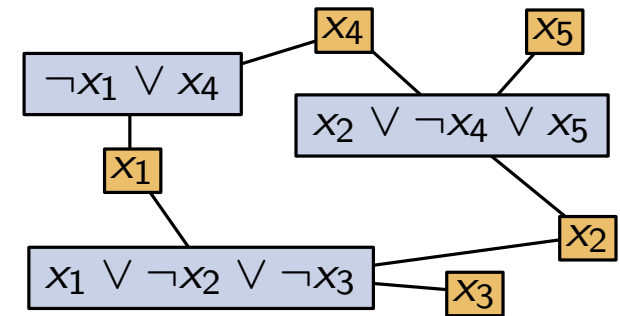
Seien  $x_1, \dots, x_n$  Boolesche Variablen und sei  $\mathcal{C} = C_1 \wedge \dots \wedge C_m$  eine Formel in CNF, wobei jede Klausel  $C_i$  maximal drei Literale enthält. Gibt es eine Variablenbelegung, die  $\mathcal{C}$  zu WAHR auswertet?

$$\begin{aligned}
 &(x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee \neg x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee x_4)
 \end{aligned}$$

## Problem: monotone 3-Sat

Jede Klausel enthält nur positive oder nur negative Literale.

$$\begin{aligned}
 &(\neg x_1 \vee \neg x_2 \vee \neg x_3) \\
 &\wedge (x_2 \vee x_4 \vee x_5) \\
 &\wedge (\neg x_1 \vee \neg x_4)
 \end{aligned}$$

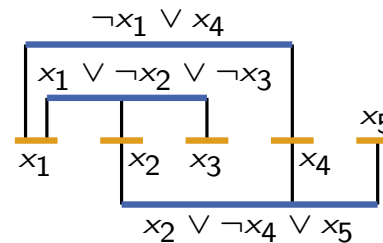


## Problem: planar 3-Sat

Der Variablen-Klausel-Graph ist planar.

## Problem: rectilinear planar 3-Sat

Der Variablen-Klausel-Graph hat eine *rechtwinklige planare* Zeichnung.

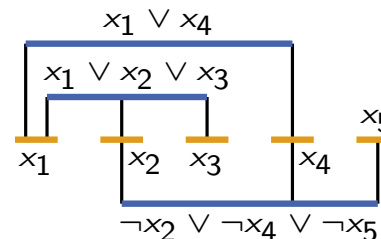


## rectw. plan. Zeichnung

- Knoten/Kanten: horizontale/vertikale Strecken
- alle Variablen-Knoten liegen auf einer Geraden

## Problem: planar monotone 3-Sat

Klauseln über/unter den Variablen enthalten nur positive/negative Literale.



# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks

# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen

# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen
- Erfüllung aller Klauseln  $\hat{=}$  großer sichtbarer Rand für jede Disk

# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen
- Erfüllung aller Klauseln  $\hat{=}$  großer sichtbarer Rand für jede Disk

## Zu modellierende Bestandteile

- Variablen
  - $n$  unabhängige binäre Entscheidungen
  - keine zusätzlichen Entscheidungen (alles weitere ist erzwungen)

# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen
- Erfüllung aller Klauseln  $\hat{=}$  großer sichtbarer Rand für jede Disk

## Zu modellierende Bestandteile

- Variablen
  - $n$  unabhängige binäre Entscheidungen
  - keine zusätzlichen Entscheidungen (alles weitere ist erzwungen)
- Klauseln
  - Klausel macht Probleme  $\Leftrightarrow$  drei bestimmte Entscheidungen sind falsch

# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen
- Erfüllung aller Klauseln  $\hat{=}$  großer sichtbarer Rand für jede Disk

## Zu modellierende Bestandteile

- Variablen
  - $n$  unabhängige binäre Entscheidungen
  - keine zusätzlichen Entscheidungen (alles weitere ist erzwungen)
- Klauseln
  - Klausel macht Probleme  $\Leftrightarrow$  drei bestimmte Entscheidungen sind falsch
- Informationstransport
  - Transport der Variablen-Entscheidungen zu den Klauseln

# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen
- Erfüllung aller Klauseln  $\hat{=}$  großer sichtbarer Rand für jede Disk

## Zu modellierende Bestandteile

- Variablen
  - $n$  unabhängige binäre Entscheidungen
  - keine zusätzlichen Entscheidungen (alles weitere ist erzwungen)
- Klauseln
  - Klausel macht Probleme  $\Leftrightarrow$  drei bestimmte Entscheidungen sind falsch
- Informationstransport
  - Transport der Variablen-Entscheidungen zu den Klauseln
  - man muss positive und negative Literale abgreifen können



# Reduktion von rectilinear planar 3-Sat

## Grundsätzliches Mindset

- wir wollen eine gegebene 3-Sat Instanz modellieren
- unsere Modellierungssprache sind überlappende Disks
- Variablenbelegung  $\hat{=}$  Entscheidung wie Disks aufeinander liegen
- Erfüllung aller Klauseln  $\hat{=}$  großer sichtbarer Rand für jede Disk

## Zu modellierende Bestandteile

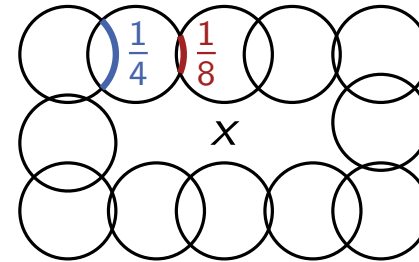
- Variablen
  - $n$  unabhängige binäre Entscheidungen
  - keine zusätzlichen Entscheidungen (alles weitere ist erzwungen)
- Klauseln
  - Klausel macht Probleme  $\Leftrightarrow$  drei bestimmte Entscheidungen sind falsch
- Informationstransport
  - Transport der Variablen-Entscheidungen zu den Klauseln
  - man muss positive und negative Literale abgreifen können
  - Informationstransport darf in eine Richtung unsauber sein: Flip von erfülltem zu nicht-erfülltem Literal ist OK

# Gadgets

**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

# Gadgets

**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar  
**Variablen**

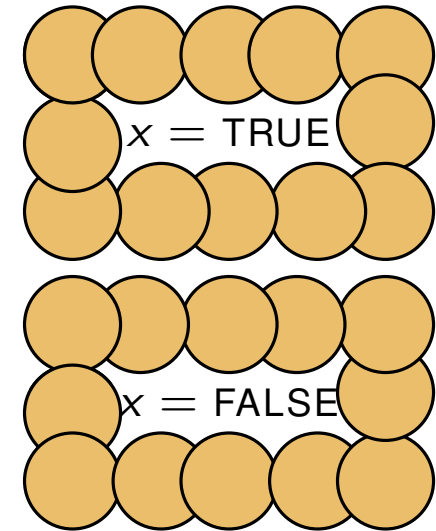
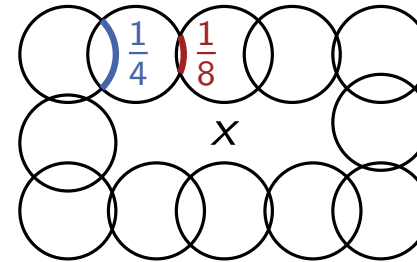


# Gadgets

**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

## Variablen

- nur zwei Konfigurationen möglich
- jede weitere Konfiguration deckt mehr als  $1/4$  einer Disk ab

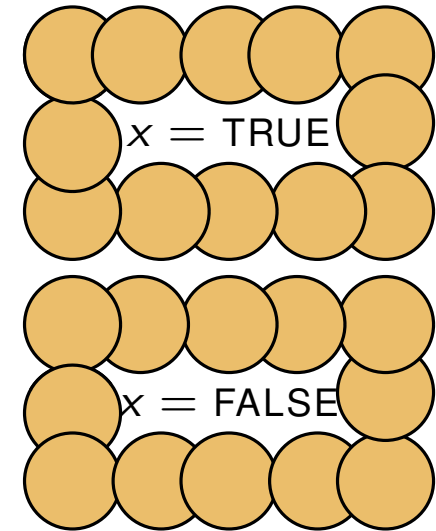
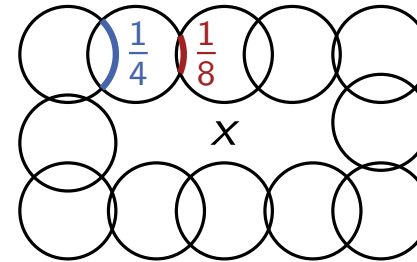


# Gadgets

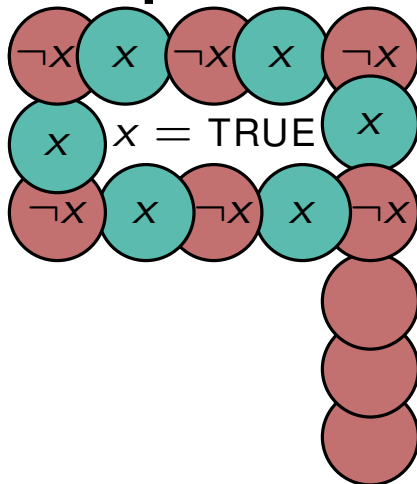
**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

## Variablen

- nur zwei Konfigurationen möglich
- jede weitere Konfiguration deckt mehr als  $1/4$  einer Disk ab



## Transport der Literale



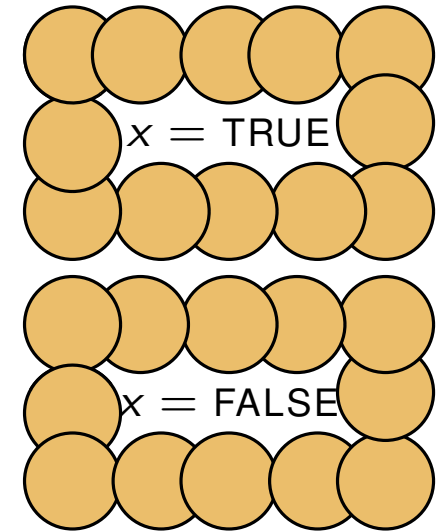
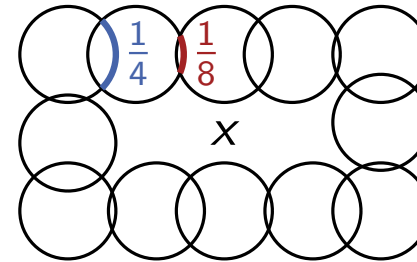
- Annahme: Konfiguration für  $x = \text{TRUE}$
- Kette startet bei einer  $\neg x$ -Disk

# Gadgets

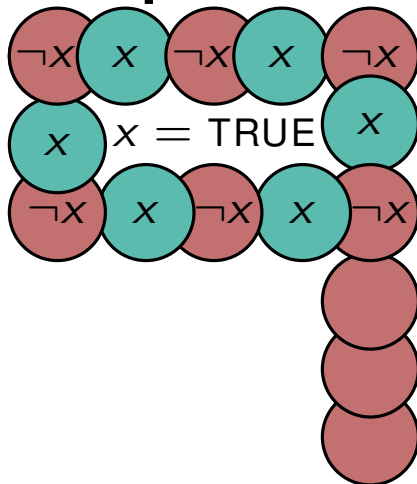
**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

## Variablen

- nur zwei Konfigurationen möglich
- jede weitere Konfiguration deckt mehr als  $1/4$  einer Disk ab



## Transport der Literale



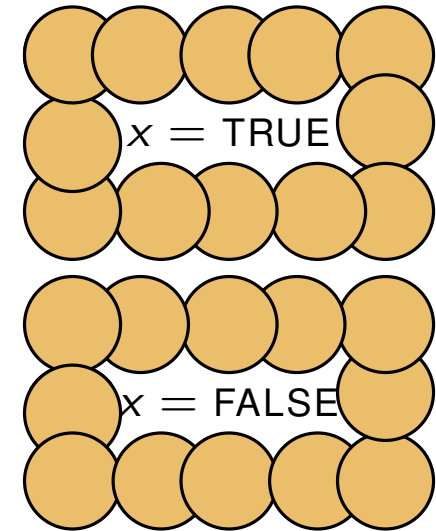
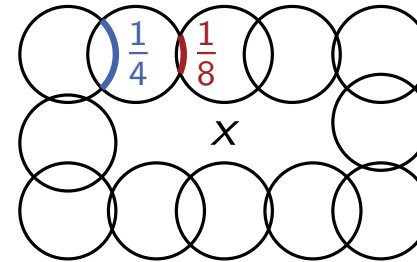
- Annahme: Konfiguration für  $x = \text{TRUE}$
- Kette startet bei einer  $\neg x$ -Disk
  - jede Entscheidung in der Kette ist erzwungen
  - letzte Disk der Kette ist schon zu  $1/4$  abgedeckt
  - letzte Disk kann im Prinzip beliebige Position haben

# Gadgets

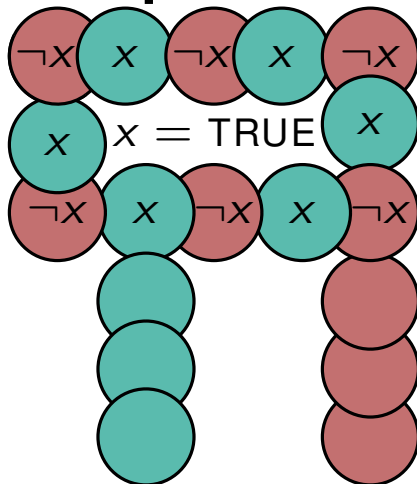
**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

## Variablen

- nur zwei Konfigurationen möglich
- jede weitere Konfiguration deckt mehr als  $1/4$  einer Disk ab



## Transport der Literale



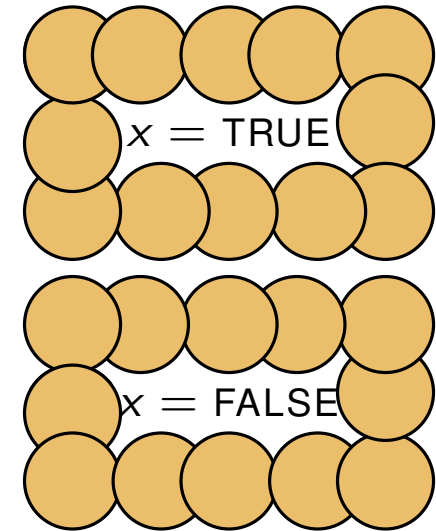
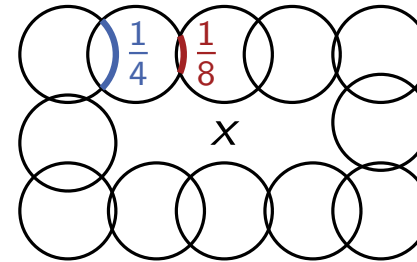
- Annahme: Konfiguration für  $x = \text{TRUE}$
- Kette startet bei einer  $\neg x$ -Disk
  - jede Entscheidung in der Kette ist erzwungen
  - letzte Disk der Kette ist schon zu  $1/4$  abgedeckt
  - letzte Disk kann im Prinzip beliebige Position haben
- Kette startet bei einer  $x$ -Disk
  - mögliche Konfiguration: letzte Disk komplett sichtbar

# Gadgets

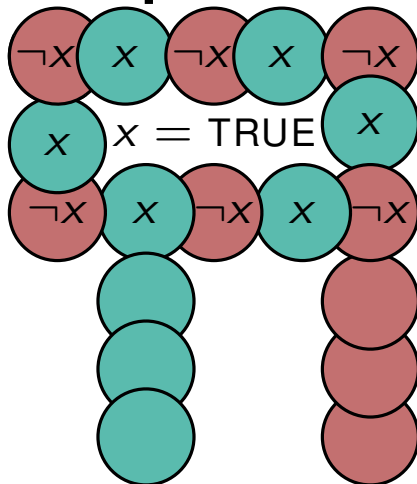
**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

## Variablen

- nur zwei Konfigurationen möglich
- jede weitere Konfiguration deckt mehr als  $1/4$  einer Disk ab



## Transport der Literale



- Annahme: Konfiguration für  $x = \text{TRUE}$
- Kette startet bei einer  $\neg x$ -Disk
  - jede Entscheidung in der Kette ist erzwungen
  - letzte Disk der Kette ist schon zu  $1/4$  abgedeckt
  - letzte Disk kann im Prinzip beliebige Position haben
- Kette startet bei einer  $x$ -Disk
  - mögliche Konfiguration: letzte Disk komplett sichtbar
- invertiertes Verhalten, für  $x = \text{FALSE}$  Konfiguration

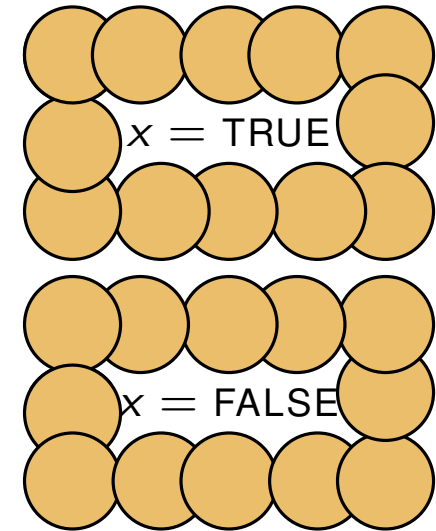
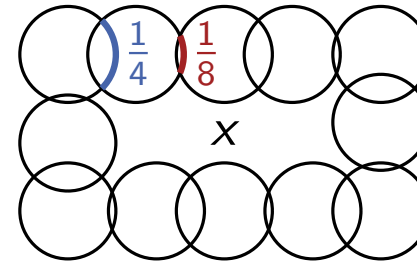


# Gadgets

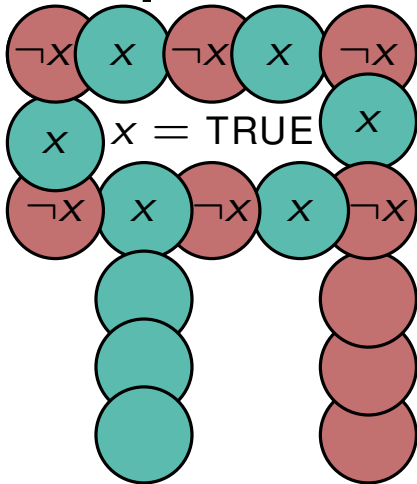
**JA-Instanzen:** für jede Disk ist der Rand zu  $\geq 3/4$  sichtbar

## Variablen

- nur zwei Konfigurationen möglich
- jede weitere Konfiguration deckt mehr als  $1/4$  einer Disk ab



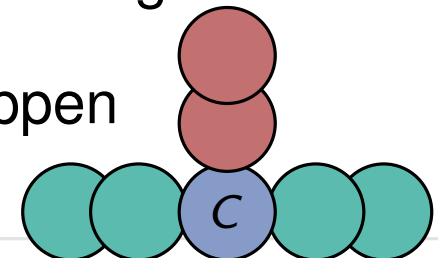
## Transport der Literale



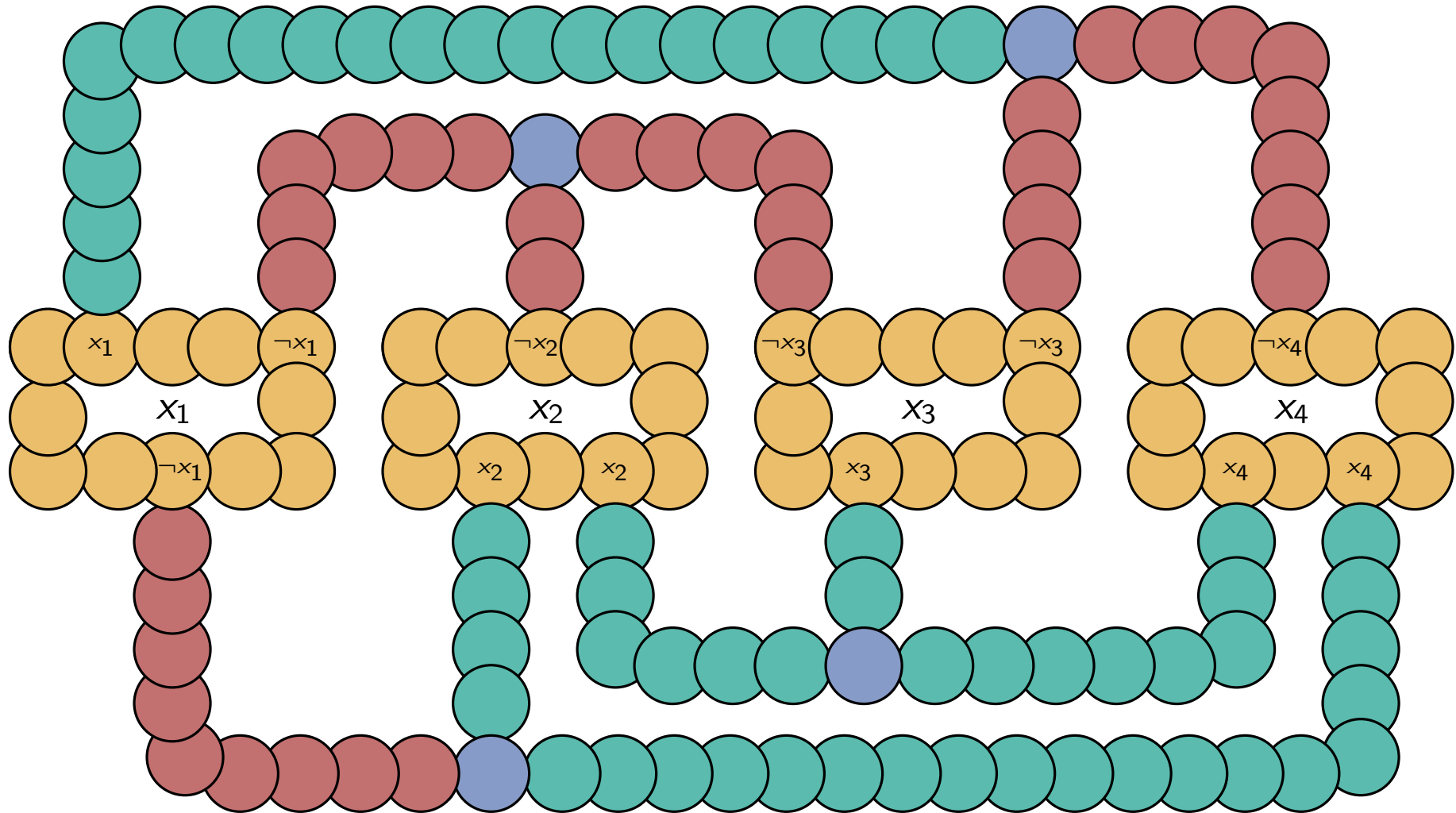
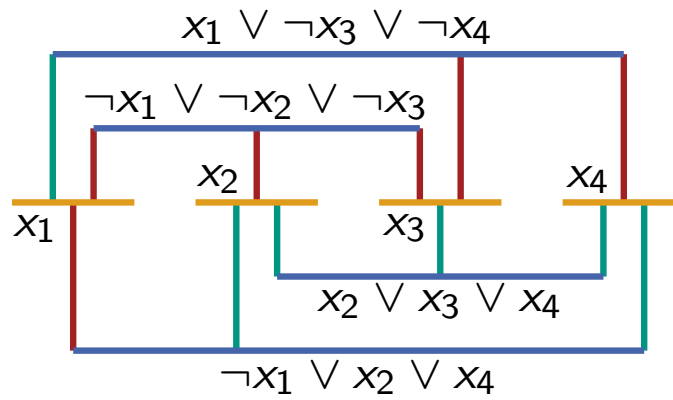
- Annahme: Konfiguration für  $x = \text{TRUE}$
- Kette startet bei einer  $\neg x$ -Disk
  - jede Entscheidung in der Kette ist erzwungen
  - letzte Disk der Kette ist schon zu  $1/4$  abgedeckt
  - letzte Disk kann im Prinzip beliebige Position haben
- Kette startet bei einer  $x$ -Disk
  - mögliche Konfiguration: letzte Disk komplett sichtbar
- invertiertes Verhalten, für  $x = \text{FALSE}$  Konfiguration

## Klausel

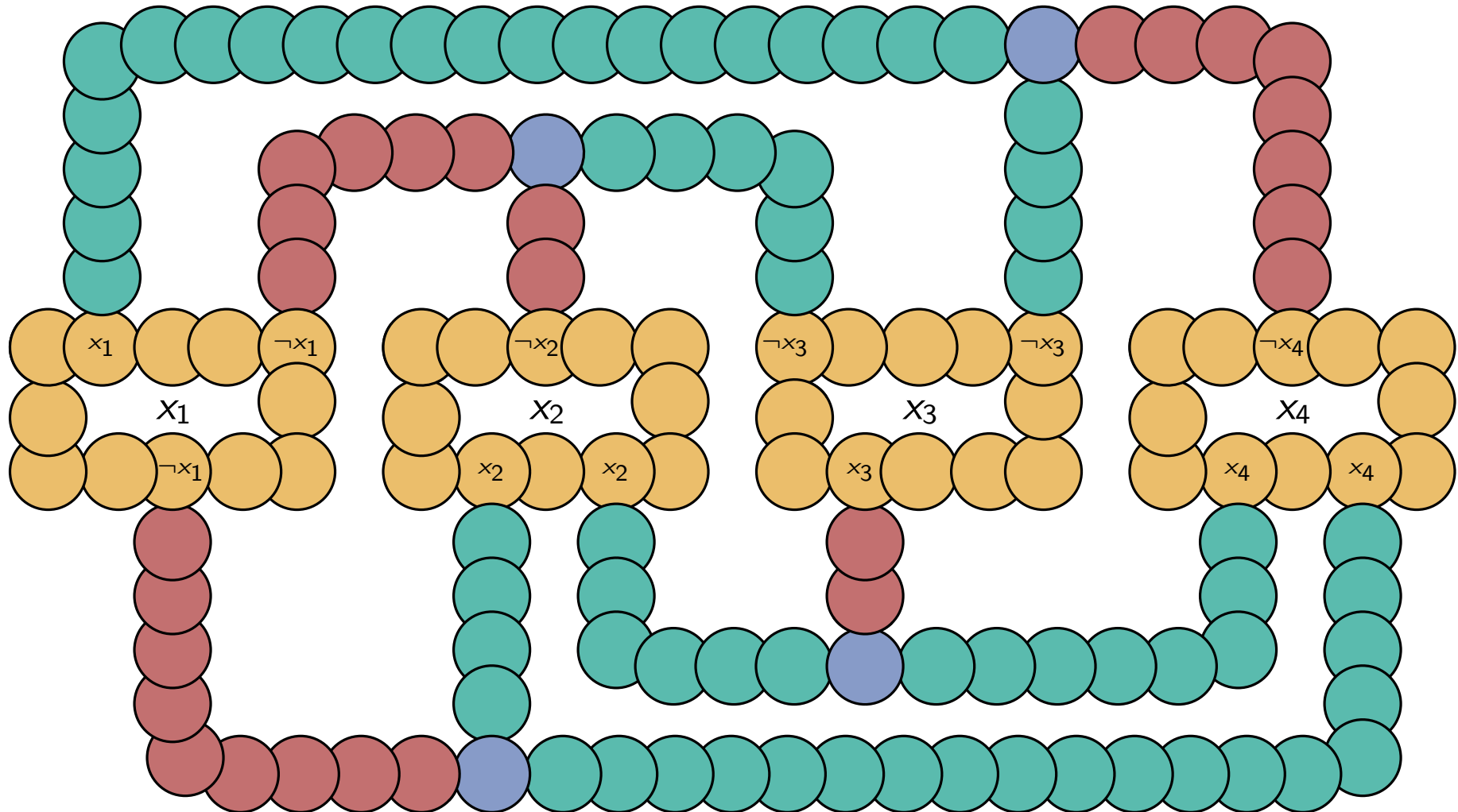
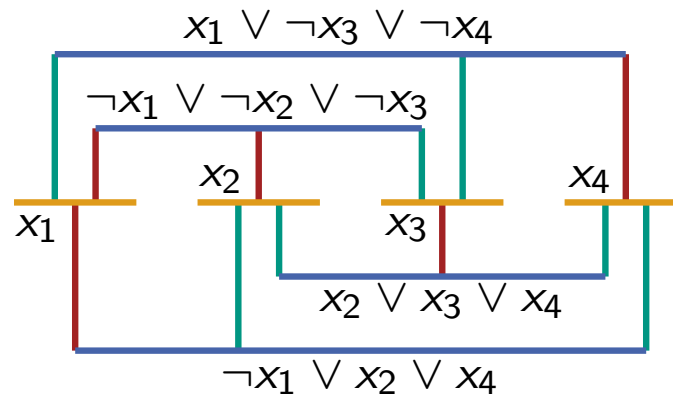
- $C$  muss mindestens eine der drei Nachbardisks überlappen
- das entspricht mindestens einem wahren Literal



# Alles auf einmal



# Alles auf einmal



# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

**Details der Reduktion** (sollte im großen und ganzen mehr oder weniger klar sein)

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

**Details der Reduktion** (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

## Details der Reduktion (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln
- Länge der Transportgadgets
  - ergibt sich aus Kantenlängen in rechtwinkliger Zeichnung der Eingabe

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

## Details der Reduktion (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln
- Länge der Transportgadgets
  - ergibt sich aus Kantenlängen in rechtwinkliger Zeichnung der Eingabe
  - Problem: Anzahl Disks ist ganzzahlig  $\Rightarrow$  ggf. nicht jede Länge möglich



# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

## Details der Reduktion (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln
- Länge der Transportgadgets
  - ergibt sich aus Kantenlängen in rechtwinkliger Zeichnung der Eingabe
  - Problem: Anzahl Disks ist ganzzahlig  $\Rightarrow$  ggf. nicht jede Länge möglich
  - Gadget funktioniert, falls Überlappung zweier Disks  $> 1/8$  und  $\leq 1/4$

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

## Details der Reduktion (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln
- Länge der Transportgadgets
  - ergibt sich aus Kantenlängen in rechtwinkliger Zeichnung der Eingabe
  - Problem: Anzahl Disks ist ganzzahlig  $\Rightarrow$  ggf. nicht jede Länge möglich
  - Gadget funktioniert, falls Überlappung zweier Disks  $> 1/8$  und  $\leq 1/4$
  - sind die Kanten ausreichend lang, dann haben wir genug Spielraum

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

## Details der Reduktion (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln
- Länge der Transportgadgets
  - ergibt sich aus Kantenlängen in rechtwinkliger Zeichnung der Eingabe
  - Problem: Anzahl Disks ist ganzzahlig  $\Rightarrow$  ggf. nicht jede Länge möglich
  - Gadget funktioniert, falls Überlappung zweier Disks  $> 1/8$  und  $\leq 1/4$
  - sind die Kanten ausreichend lang, dann haben wir genug Spielraum
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit

# Was müssen wir genau zeigen?

## Theorem

Es ist NP-schwer zu entscheiden, ob eine physikalisch realisierbare Anordnung existiert, die von jeder Disk mindestens  $3/4$  des Rands zeigt.

## Details der Reduktion (sollte im großen und ganzen mehr oder weniger klar sein)

- Größe des Variablen-Gadgets: abh. von Anzahl Vorkommen in Klauseln
- Länge der Transportgadgets
  - ergibt sich aus Kantenlängen in rechtwinkliger Zeichnung der Eingabe
  - Problem: Anzahl Disks ist ganzzahlig  $\Rightarrow$  ggf. nicht jede Länge möglich
  - Gadget funktioniert, falls Überlappung zweier Disks  $> 1/8$  und  $\leq 1/4$
  - sind die Kanten ausreichend lang, dann haben wir genug Spielraum
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit

## Korrektheit

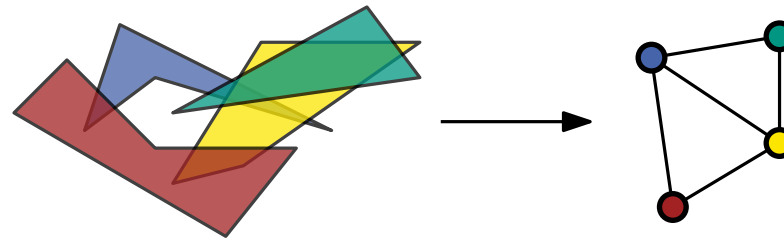
- $3/4$  des Rands sichtbar  $\Rightarrow$  Formel erfüllbar
- Formel erfüllbar  $\Rightarrow 3/4$  des Rands sichtbar

Warum?

# Unit Disk Graphs

## Definition

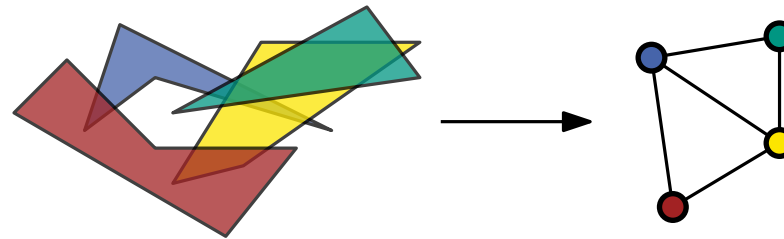
Eine Menge geometrischer Objekte  $V$  definiert den **geometrischen Schnittgraphen**  $G = (V, E)$  mit  $uv \in E \Leftrightarrow u \cap v \neq \emptyset$ .



# Unit Disk Graphs

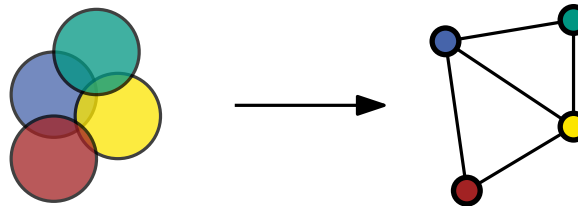
## Definition

Eine Menge geometrischer Objekte  $V$  definiert den **geometrischen Schnittgraphen**  $G = (V, E)$  mit  $uv \in E \Leftrightarrow u \cap v \neq \emptyset$ .



## Definition

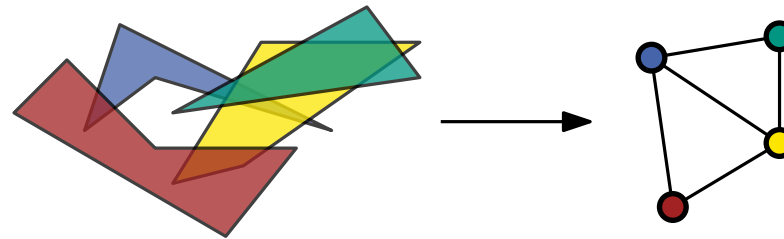
Ein Graph ist ein **Unit Disk Graph**, wenn er der geometrische Schnittgraph einer Menge von Disks mit Radius 1 ist.



# Unit Disk Graphs

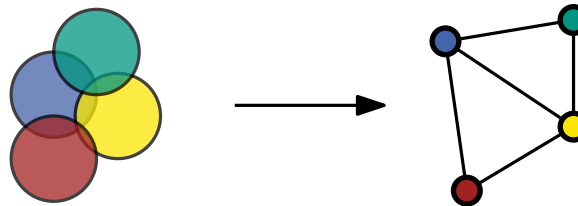
## Definition

Eine Menge geometrischer Objekte  $V$  definiert den **geometrischen Schnittgraphen**  $G = (V, E)$  mit  $uv \in E \Leftrightarrow u \cap v \neq \emptyset$ .



## Definition

Ein Graph ist ein **Unit Disk Graph**, wenn er der geometrische Schnittgraph einer Menge von Disks mit Radius 1 ist.



## Problem: Erkennung

Ist ein gegebener Graph ein Unit Disk Graph?

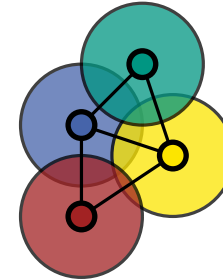
# Grundlegende Beobachtungen

## Ziel im Folgenden

- zeige, dass die Erkennung NP-schwer ist
- Reduktion von planar monotone 3-Sat

## Grundlage für die Reduktion

- äquivalente Fragestellung: Gibt es Positionen für die Knoten, sodass  $\text{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$ ?





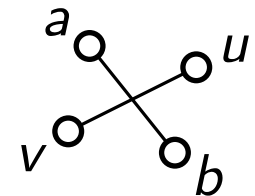
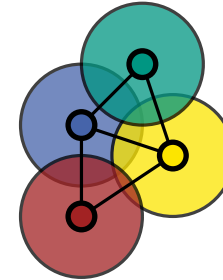
# Grundlegende Beobachtungen

## Ziel im Folgenden

- zeige, dass die Erkennung NP-schwer ist
- Reduktion von planar monotone 3-Sat

## Grundlage für die Reduktion

- äquivalente Fragestellung: Gibt es Positionen für die Knoten, sodass  $\text{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$ ?
- wenn sich zwei Kanten  $ab$  und  $uv$  in dieser Darstellung kreuzen, dann bilden drei der Knoten  $a, b, u, v$  ein Dreieck



Warum?

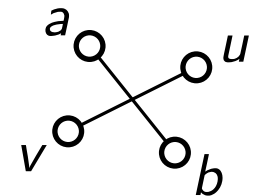
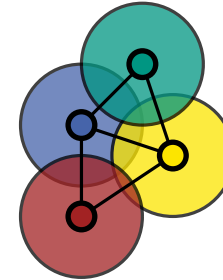
# Grundlegende Beobachtungen

## Ziel im Folgenden

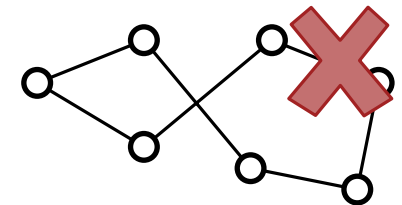
- zeige, dass die Erkennung NP-schwer ist
- Reduktion von planar monotone 3-Sat

## Grundlage für die Reduktion

- äquivalente Fragestellung: Gibt es Positionen für die Knoten, sodass  $\text{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$ ?
- wenn sich zwei Kanten  $ab$  und  $uv$  in dieser Darstellung kreuzen, dann bilden drei der Knoten  $a, b, u, v$  ein Dreieck
- induzierte Kreise sind kreuzungsfrei



Warum?



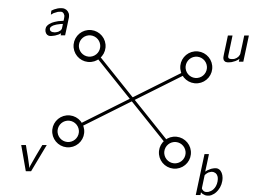
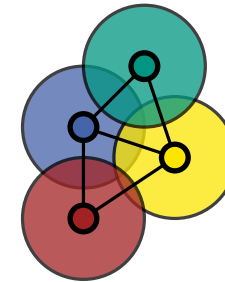
# Grundlegende Beobachtungen

## Ziel im Folgenden

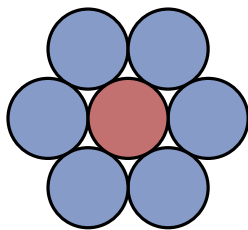
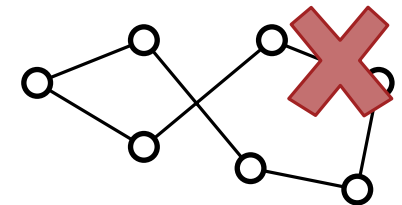
- zeige, dass die Erkennung NP-schwer ist
- Reduktion von planar monotone 3-Sat

## Grundlage für die Reduktion

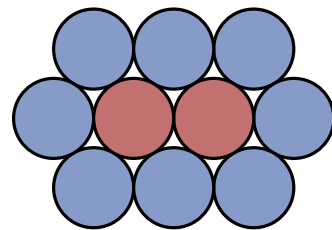
- äquivalente Fragestellung: Gibt es Positionen für die Knoten, sodass  $\text{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$ ?
- wenn sich zwei Kanten  $ab$  und  $uv$  in dieser Darstellung kreuzen, dann bilden drei der Knoten  $a, b, u, v$  ein Dreieck
- induzierte Kreise sind kreuzungsfrei
- Kreise können nur begrenzt viele unabhängige Knoten enthalten ( $i$ -Käfig enthält maximal  $i$  unabhängige Knoten)



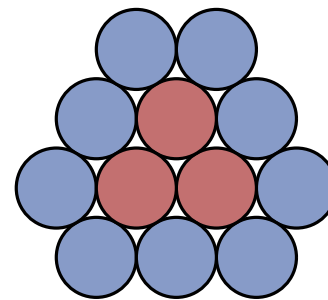
Warum?



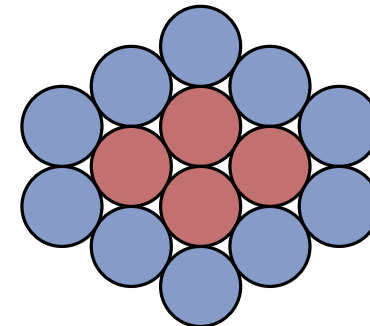
0-Käfig



1-Käfig



2-Käfig



3-Käfig

# Gadgets

## Was brauchen wir?

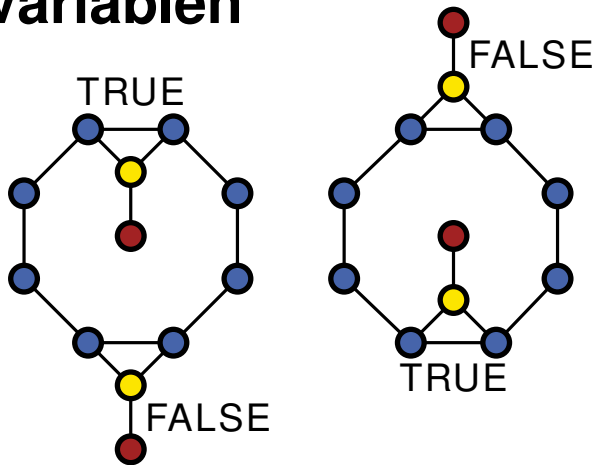
- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

# Gadgets

## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

## Variablen

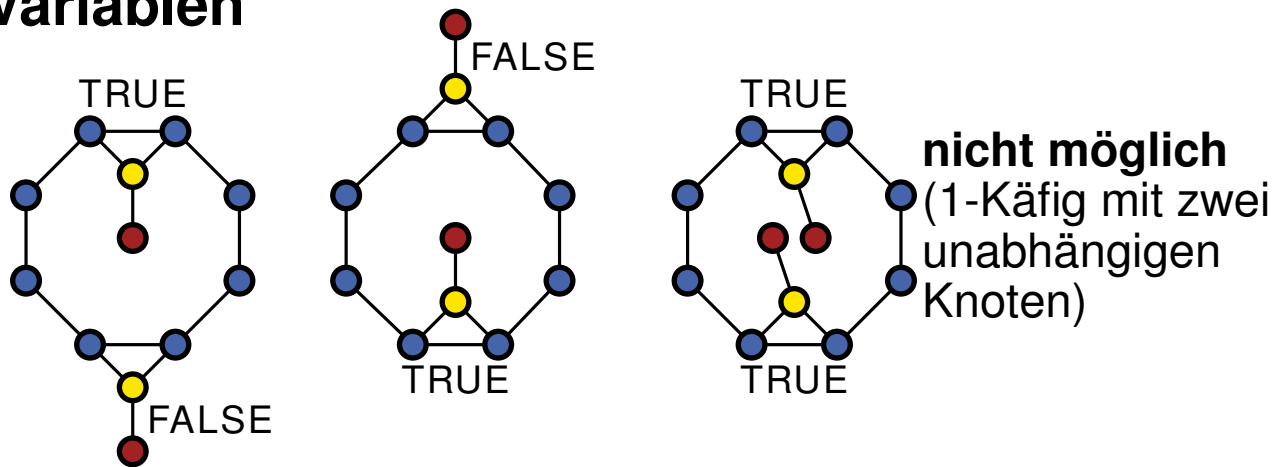


# Gadgets

## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

## Variablen

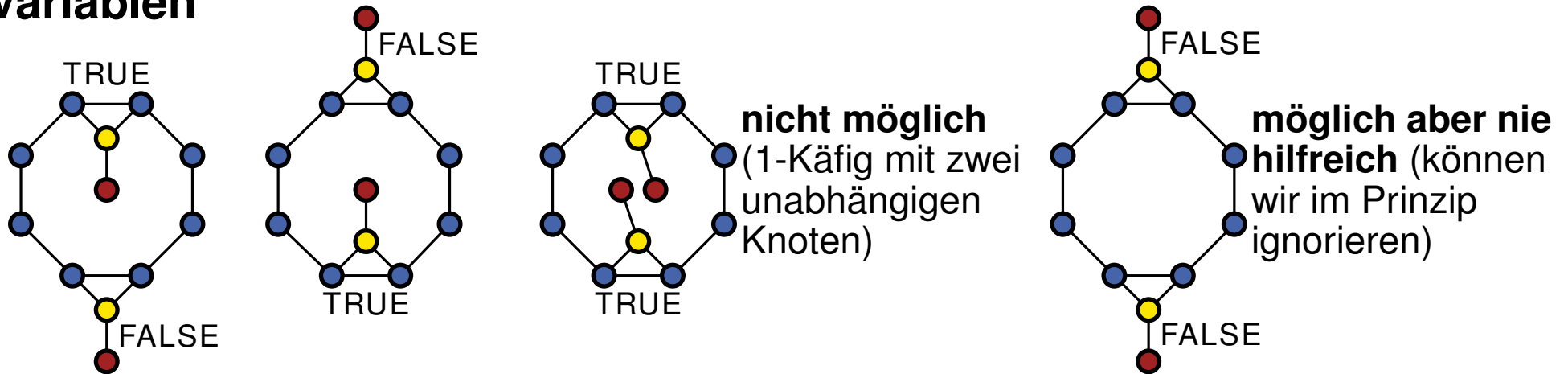


# Gadgets

## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

## Variablen

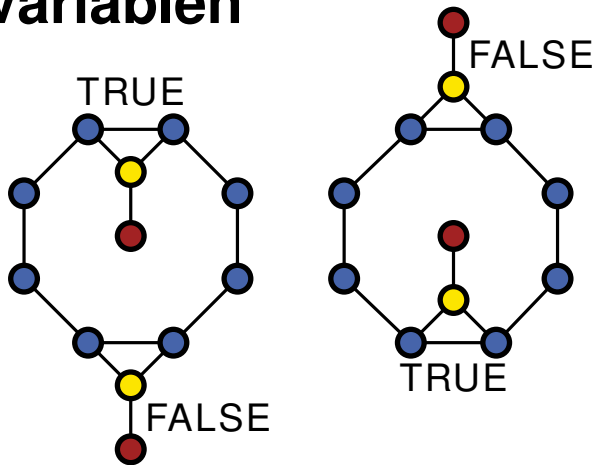


# Gadgets

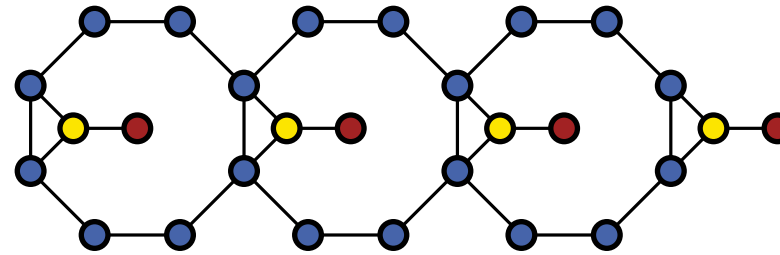
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

## Variablen



## Informationstransport



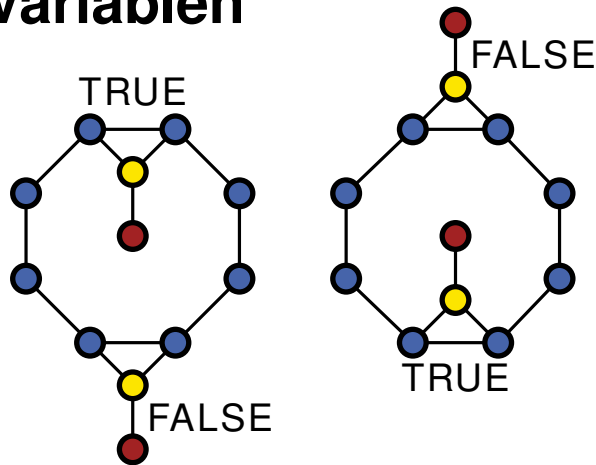


# Gadgets

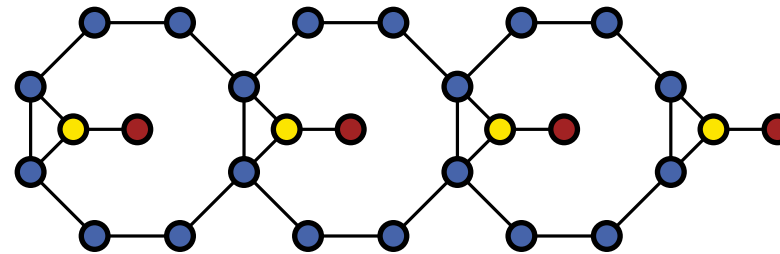
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

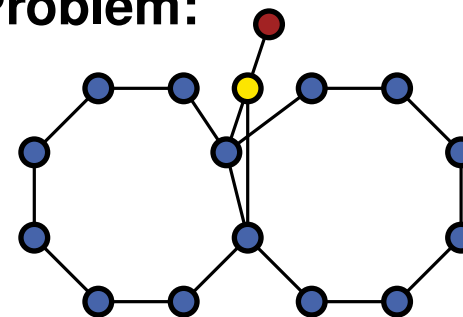
## Variablen



## Informationstransport



## Problem:

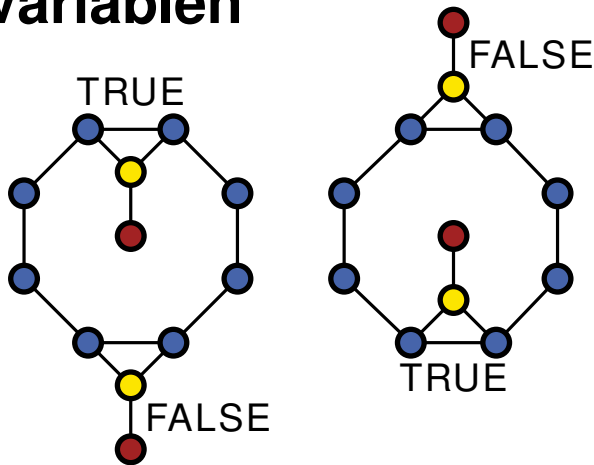


# Gadgets

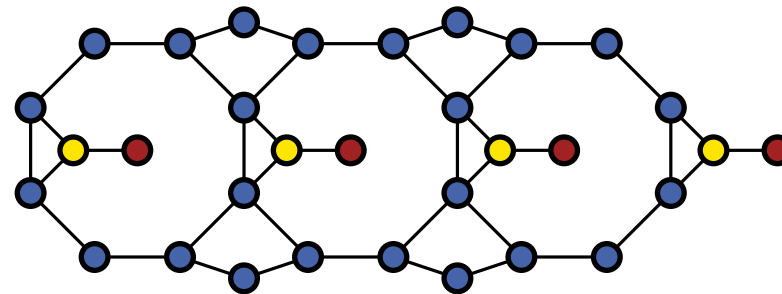
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

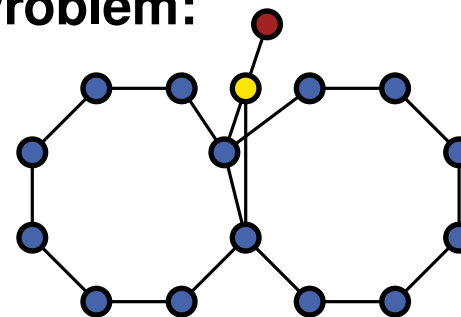
## Variablen



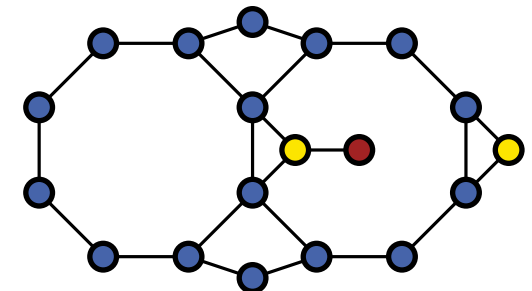
## Informationstransport



## Problem:



## Lösung:

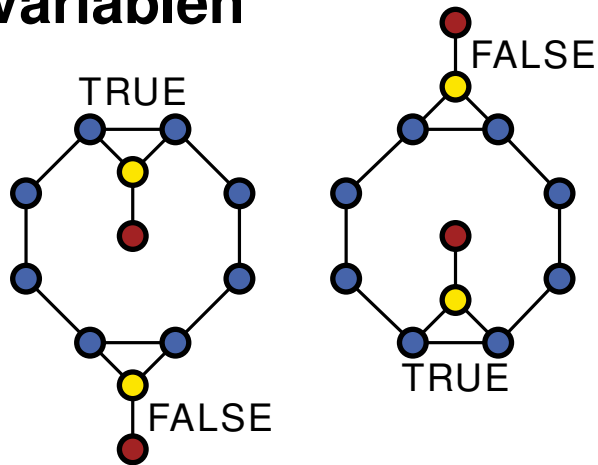


# Gadgets

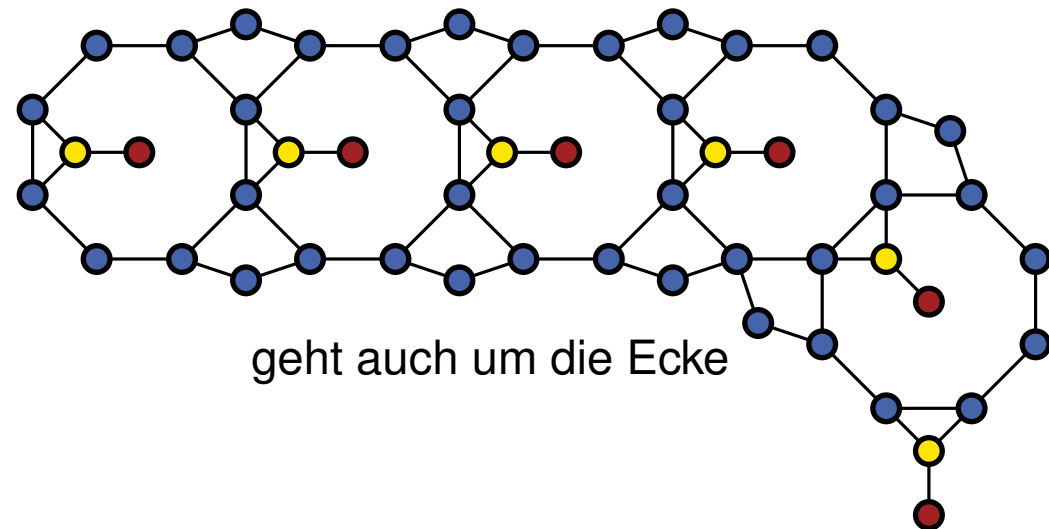
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

## Variablen



## Informationstransport

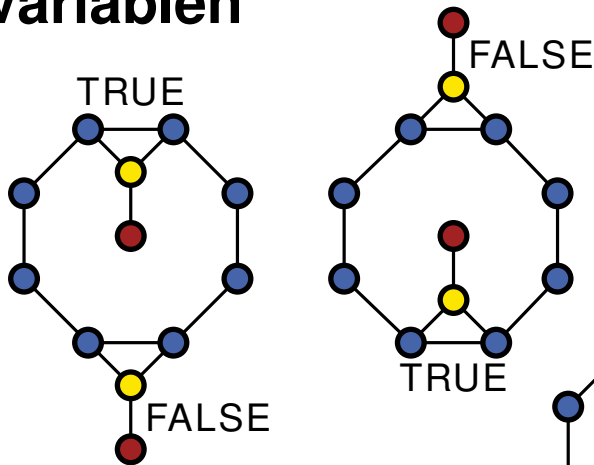


# Gadgets

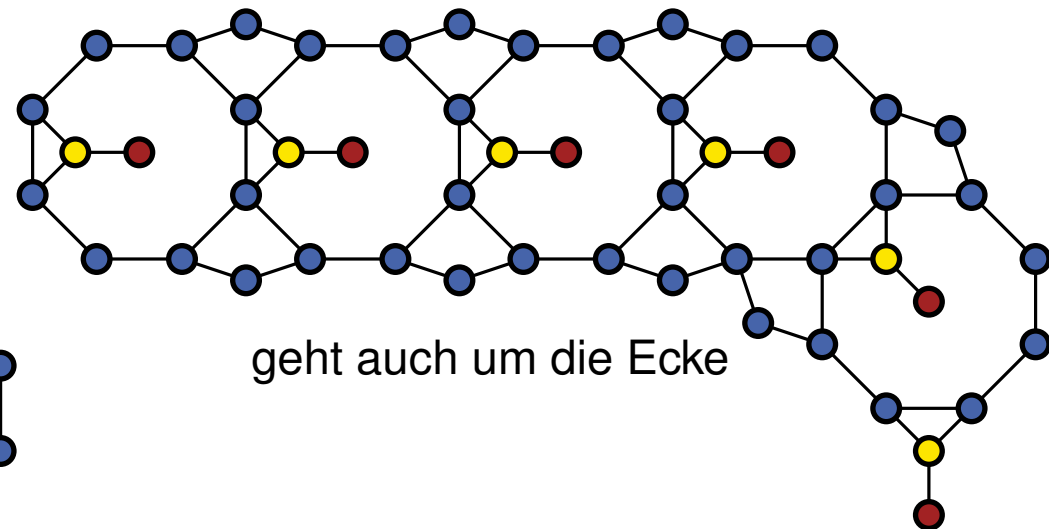
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

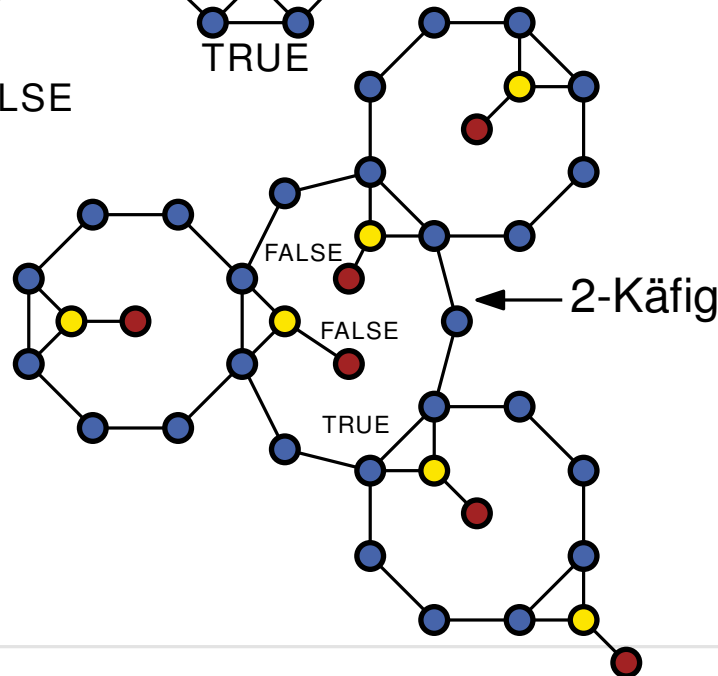
### Variablen



### Informationstransport



### Klausel

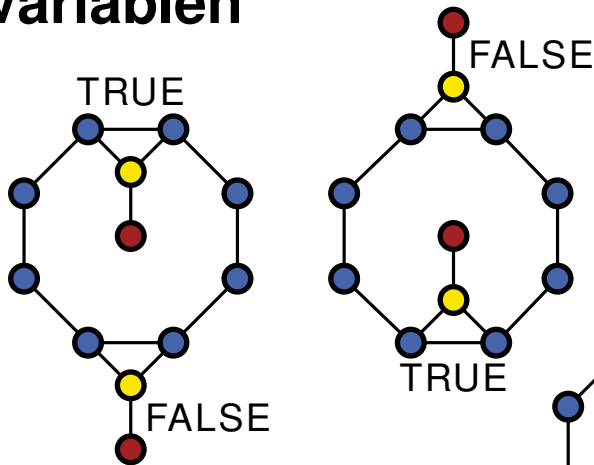


# Gadgets

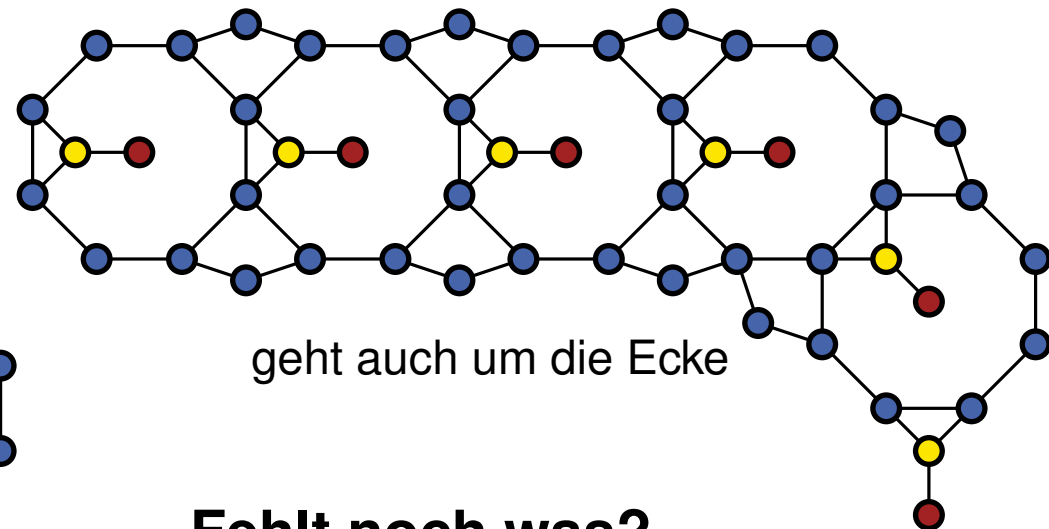
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

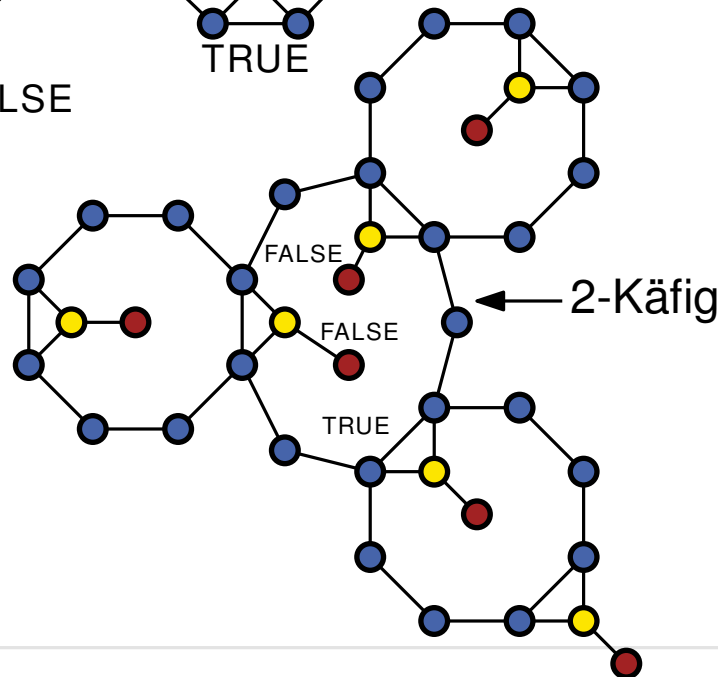
### Variablen



### Informationstransport



### Klausel



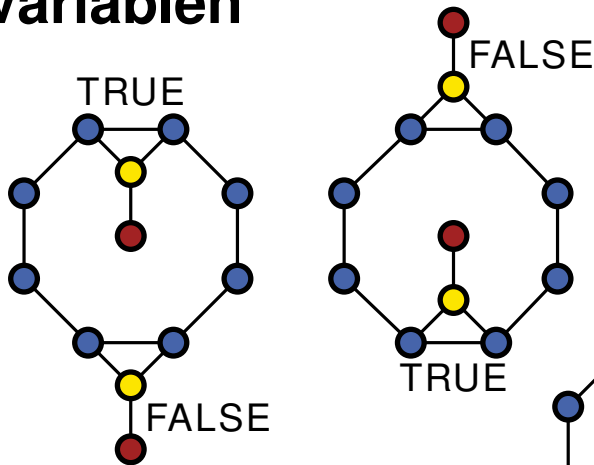
Fehlt noch was?

# Gadgets

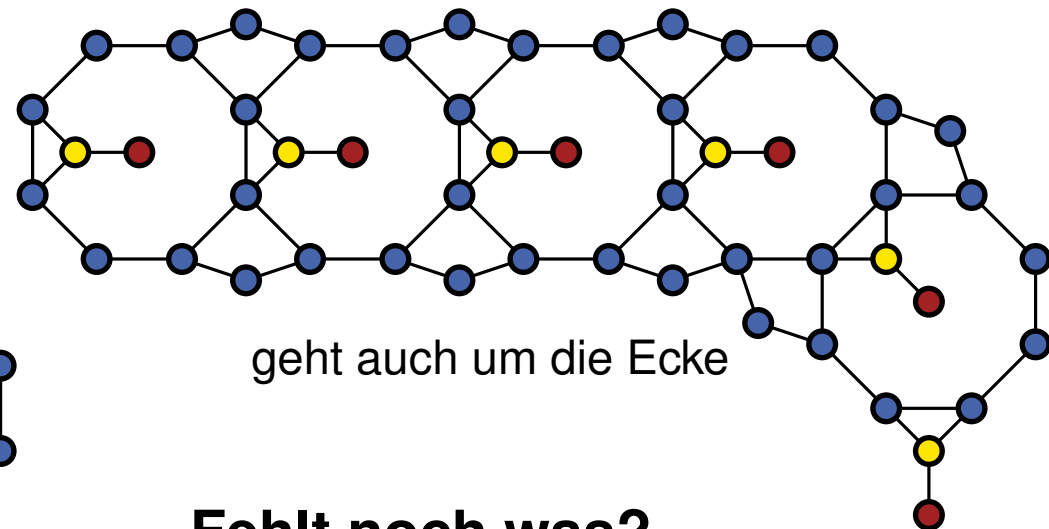
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

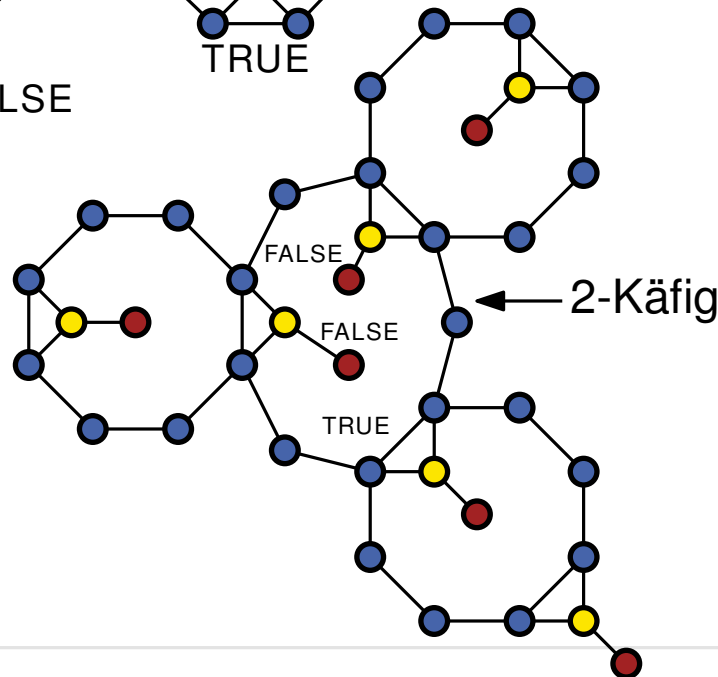
### Variablen



### Informationstransport



### Klausel



### Fehlt noch was?

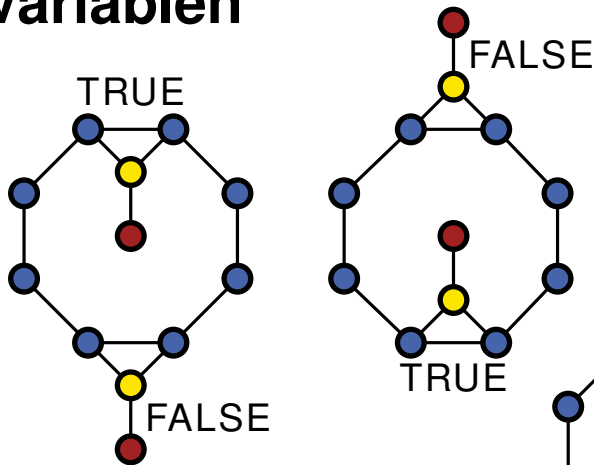
- wir können Entscheidung der Variable nur zu einer Klausel bringen
- Variablen sind aber in mehreren Klauseln enthalten

# Gadgets

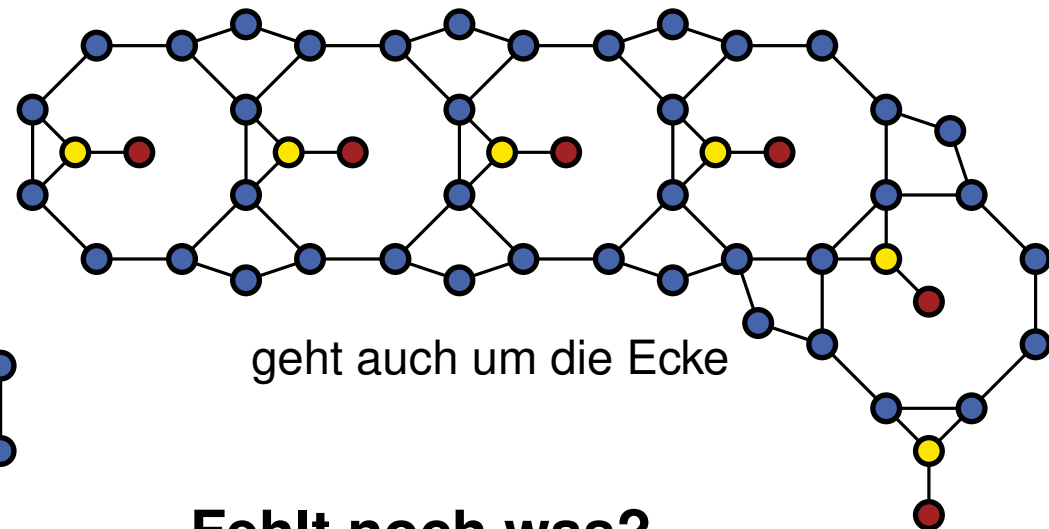
## Was brauchen wir?

- Variablen, Klauseln
- Kanäle für Informationstransport der Literale von Variablen zu Klauseln

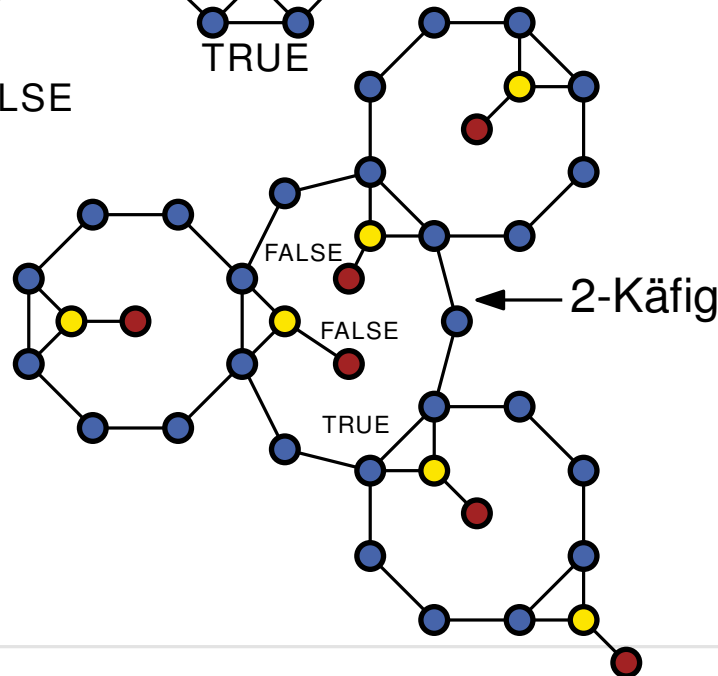
### Variablen



### Informationstransport



### Klausel



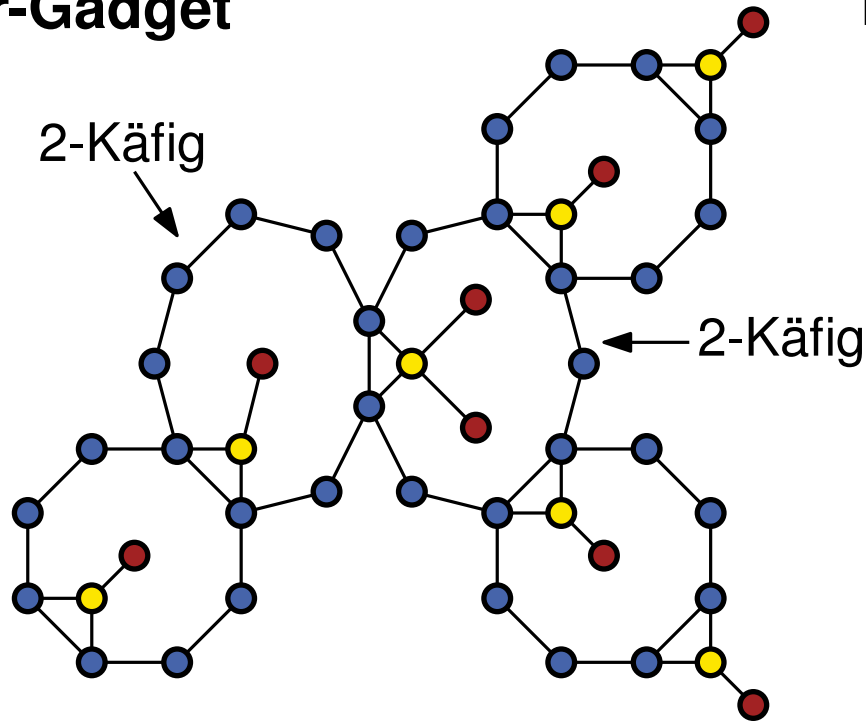
### Fehlt noch was?

- wir können Entscheidung der Variable nur zu einer Klausel bringen
- Variablen sind aber in mehreren Klauseln enthalten
- wir brauchen ein Splitter-Gadget

# Vervielfältigung von Information

Splitter-Gadget

Klausel



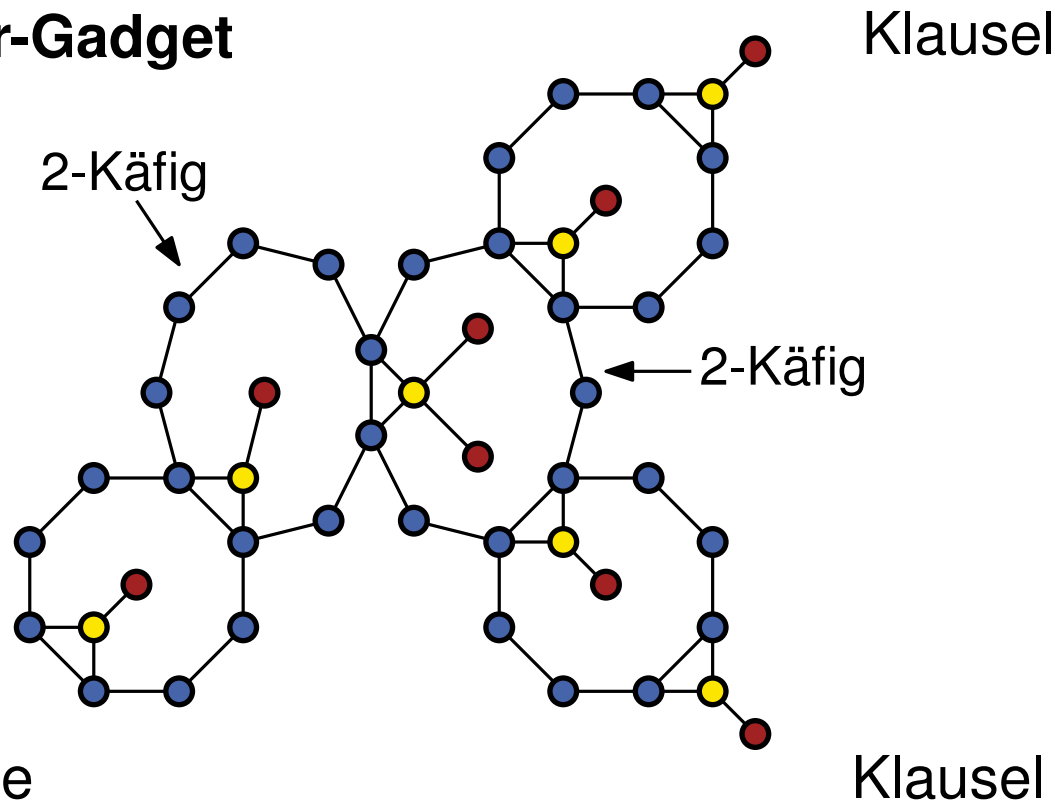
Variable

Klausel



# Vervielfältigung von Information

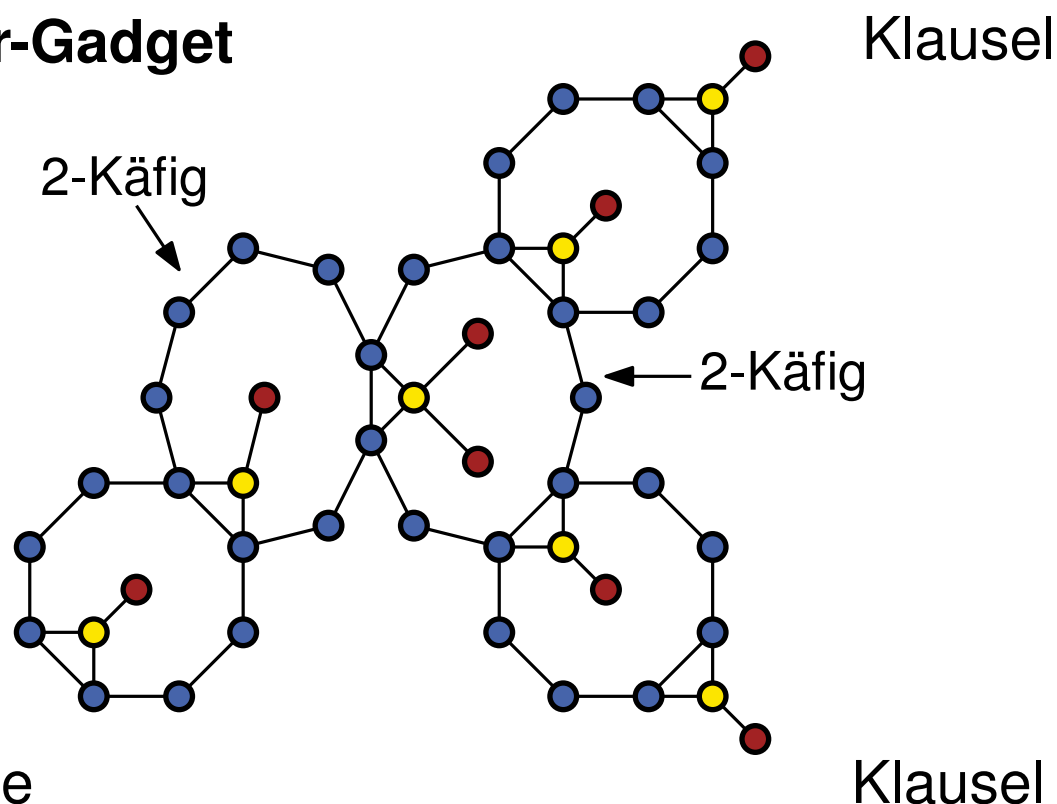
## Splitter-Gadget



- kommt von der Variable die FALSE-Konfiguration, so gehen zu den Klauseln FALSE-Konfigurationen weiter (in jeder Unit-Disk Zeichnung)

# Vervielfältigung von Information

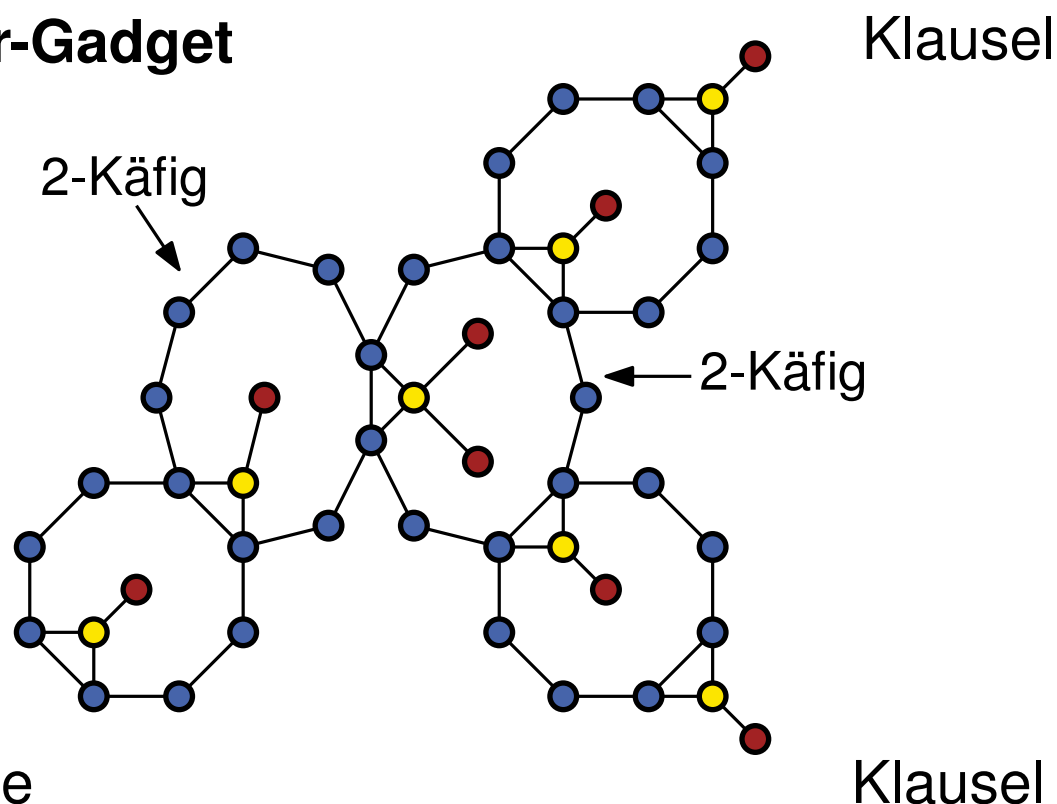
## Splitter-Gadget



- kommt von der Variable die FALSE-Konfiguration, so gehen zu den Klauseln FALSE-Konfigurationen weiter (in jeder Unit-Disk Zeichnung)
- bei einkommender TRUE-Konfiguration kann man ausgehende TRUE-Konfiguration erhalten

# Vervielfältigung von Information

## Splitter-Gadget



- kommt von der Variable die FALSE-Konfiguration, so gehen zu den Klauseln FALSE-Konfigurationen weiter (in jeder Unit-Disk Zeichnung)
- bei einkommender TRUE-Konfiguration kann man ausgehende TRUE-Konfiguration erhalten
- Gadget tut also was es soll (ein Flip von TRUE auf FALSE ist OK)

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

**Was müssen wir für den Beweis noch beachten?**

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit
- Korrektheit



# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit
- Korrektheit
  - Graph ist Unit Disk Graph  $\Rightarrow$  Formel erfüllbar
  - Formel erfüllbar  $\Rightarrow$  Graph ist Unit Disk Graph

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit
- Korrektheit
  - Graph ist Unit Disk Graph  $\Rightarrow$  Formel erfüllbar
  - Formel erfüllbar  $\Rightarrow$  Graph ist Unit Disk Graph

## Ist das Problem auch NP-Vollständig?

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit
- Korrektheit
  - Graph ist Unit Disk Graph  $\Rightarrow$  Formel erfüllbar
  - Formel erfüllbar  $\Rightarrow$  Graph ist Unit Disk Graph

## Ist das Problem auch NP-Vollständig?

- vermutlich nicht
- Was geht schief?

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit
- Korrektheit
  - Graph ist Unit Disk Graph  $\Rightarrow$  Formel erfüllbar
  - Formel erfüllbar  $\Rightarrow$  Graph ist Unit Disk Graph

## Ist das Problem auch NP-Vollständig?

- vermutlich nicht
- Was geht schief?
  - rate Positionen als Zertifikat und überprüfen ob  $uv \in E \Leftrightarrow \text{dist}(u, v) \leq 2$

# Schwer zu erkennende Graphen

## Theorem

Es ist NP-schwer zu entscheiden, ob ein Graph ein Unit Disk Graph ist.

## Was müssen wir für den Beweis noch beachten?

- Länge der Transportgadgets: ergibt sich wieder durch Kantenlängen in der rectilinearen Zeichnung der Eingabe
- resultierende Instanz ist polynomiell groß
- Reduktion läuft in polynomieller Zeit
- Korrektheit
  - Graph ist Unit Disk Graph  $\Rightarrow$  Formel erfüllbar
  - Formel erfüllbar  $\Rightarrow$  Graph ist Unit Disk Graph

## Ist das Problem auch NP-Vollständig?

- vermutlich nicht
- Was geht schief?
  - rate Positionen als Zertifikat und überprüfen ob  $uv \in E \Leftrightarrow \text{dist}(u, v) \leq 2$
  - Problem: es könnte sein, dass es kein polynomiell großes Zertifikat gibt (weil wir große bzw. genau aufgelöste Zahlen für die Koordinaten brauchen)

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

## Die Komplexitätsklasse $\exists\mathbb{R}$

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar  
(maximal so schwer)

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

## Die Komplexitätsklasse $\exists\mathbb{R}$

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar  
(maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind  
(mindestens so schwer)



# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar  
(maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind  
(mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer  
(genauso schwer)

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar (maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind (mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer (genauso schwer)

## Unit Disk Graphen

- Erkennung von Unit Disk Graphen liegt in  $\exists\mathbb{R}$  **Warum?**

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar (maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind (mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer (genauso schwer)

## Unit Disk Graphen

- Erkennung von Unit Disk Graphen liegt in  $\exists\mathbb{R}$  **Warum?**
- tatsächlich ist es sogar  $\exists\mathbb{R}$ -vollständig (ohne Beweis)

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar (maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind (mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer (genauso schwer)

## Unit Disk Graphen

- Erkennung von Unit Disk Graphen liegt in  $\exists\mathbb{R}$  **Warum?**
- tatsächlich ist es sogar  $\exists\mathbb{R}$ -vollständig (ohne Beweis)

## Sonstige Einordnung

- $NP \subseteq \exists\mathbb{R}$  **Warum?**

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar (maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind (mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer (genauso schwer)

## Unit Disk Graphen

- Erkennung von Unit Disk Graphen liegt in  $\exists\mathbb{R}$  **Warum?**
- tatsächlich ist es sogar  $\exists\mathbb{R}$ -vollständig (ohne Beweis)

## Sonstige Einordnung

- $NP \subseteq \exists\mathbb{R}$  **Warum?**
- $\exists\mathbb{R} \subseteq PSPACE$  (ohne Beweis)

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar (maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind (mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer (genauso schwer)

## Unit Disk Graphen

- Erkennung von Unit Disk Graphen liegt in  $\exists\mathbb{R}$  **Warum?**
- tatsächlich ist es sogar  $\exists\mathbb{R}$ -vollständig (ohne Beweis)

## Sonstige Einordnung

- $NP \subseteq \exists\mathbb{R}$  **Warum?**
- $\exists\mathbb{R} \subseteq PSPACE$  (ohne Beweis)
- Vermutung  $NP \neq \exists\mathbb{R}$

# Existential theory of the reals

## Problem: Existential theory of the reals

Sei  $F(X_1, \dots, X_n)$  eine quantorenfreie Boolesche Formel über (Un-)gleichungen von reellen Polynomen. Ist  $\exists X_1 \dots \exists X_n F(X_1, \dots, X_n)$  wahr?

(In der Logik ist eine *Theorie* eine Menge von Aussagen. Die *existential theory of the reals* ist die Menge aller wahren Aussagen dieser Form.)

## Die Komplexitätsklasse $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$  ist polynomiell auf existential theory of the reals reduzierbar (maximal so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -schwer, wenn alle Probleme aus  $\exists\mathbb{R}$  polynomiell auf  $\Pi$  reduzierbar sind (mindestens so schwer)
- $\Pi$  ist  $\exists\mathbb{R}$ -vollständig, wenn  $\Pi \in \exists\mathbb{R}$  und  $\Pi$   $\exists\mathbb{R}$ -schwer (genauso schwer)

## Unit Disk Graphen

- Erkennung von Unit Disk Graphen liegt in  $\exists\mathbb{R}$  **Warum?**
- tatsächlich ist es sogar  $\exists\mathbb{R}$ -vollständig (ohne Beweis)

## Sonstige Einordnung

- $NP \subseteq \exists\mathbb{R}$  **Warum?**
- $\exists\mathbb{R} \subseteq PSPACE$  (ohne Beweis)
- Vermutung  $NP \neq \exists\mathbb{R}$

**Bemerkung:** die Erkennung von Unit-Disk Graphen ist also vermutlich echt schwerer als jedes Problem in NP

# Zusammenfassung

## Heute gesehen

- Problem aus der Kartographie
- Unit Disk Graphen
- Komplexitätsklasse  $\exists\mathbb{R}$
- geometrische Probleme liegen regelmäßig nicht in NP



# Zusammenfassung

## Heute gesehen

- Problem aus der Kartographie
- Unit Disk Graphen
- Komplexitätsklasse  $\exists\mathbb{R}$
- geometrische Probleme liegen regelmäßig nicht in NP
- Reduktionen von Sat-Varianten oft gar nicht schwer; man braucht nur:

# Zusammenfassung

## Heute gesehen

- Problem aus der Kartographie
- Unit Disk Graphen
- Komplexitätsklasse  $\exists\mathbb{R}$
- geometrische Probleme liegen regelmäßig nicht in NP
- Reduktionen von Sat-Varianten oft gar nicht schwer; man braucht nur:
  - Variablen-Gadget
  - Klausel-Gadget
  - Transport-Gadget

# Zusammenfassung

## Heute gesehen

- Problem aus der Kartographie
- Unit Disk Graphen
- Komplexitätsklasse  $\exists \mathbb{R}$
- geometrische Probleme liegen regelmäßig nicht in NP
- Reduktionen von Sat-Varianten oft gar nicht schwer; man braucht nur:
  - Variablen-Gadget
  - Klausel-Gadget
  - Transport-Gadget
  - ggf. Split-Gadget (wenn das Variablen-Gadget zu wenige Literale produziert)

# Zusammenfassung

## Heute gesehen

- Problem aus der Kartographie
- Unit Disk Graphen
- Komplexitätsklasse  $\exists \mathbb{R}$
- geometrische Probleme liegen regelmäßig nicht in NP
- Reduktionen von Sat-Varianten oft gar nicht schwer; man braucht nur:
  - Variablen-Gadget
  - Klausel-Gadget
  - Transport-Gadget
  - ggf. Split-Gadget (wenn das Variablen-Gadget zu wenige Literale produziert)

## Was gibt es sonst noch?

- Negations-Gadget (wenn man an einer Stelle, an der man negative Literale braucht, nur positive Literale abgreifen kann)

# Zusammenfassung

## Heute gesehen

- Problem aus der Kartographie
- Unit Disk Graphen
- Komplexitätsklasse  $\exists \mathbb{R}$
- geometrische Probleme liegen regelmäßig nicht in NP
- Reduktionen von Sat-Varianten oft gar nicht schwer; man braucht nur:
  - Variablen-Gadget
  - Klausel-Gadget
  - Transport-Gadget
  - ggf. Split-Gadget (wenn das Variablen-Gadget zu wenige Literale produziert)

## Was gibt es sonst noch?

- Negations-Gadget (wenn man an einer Stelle, an der man negative Literale braucht, nur positive Literale abgreifen kann)
- Kreuzungs-Gadget (wenn man von einer nicht-planaren Sat-Variante reduziert)

# Literaturhinweise

- **Algorithmic Aspects of Proportional Symbol Maps** (2009)  
Sergio Cabello, Herman Haverkort, Marc van Kreveld, Bettina Speckmann  
<https://doi.org/10.1007/s00453-009-9281-8>
- **Unit disk graph recognition is NP-hard** (1998)  
Heinz Breu, David G. Kirkpatrick  
[https://doi.org/10.1016/S0925-7721\(97\)00014-X](https://doi.org/10.1016/S0925-7721(97)00014-X)
- **Optimal Binary Space Partitions in the Plane** (2010)  
Mark de Berg, Amirali Khosravi  
(NP-Schwere für monotone planar 3-Sat) [https://doi.org/10.1007/978-3-642-14031-0\\_25](https://doi.org/10.1007/978-3-642-14031-0_25)
- **Sphere and Dot Product Representations of Graphs** (2012)  
Ross J. Kang, Tobias Müller  
( $\exists\mathbb{R}$ -Schwere für die Erkennung von Unit Disk Graphen) <https://doi.org/10.1007/s00454-012-9394-8>