

Prof. Dr.-Ing. Dr. h. c. J. Becker

becker@kit.edu

Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

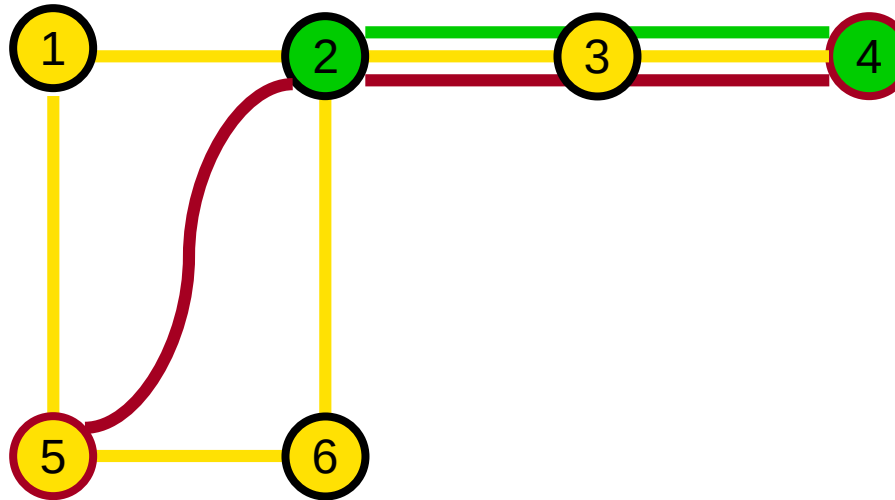
Digitaltechnik

Mathematische Grundlagen - Graphen -

Was ist ein Graph?

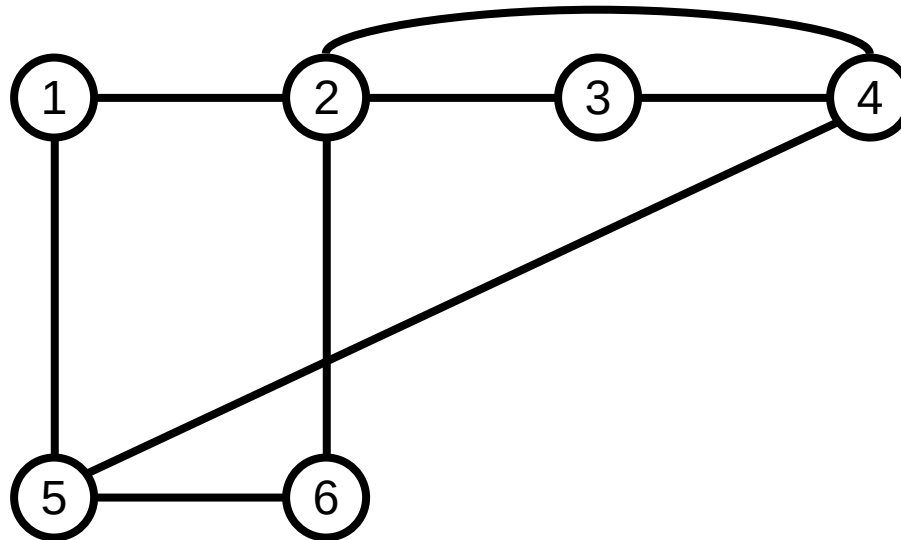
- **Graphen** dienen zur **Abstraktion** von Problemen
 - abstrakte **Darstellung** von **Zusammenhängen**
 - **verbundene Objekte** (→ Relationen zwischen Objekten)
- Bei Spielen wie Scotland Yard kommt es beispielsweise für die Strategie nicht darauf an, wie die Wege verlaufen, sondern nur welche Punkte wie miteinander verbunden sind
- **Graphen** stellen das **Problem** durch **Knoten** und **Kanten** dar
- Ein **Graph** muss mindestens einen **Knoten** besitzen
- **Kanten** können **je zwei Knoten** verbinden

Problemerkfassung:



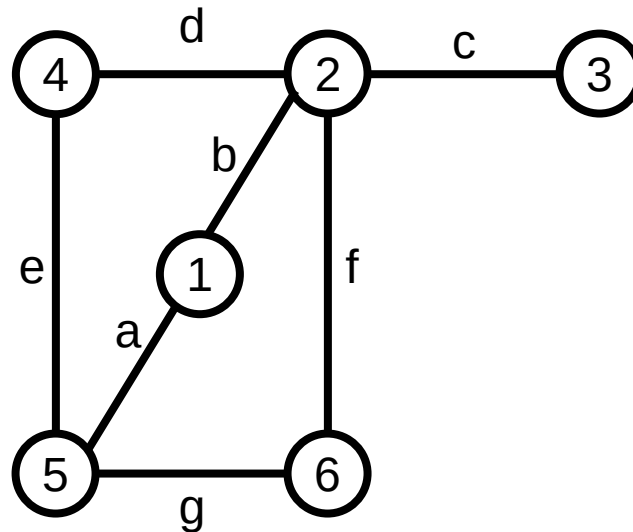
- Darstellung der **Zusammenhänge** und **Abhängigkeitsbeziehungen**
- Hervorheben wesentlicher Beziehungen

Repräsentation durch einen Graphen:



- **Abstraktion** der **Zusammenhänge** und **Abhängigkeitsbeziehungen**
- Repräsentation der Beziehungen durch direkte Kantenverbindungen

Graph ohne Überschneidungen der Kanten:



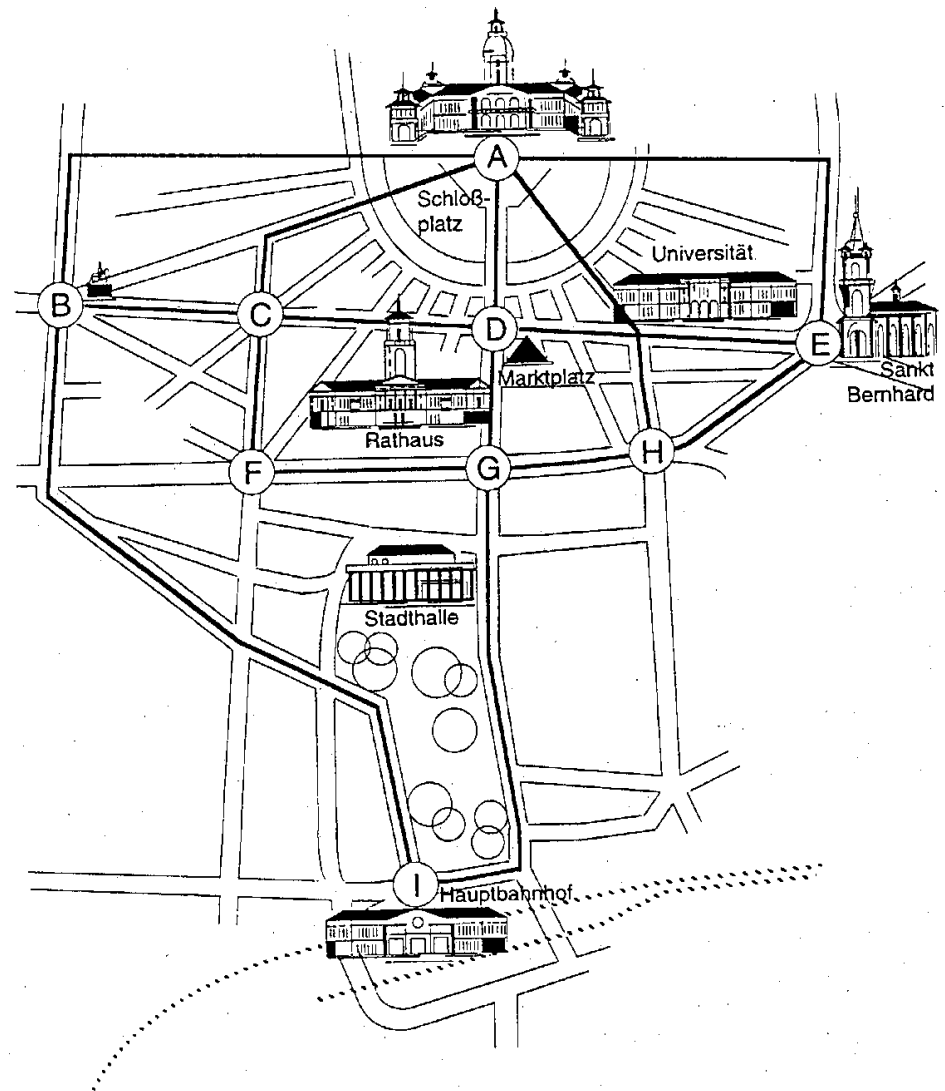
- Weitere **Vereinfachung/Optimierung** der Abhängigkeitsbeziehungen
→ ohne **kreuzende Verbindungen**
- Einfügen von Kantenbezeichnern

Beispiel:

■ Darstellung von Verkehrsbeziehungen im Karlsruher Stadtplan

→ wichtige Punkte

→ wesentliche Straßenverbindungen

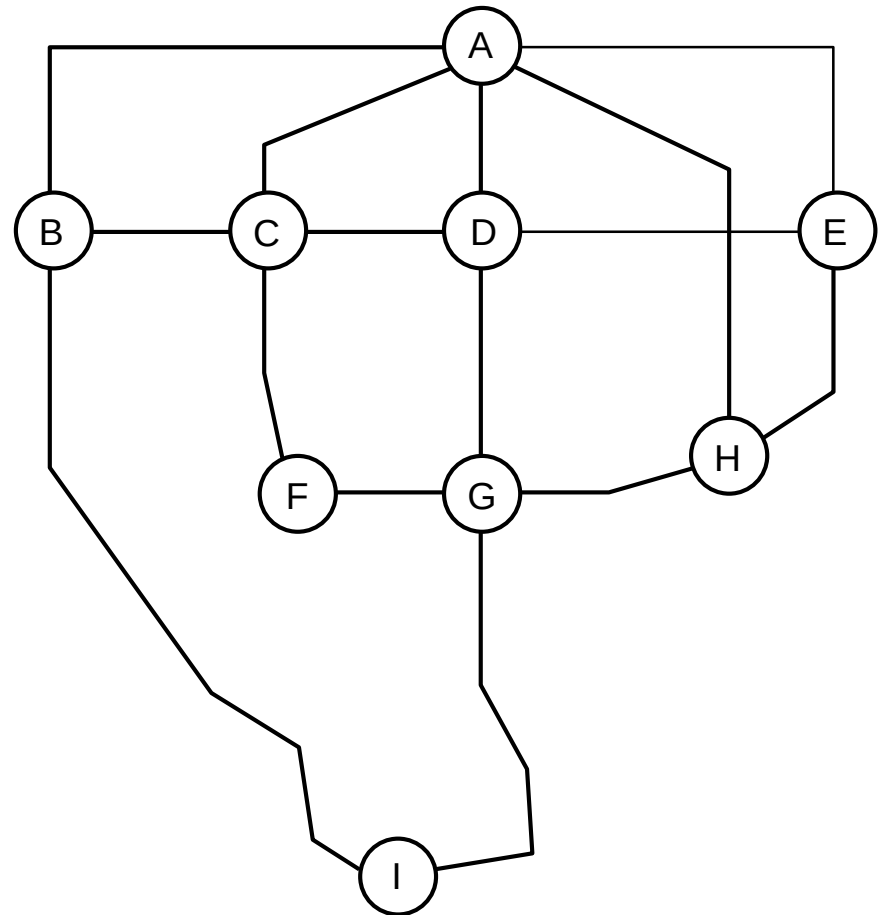


Beispiel:

- Abstrahierte Darstellung der Verkehrsbeziehungen

→ Weglassen von Details:
Bottom-up Abstraktion

→ **exakte Geometrie**
der Verbindungen
nicht relevant

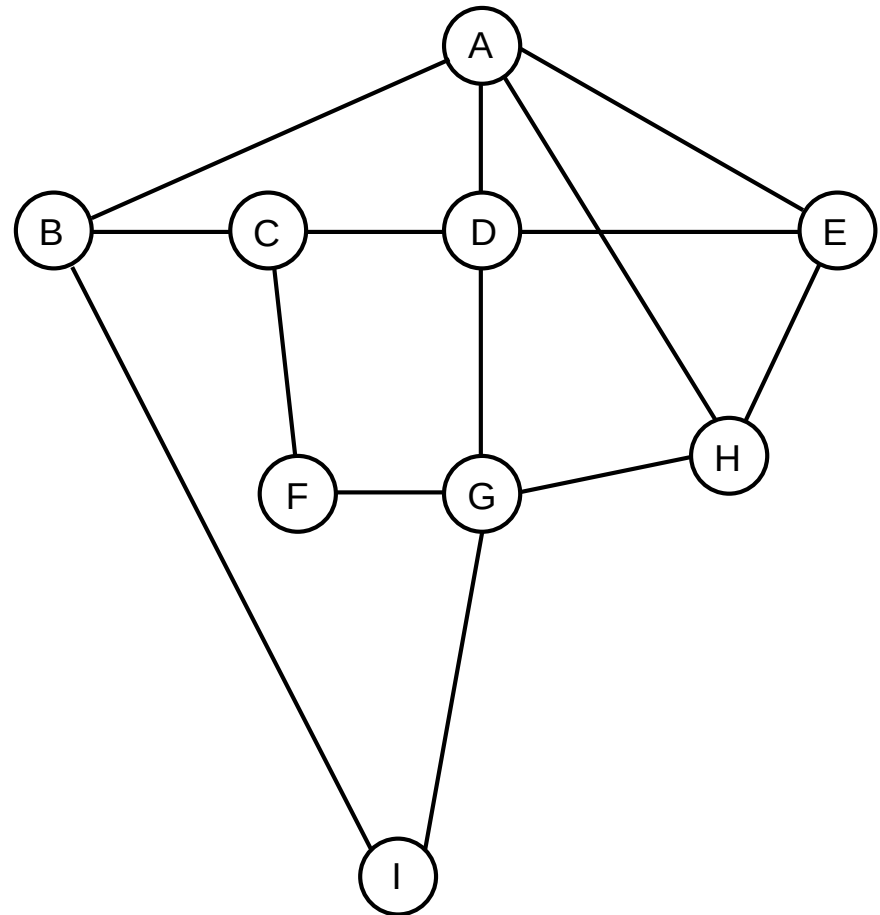


Beispiel:

- Weitere Abstraktion in der Darstellung der Verkehrsbeziehungen

→ **gerade** Verbindungslinien

→ **Störend in Darstellung:**
Kreuzung von
Verbindungen



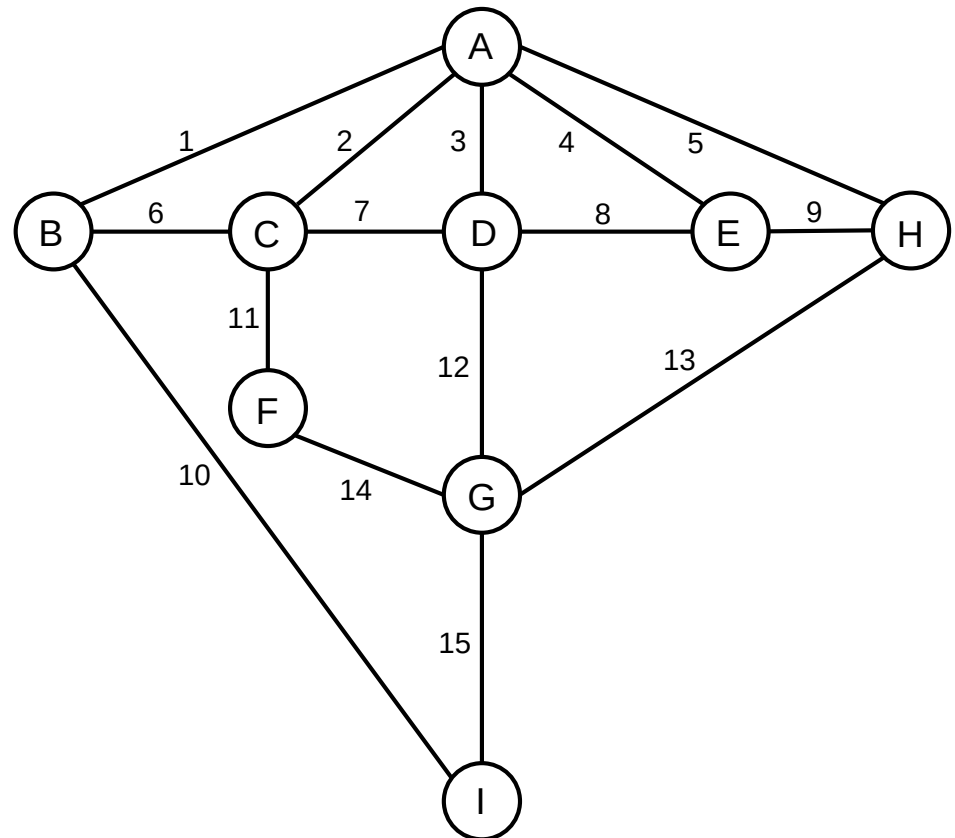
Beispiel:

- Darstellung ohne kreuzende Verbindungen

→ **planare** Darstellung

→ Einfügen von Kantenbezeichnern

→ weiterhin:
automatisierte
Verarbeitung mit
Graphenalgorithmen

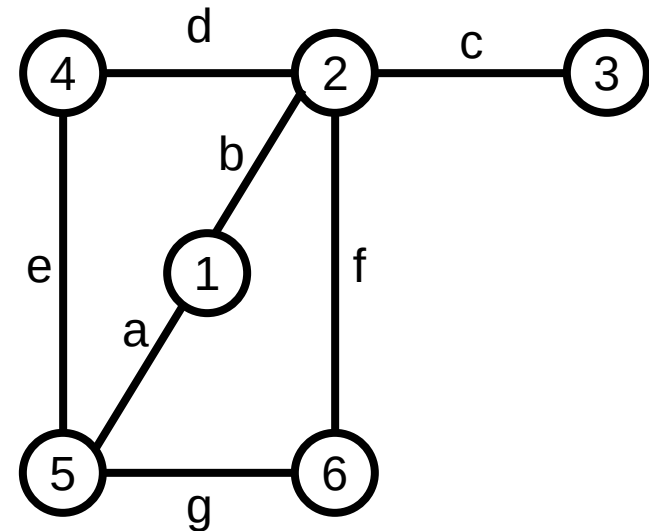


Formale mathematische Beschreibung:

- Weitere Abstraktion von der Bedeutung der Darstellungselemente:
→ Verknüpfung mit der Begriffswelt der Mengen und Relationen
- **Graphen** können ganz **unabhängig** von der **Darstellung** durch **zwei Mengen** und **eine Abbildung** beschrieben werden:
 - **V = Menge der Knoten**
 - **E = Menge der Kanten**
 - **$\Phi(e)$ ordnet jeder Kante $e \in E$ zwei Knoten aus V zu**
→ diejenigen, die durch die Kante e verbunden sind
- **$G(V, E, \Phi)$ wird abstrakter Graph genannt**

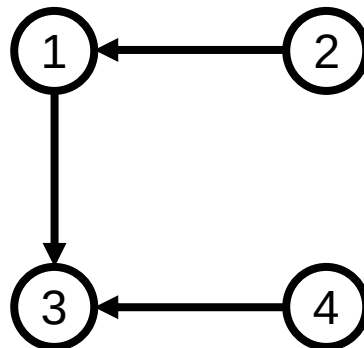
Beispiel:

- $V = \{ 1, 2, \dots, 6 \}$
- $E = \{ a, b, \dots, g \}$
- $\Phi: E \rightarrow \{ v, w \} \quad v, w \in V$
 - $\Phi(a) = \{ 1, 5 \}$
 - $\Phi(b) = \{ 1, 2 \}$
 - $\Phi(c) = \{ 2, 3 \}$
 - $\Phi(d) = \{ 2, 4 \}$
 - $\Phi(e) = \{ 4, 5 \}$
 - $\Phi(f) = \{ 2, 6 \}$
 - $\Phi(g) = \{ 5, 6 \}$



Gerichteter Graph:

- Für manche Probleme benutzt man auch **gerichtete Graphen**
→ **Kanten** haben eine **festgelegte Richtung**
- Bei **gerichteten Graphen** gilt:
 Φ bildet **Kanten** auf **geordnetes Knoten-Tupel** aus **$V \times V$** ab
- Ein gerichteter Graph muss mindestens eine Kante zwischen zwei Knoten (g,h) besitzen, so dass keine Kante in umgekehrter Richtung (h,g) existiert
- Gerichtete Graphen werden auch als **Digraphen** bezeichnet

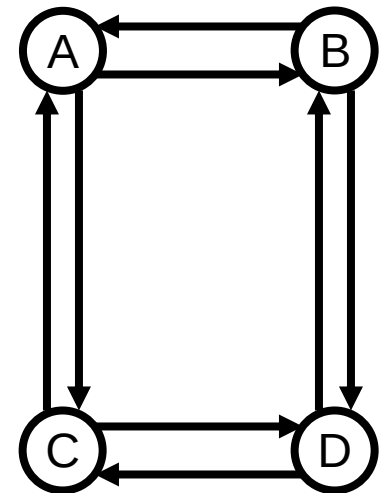
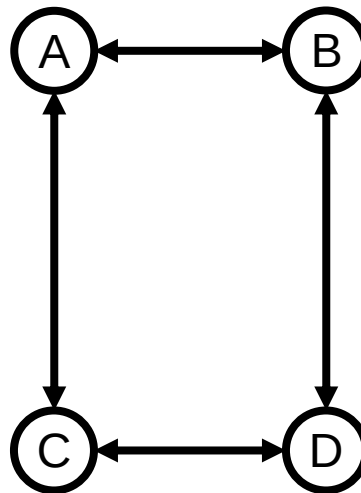
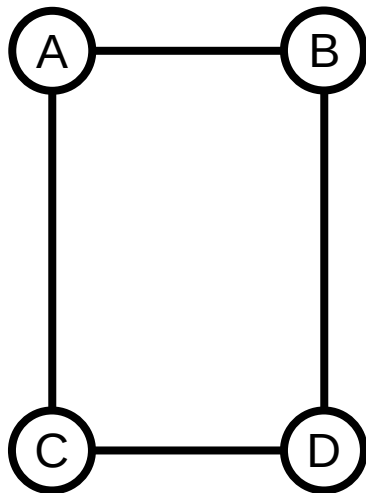


Ungerichteter Graph:

■ Ungerichtete Graphen

- sind **immer** auch mit **gerichteten Kanten** darstellbar
- zu jeder Kante existiert eine weitere Kante in umgekehrter Richtung

Beispiel:



Sprechweisen:

- Verbindet die **Kante e** die **Knoten g** und **h**, so sagt man:
„e ist inzident zu g bzw. zu h“ und schreibt:

- $\Phi(e) = (g, h)$ für **gerichtete** Graphen

- $\Phi(e) = \{g, h\} = \{h, g\}$ für **ungerichtete** Graphen

- Φ wird daher **Inzidenzabbildung** genannt

→ Graph lässt sich formal durch eine **Inzidenzmatrix** beschreiben

- die **Knoten g** und **h** heißen **adjazent zur Kante e**

- Weitere Möglichkeit zur **formalen Beschreibung** eines Graphen ist die **Adjazenzmatrix**

Globale Charakterisierung von Graphen:

- Der **Graphentheorie** fehlt es selbstverständlich nicht an Begriffen, um die unerschöpfliche Vielfalt der Graphen zu kategorisieren
- Im folgenden wollen wir uns nur mit **Graphen** beschäftigen, deren **Mengen V** und **E endlich** sind:
 - **endliche Graphen** (insbesondere für technische Anwendungen)
- Ist die Menge der **Kanten E** leer
 - so handelt es sich um einen **entarteten Graphen**
 - dieser besteht nur aus **isolierten Knoten**
- Wenn zu je zwei verschiedenen **Knoten** höchstens eine **Kante** existiert
 - so handelt es sich um einen **einfachen Graphen**

Globale Charakterisierung von Graphen:

■ Vergleichbarkeit zweier **Graphen**:

→ es werden **Abbildungen** zwischen **Knoten** und **Kanten** gesucht,
so dass die **Inzidenzbeziehungen** erhalten bleiben

■ Sind diese Zuordnungen eindeutig

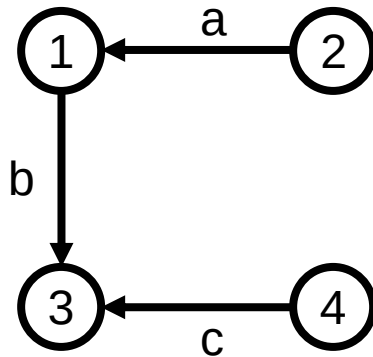
→ so wird der **Graph isomorph** genannt

■ **Isomorphie** von Graphen:

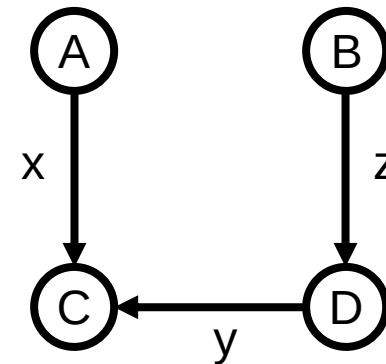
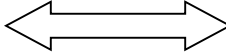
→ Strukturen isomorpher Graphen sind gleich
→ wichtige Eigenschaft in der Vergleichbarkeit
und **formalen Verifikation** digitaltechnischer
Schaltungen und Systeme

Beispiel: Isomorphie

- Die Folgenden Graphen sind **isomorph**



isomorph



- Zuordnung der **Knoten** φ : $(1,2,3,4) \Rightarrow (D,B,C,A)$

- Zuordnung der **Kanten** ψ : $(a,b,c) \Rightarrow (z,y,x)$

- Inzidenzbeziehungen:**

$$\Phi(a) = (2,1) \Leftrightarrow \Phi(\psi(a)) = \Phi(z) = (B,D) = (\varphi(2), \varphi(1))$$

$$\Phi(b) = (1,3) \Leftrightarrow \Phi(\psi(b)) = \Phi(y) = (D,C) = (\varphi(1), \varphi(3))$$

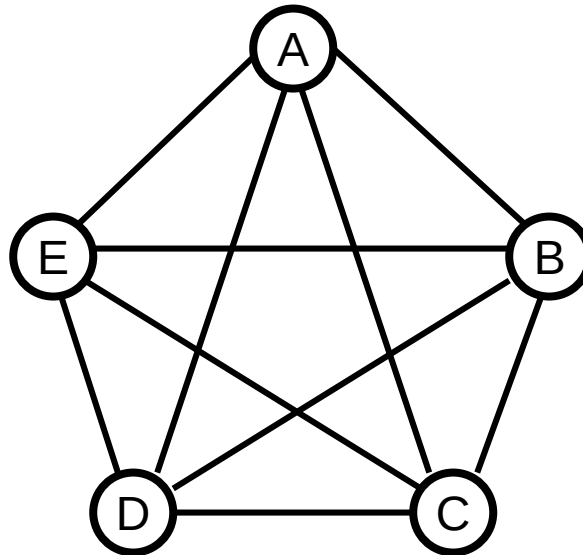
$$\Phi(c) = (4,3) \Leftrightarrow \Phi(\psi(c)) = \Phi(x) = (A,C) = (\varphi(4), \varphi(3))$$

Globale Charakterisierung von Graphen:

Vollständigkeit von (Sub-) Graphen

- sind je zwei verschiedene Knoten durch eine **Kante** verbunden, so ist der (Sub-) Graph **vollständig** → (Sub-) Graph ist **Clique**

Beispiel:

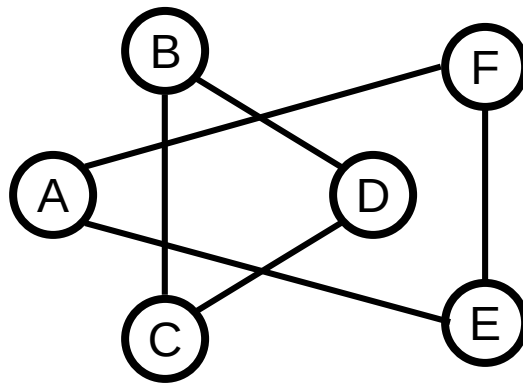


Globale Charakterisierung von Graphen:

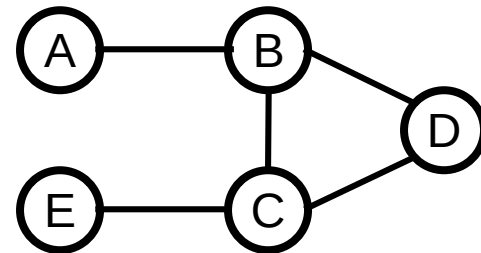
Zusammenhängende Graphen

- durch Folgen von Kanten und Knoten kann man von jedem beliebigen Knoten des Graphen zu jedem anderen Knoten des Graphen gelangen
→ der **Graph** ist dann **zusammenhängend**
- Ist der Graph **nicht zusammenhängend**
→ so besteht er aus mindestens zwei **Teilgraphen**

Beispiele:



nicht zusammenhängend



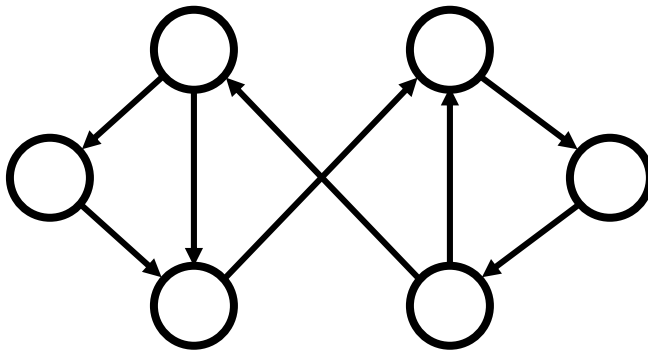
zusammenhängend

Globale Charakterisierung von Graphen:

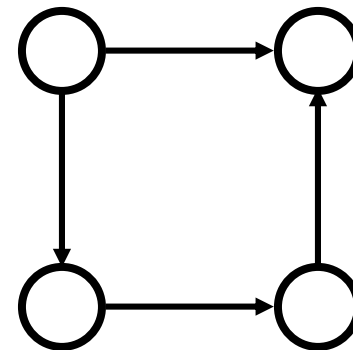
Streng zusammenhängende Graphen

- **Gerichtete Graphen** bezeichnet man als **zusammenhängend**, wenn der zugehörige **ungerichtete Graph zusammenhängend** ist
- Findet man zusätzlich von jedem beliebigen Knoten des Graphen zu jedem anderen Knoten eine **gerichtete Folge von Kanten** (Weg unter Einbezug der Richtungen!) → so ist der **Graph streng zusammenhängend**

Beispiele:



streng zusammenhängend



nicht streng zusammenhängend

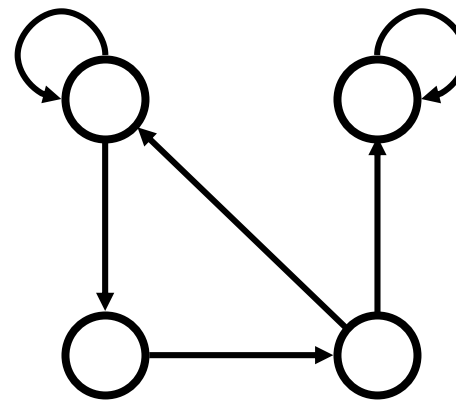
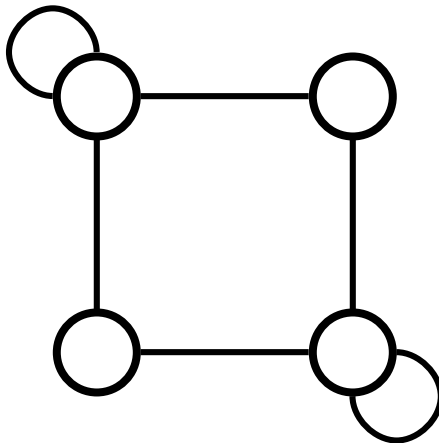
Lokale Eigenschaften von Graphen:

Schlinge

■ Verbindet eine Kante einen Knoten mit sich selbst, so wird diese als **Schleife** oder **Schlinge** bezeichnet

■ $\Phi(e) = (g, g)$

Beispiele:



Lokale Eigenschaften von Graphen:

Mehrfachkanten

- Existieren **mehrere Kanten** zwischen zwei Knoten g und h , so heißen diese **parallel** bzw. **Mehrfachkanten**

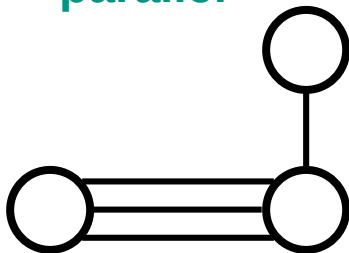
$$\Phi(e) = \Phi(f) = \{g, h\} \text{ bzw. } (g, h) \quad e \neq f$$

- Bei **gerichteten Graphen** nennt man Kanten **antiparallel**, falls sie zwei Knoten in entgegengesetzter Richtung verbinden

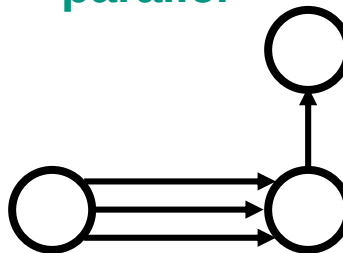
$$\Phi(e) = (g, h) \quad \Phi(f) = (h, g)$$

Beispiele:

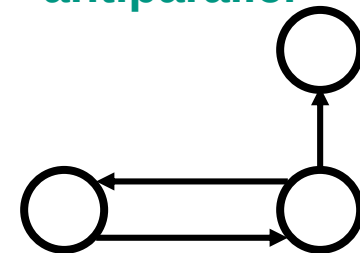
parallel



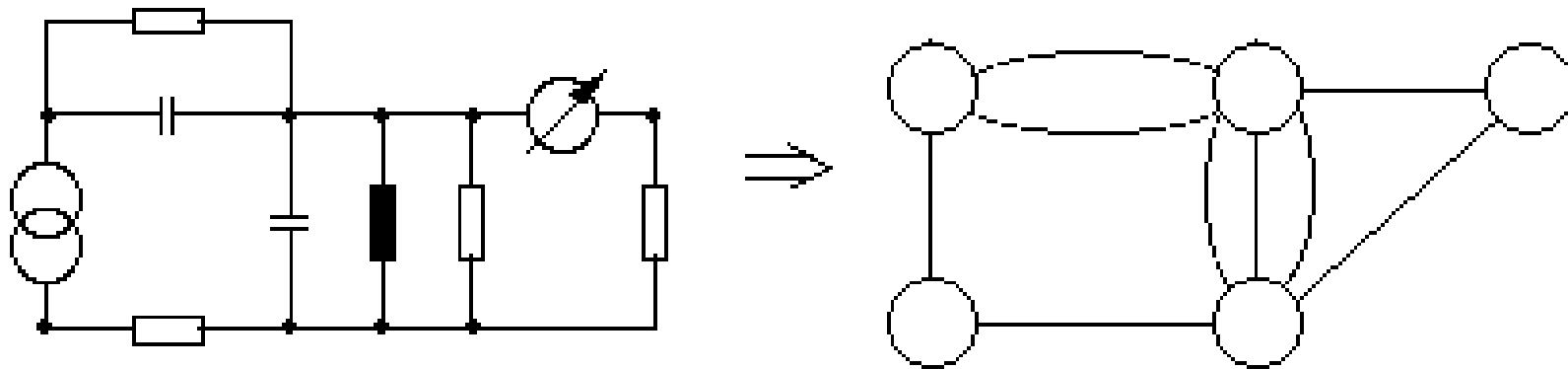
parallel



antiparallel



Technisches Beispiel: Mehrfachkanten



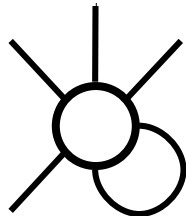
- Analoger Schaltkreis:
 - **Mehrfachkante** als Abbild schaltungstechnischer Merkmale
 - Bauteile **parallel** geschaltet

Lokale Eigenschaften von Graphen:

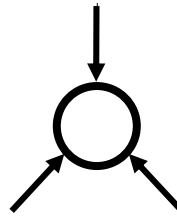
Grad eines Knotens

- Betrachtet man einen **Knoten**, so wird die **Anzahl** der damit **inzidenten Kanten** als **Grad $d(g)$** des Knotens bezeichnet
- Bei **gerichteten Graphen** unterscheidet man zusätzlich
 - abgehende Kanten $\Rightarrow$ **Ausgangsgrad $d^+(g)$**
 - von ankommenden Kanten $\Rightarrow$ **Eingangsgrad $d^-(g)$**

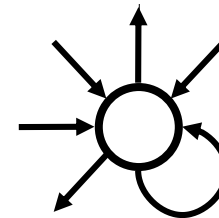
Beispiele:



$$d(g) = 6$$



$$d^-(g) = 3$$



$$\begin{aligned} d^+(g) &= 3 \\ d^-(g) &= 4 \end{aligned}$$

Lokale Eigenschaften von Graphen:

Unmittelbare Nachbarschaft von Knoten

- Sind **zwei Knoten** durch **eine Kante** verbunden
→ so sind die Knoten **unmittelbar benachbart**
- Menge der **unmittelbar benachbarten Knoten**
→ wird mit **$V'(g)$** bezeichnet
- Bei **gerichteten Graphen** gilt:
 - die Menge der benachbarten Knoten wird **entsprechend** der Richtungen der Kanten eingeteilt in:
 - **unmittelbare Vorgänger** $V'_1(g)$ und
 - **unmittelbare Nachfolger** $V'_2(g)$

Lokale Eigenschaften von Graphen:

Unmittelbare Nachbarschaft:

→ mengenalgebraische Eigenschaften

■ Für die **Mengen** der **unmittelbaren Nachbarn** gilt:

$$■ |V'_1(g)| = d^-(g)$$

$$■ |V'_2(g)| = d^+(g)$$

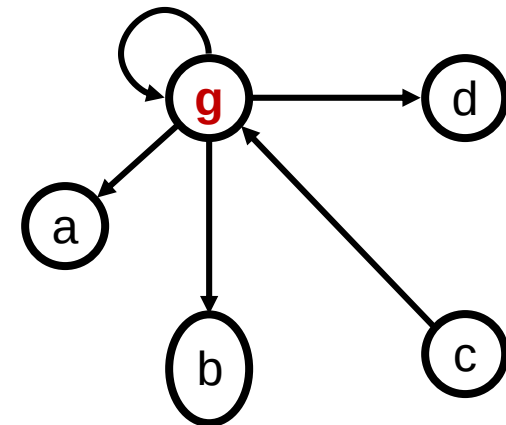
$$■ V' = V'_1 \cup V'_2$$

■ Da jedoch Knoten sowohl Vorgänger als auch Nachfolger sein können, gilt:

$$■ |V'(g)| \leq d(g)$$

Beispiel: **Unmittelbare Nachbarschaft** des **Knotens g**

- **Vorgänger:** $V'_1(g) = \{ c, g \}$
- **Nachfolger:** $V'_2(g) = \{ a, b, d, g \}$
- $V'(g) = \{ a, b, c, d, g \}$
- $d^-(g) = |V'_1(g)| = 2$
- $d^+(g) = |V'_2(g)| = 4$
- $d(g) = d^-(g) + d^+(g) = 6$
- $|V'(g)| = 5$



Lokale Eigenschaften von Graphen:

Mittelbare Nachbarschaft von Knoten

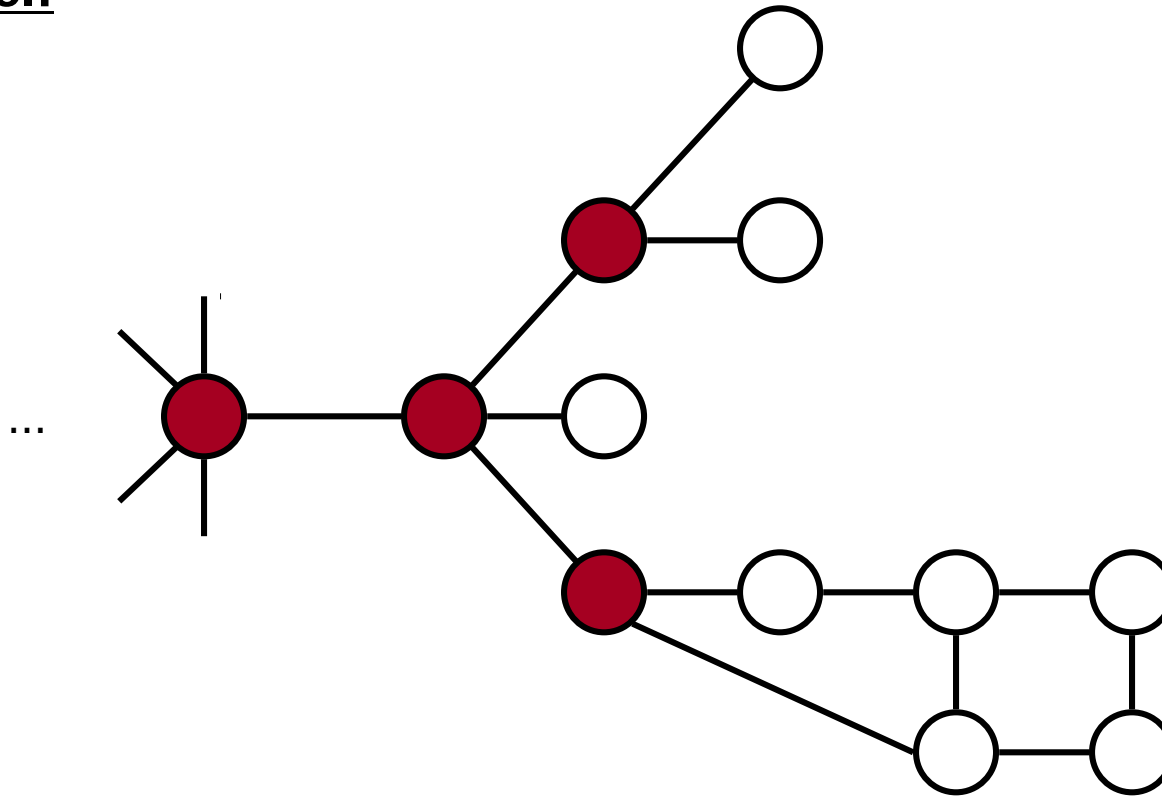
- Schwächt man die Forderung so ab, dass nur eine **Folge** von **Kanten** **zwischen zwei Knoten** existieren muss
→ so sind die Knoten **mittelbar benachbart**
- Nützlich ist die mittelbare Nachbarschaft besonders bei gerichteten Graphen bzgl. der Unterscheidung in
 - **mittelbare Vorgänger** und
 - **mittelbare Nachfolger**
- Beispiel: Stammbaum
 - die Frage nach der Abstammung in direkter Linie ist eine Frage nach mittelbaren Vorgängern

Lokale Eigenschaften von Graphen:

Artikulation eines Graphen

- Bei der **Modellierung** von technischen Systemen mit Graphen werden
 - **Komponenten** häufig durch **Knoten** dargestellt
 - **Verbindungen** zwischen Komponenten durch **Kanten** abgebildet
- Der **Ausfall** einer **Komponente**
 - entspricht dem **Entfernen** eines **Knotens** aus dem Graphen
- Besonders **kritisch** bei Ausfall einer Komponente (Knoten):
 - Graph **zerfällt** in **zwei nicht zusammenhängende Teilgraphen**
 - solche **kritischen Knoten** erhalten den Namen **Artikulation**
- Die **Artikulationen** eines Graphen repräsentieren **kritische Systemstellen**

Beispiel:



Graph mit **Artikulationen**

Spezielle Kantenfolgen in Graphen:

- Bereits bei der mittelbaren Nachbarschaft haben wir uns dafür interessiert, ob Knoten über **Kantenfolgen** verbunden sind

- **Definition:** **Kantenprogression** der **Länge n**

→ **endliche Folge** von **n nicht** notwendigerweise **verschiedenen Kanten (e^i)**, die $n + 1$ nicht notwendigerweise verschiedene Knoten (g^i) verbinden

$$\Phi(e^i) = (g^i, g^{i+1}) \quad \text{für } i = 1, 2, \dots, n$$

- Sind in einer **Kantenprogression alle Knoten (g^i)** voneinander **verschieden** und damit auch alle Kanten, so heißt sie **einfach**
- Je nachdem, ob der Anfangsknoten gleich dem Endknoten ist ($g^1 = g^{n+1}$), oder ob es sich um einen gerichteten Graphen handelt, verwendet man unterschiedliche Bezeichnungen

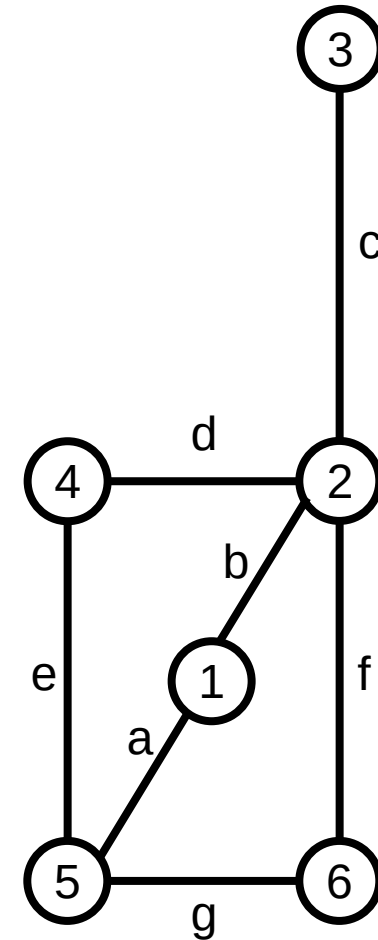
Spezielle Kantenfolgen in Graphen

Ungerichteter Graph		Gerichteter Graph	
$g^1 \neq g^{n+1}$	$g^1 = g^{n+1}$	$g^1 \neq g^{n+1}$	$g^1 = g^{n+1}$
offene Kanten-progression	geschlossene Kanten-progression	offene Kanten-progression	geschlossene Kanten-progression
Sind alle Kanten einer Progression voneinander verschieden			
Ketten-progression	geschlossene Kantenzug-progression	Weg-progression	Zyklus-progression
Berücksichtigt man die Kanten einer Progression ohne Ordnung			
Kette	geschlossener Kantenzug	Weg	Zyklus

Spezielle Kantenfolgen in Graphen

Beispiele am ungerichteten Graphen:

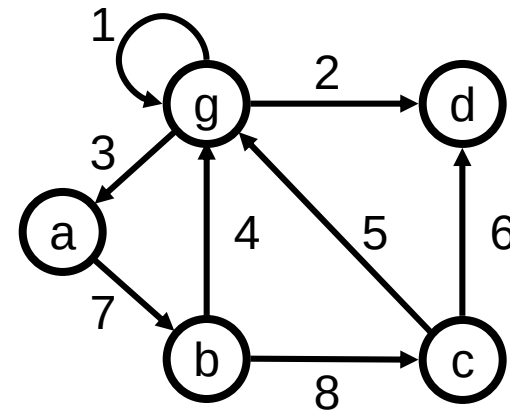
<u>Folge</u>	<u>Merkmal</u>
g-f-c-c-d	offene Kantenprogression
d-c-c-b-a-e	geschlossene Kantenprogression
e-g-f	Kettenprogression
a-b-f-g	geschlossene Kantenzugprogression
{ e, b, a }	Kette
{ g, d, f, e }	geschlossener Kantenzug



Spezielle Kantenfolgen in Graphen

Beispiele am gerichteten Graphen:

<u>Folge</u>	<u>Merkmal</u>
4-3-7-4	offene Kantenprogression
3-7-4-1-1	geschlossene Kantenprogression
4-1-2	Wegprogression
5-1-3-7-8	Zyklusprogression
{ 2, 8, 5, 1 }	Weg
{ 1 }	Zyklus



Spezielle Graphen:

Begriff des **Zyklus/Zyklen** in Graphen:

- Für viele Algorithmen, die auf Graphen arbeiten ist es wichtig zu wissen, ob es möglich ist, mit unterschiedlichen Kanten „im Kreis zu laufen“
- Man bezeichnet einen kompletten **Graphen** als **zyklisch**, wenn
 - wenigstens eine **geschlossene Kantenzugprogression** (= **Zyklus**) existiert in diesem Graphen
 - eine **Schleife** ist ebenfalls ein **Zyklus**
- Wenn ein **Graph nicht zyklisch** ist
 - nennt man ihn **zyklenfrei** oder **azyklisch**

Spezielle Graphen:

Baum

■ Definition: Baum

→ ein **zusammenhängender, zyklensfreier** Graph

■ Also: bei einem **ungerichteten Baum**

→ es darf **kein geschlossener Kantenzug** existieren

■ Weiterhin: bei einem **gerichteten Baum** gilt:

→ es darf **kein Zyklus** existieren

→ der zugehörige ungerichtete Graph
muss **zusammenhängend** sein

Spezielle Graphen: Bäume

Definition: Wurzel und Blätter

■ In einem gerichteten Baum

→ gibt es genau einen Knoten, der keine Vorgänger hat

$$d^- = 0$$

→ dieser Knoten wird Wurzel genannt

■ Die Knoten ohne Nachfolger ($d^+ = 0$) heißen Blätter

■ Bei einem ungerichteten Baum

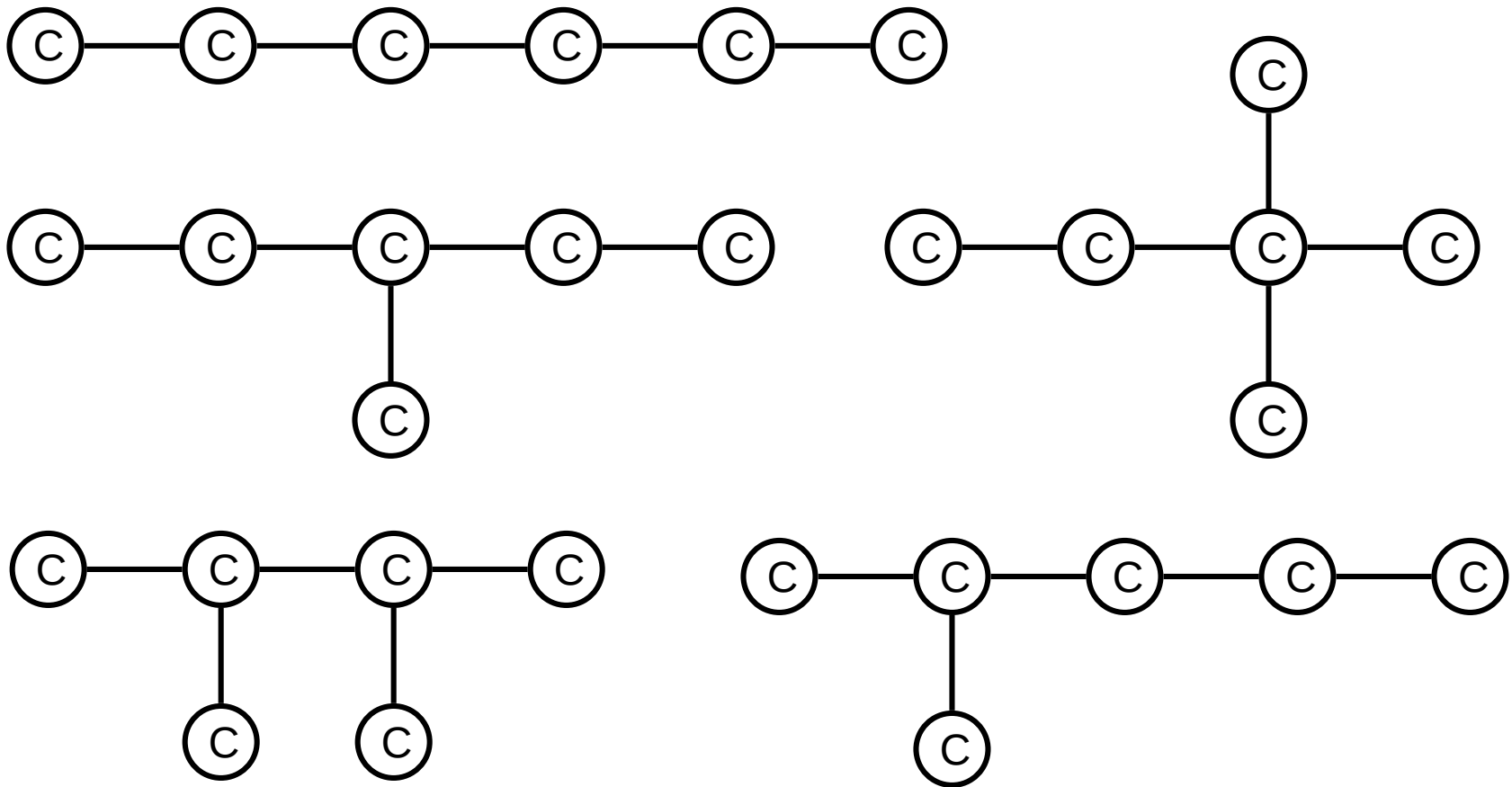
→ kann die Wurzel frei gewählt werden

→ daraus ergeben sich die Blätter

als übrige Knoten mit Knotengrad eins

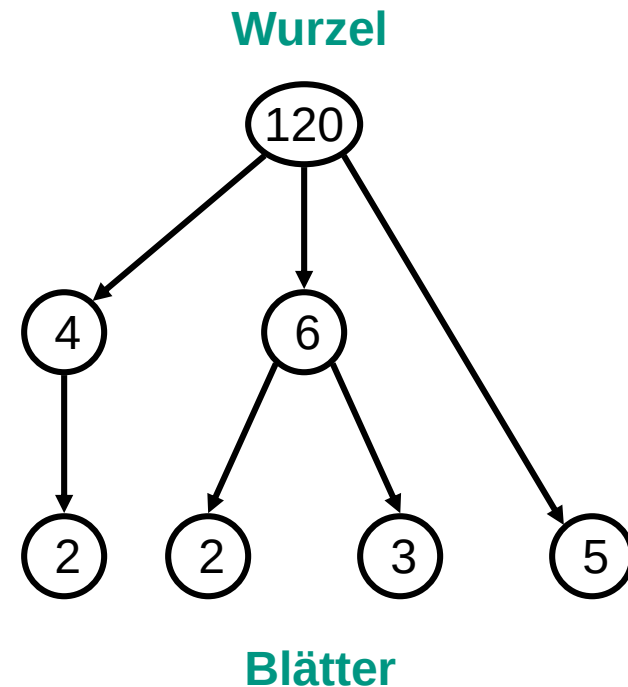
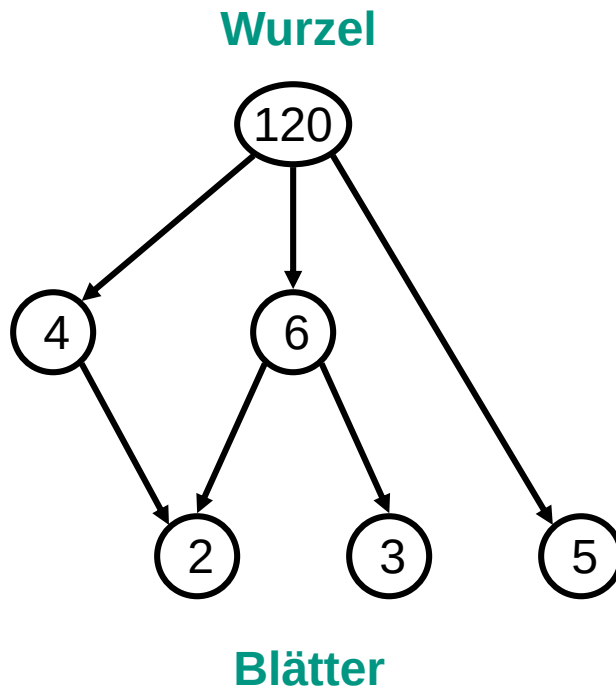
Beispiel: ungerichteter Baum

■ Die **Molekülstrukturen** der **Hexane** ergeben **ungerichtete Bäume**



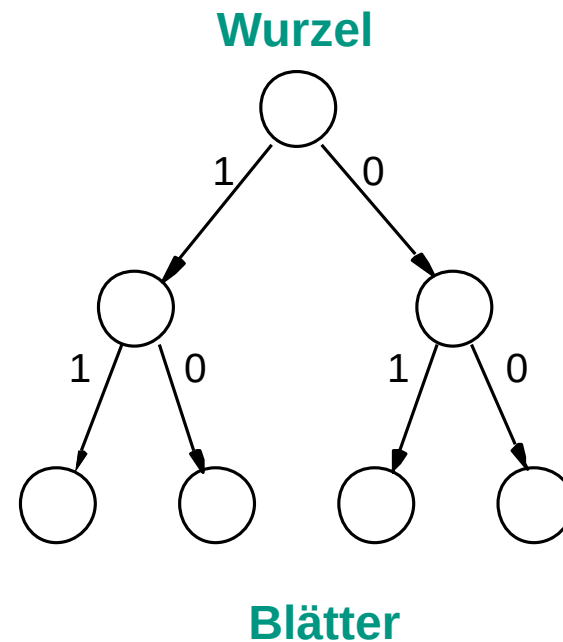
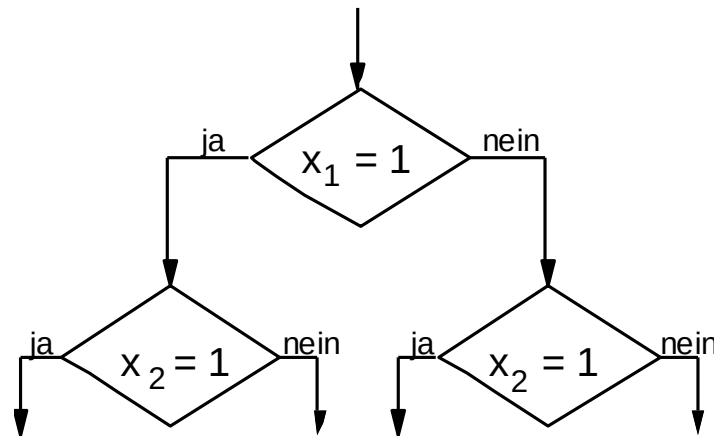
Beispiel: gerichteter Baum

- Die Relation „ist teilbar durch“ führt zu einem gerichteten Baum



Definition: Binärbaum

- Hat jeder Knoten eines Baumes, außer den Blättern, **genau zwei** Nachfolger: $d^+(g) = 2$,
so heißt ein solcher Baum **Binärbaum**



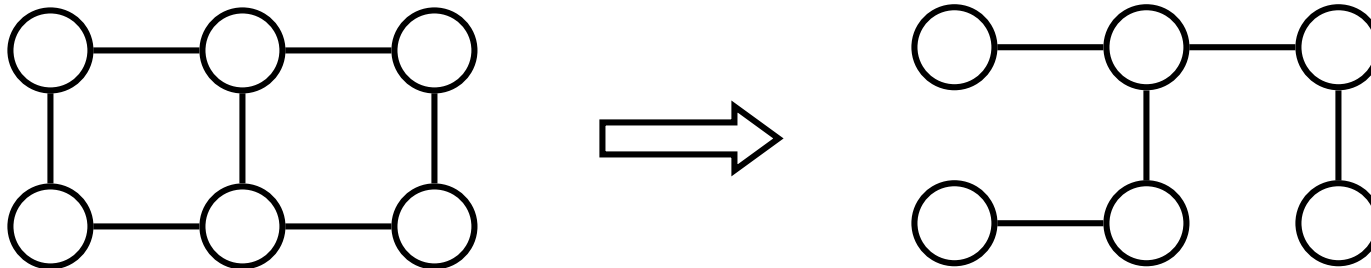
Weitere Begriffe: Binärbaum

- Der **Abstand** von der **Wurzel** zu den **Blättern**
→ wird als **Tiefe** des **Baumes** bezeichnet
- Ist der Abstand von der **Wurzel** zu den **Blättern**
für alle Blätter **identisch**
→ so handelt es sich um einen
symmetrischen Binärbaum
- Unterscheiden sich die Abstände um maximal eins
→ so nennt man den **Baum ausgeglichen**

Definition: **Aufspannender Baum**

- Es gilt: bei jedem **zusammenhängenden Graphen** kann man **alle Zyklen** durch **gezieltes Entfernen** von **Kanten** auflösen
(→ *Kruskal Algorithmus*: Minimal Aufspannender Baum, MAB)
ohne dass der Graph in **zwei Teilgraphen zerfällt**
- Auf diese Weise erhält man zu jedem beliebigen Graphen einen zugehörigen **aufspannenden Baum**

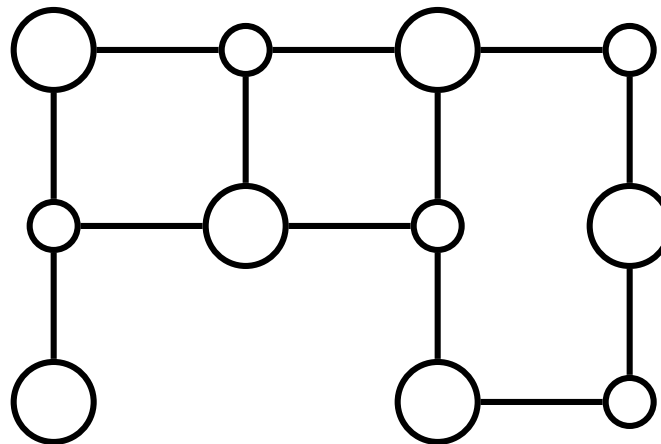
Beispiel:



Definition: **Bipartiter Graph**

- **Knoten** eines **Graphen** sind folgendermaßen in **zwei Teilmengen** aufteilbar:
 - **keine Kante** verbindet **zwei Knoten** in **derselben Teilmenge**
 - so nennt man den **Graphen bipartit**
- In technischen Systemen:
 - **zwei Teilmengen** entsprechen bspw. **zwei** verschiedenen **Typen** von **Komponenten**, die nicht untereinander verbunden werden dürfen

Beispiel:



Definition: Geometrische Graphen

- Bisher hatte die Darstellung der Knoten und Kanten des Graphen in der Zeichenebene keine Bedeutung
- **Geometrische Graphen** ordnen den **Knoten** jedoch eine geometrische **Position** zu:
 - im **n-dimensionalen Raum $\mathbb{R}^n$**
 - zu den **Kanten** „glatte“ (**kreuzungsfreie!**) **Kurven zwischen den Knoten**, die außer den Endknoten selbst keinen Punkt gemeinsam haben
- **Zweidimensionale** geometrische **Graphen**
 - müssen im zweidimensionalen Raum (2D) **kreuzungsfrei** sein
- Man kann beweisen, dass zu **jedem Graphen** ein **isomorpher dreidimensionaler geometrischer Graph** existiert

Eigenschaften: Geometrische Graphen

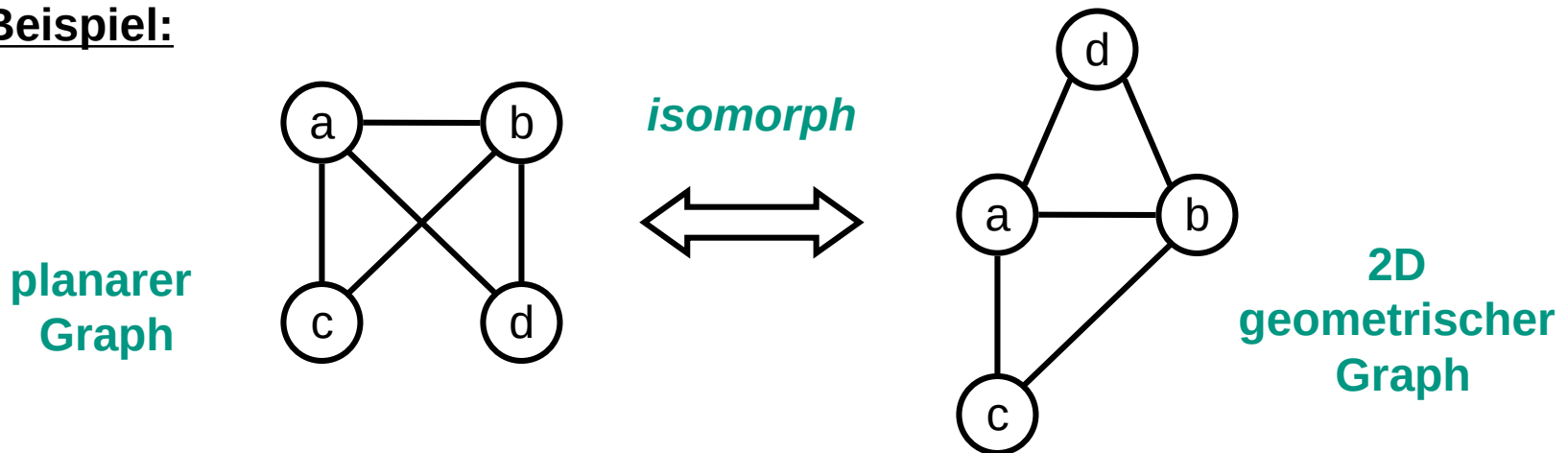
■ In zweidimensionaler Ebene:

→ es existiert **nicht immer** im 2D Raum $\mathbb{R}^2$ ein **isomorpher kreuzungsfreier** Graph mit „**glatten**“ Kanten

■ Graphen, zu denen ein **isomorpher geometrischer** Graph in 2D existiert
→ nennt man **planare Graphen**

■ **Planare Graphen** müssen **nicht immer kreuzungsfrei** dargestellt sein

Beispiel:



Definition: Duale Graphen

■ Planarer Graph:

- teilt die **Zeichenebene** in **verschiedene Gebiete** ein
- **äußere Umgebung** des **Graphen** wird hier auch als **Gebiet** betrachtet

■ Planarität des Graphen:

- **notwendige** und **hinreichende** Bedingung dafür, dass man einen **dualen Graphen** dazu konstruieren kann

■ Vorgehensweise: Konstruktion eines dualen Graphen

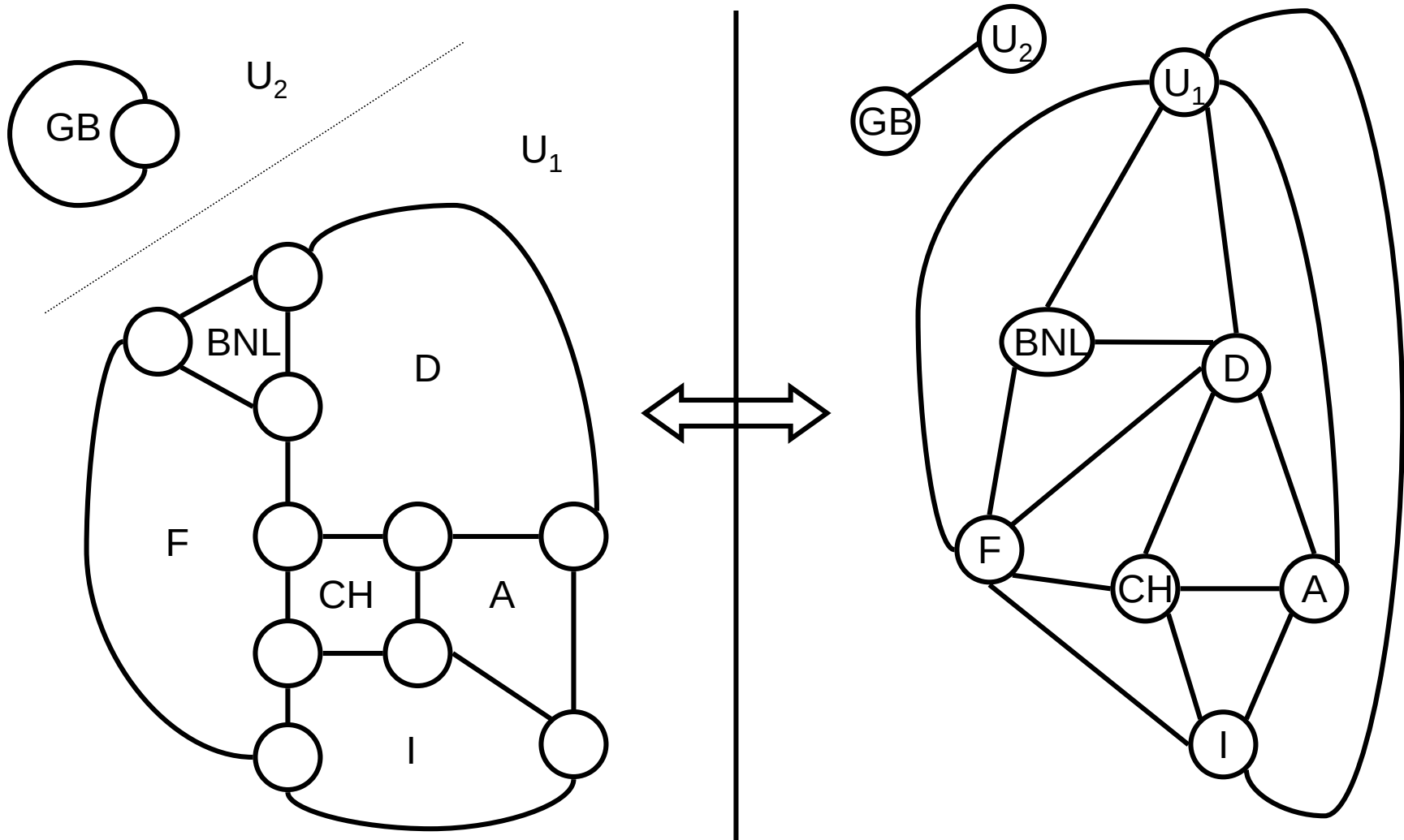
- **Gebiete** eines zusammenhängenden **Graphen** in der Zeichenebene entsprechen den **Knoten** im **dualen Graphen**, **jede Grenze zwischen zwei Gebieten** (Kanten im Ausgangsgraph!) entspricht dabei **einer kreuzungsfreien Kante** im zu konstruierenden **dualen Graph**; jedem eigenständigen zusammenhängenden Graphen wird dabei ein eigener **Umgebungsknoten U** zugeordnet

■ Duale Graphen:

- spielen beispielsweise eine Rolle, bei der **Beurteilung**, ob eine **Schaltung planar** in einer 2D-Ebene **realisiert** werden kann

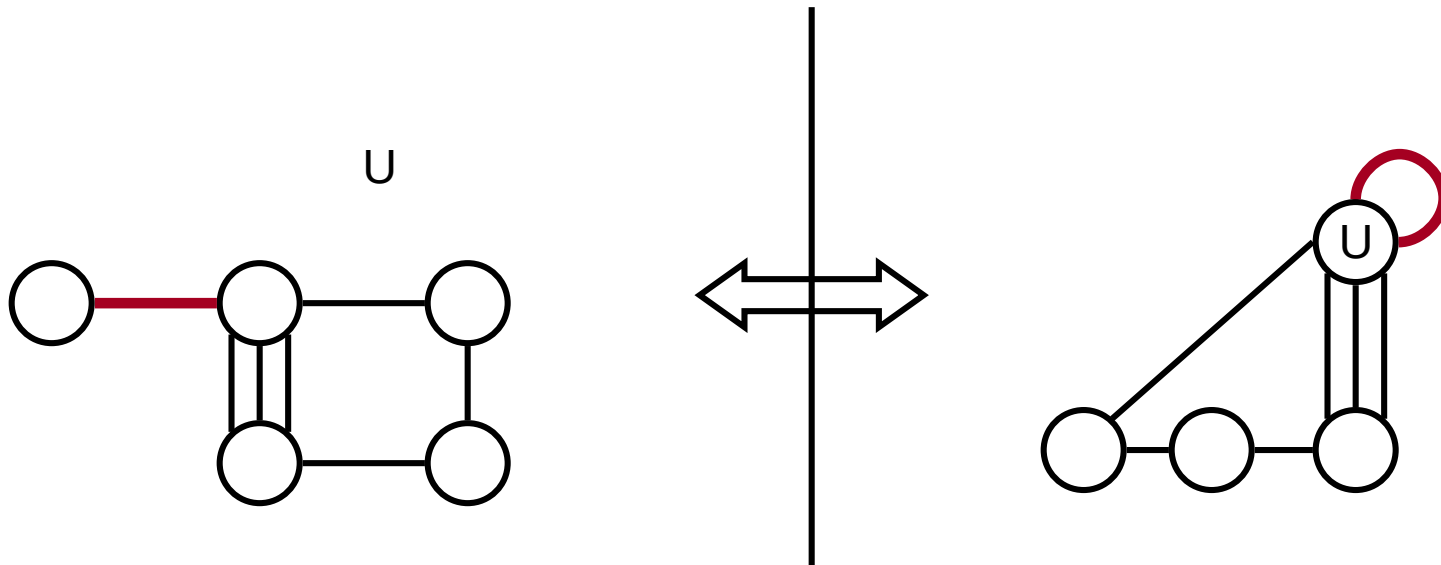
Duale Graphen

Beispiel 1: Duale Graphen



Beispiel 2: **Duale Graphen**

→ Konstruktion von Schlingen



Definition: Gewichtete Graphen

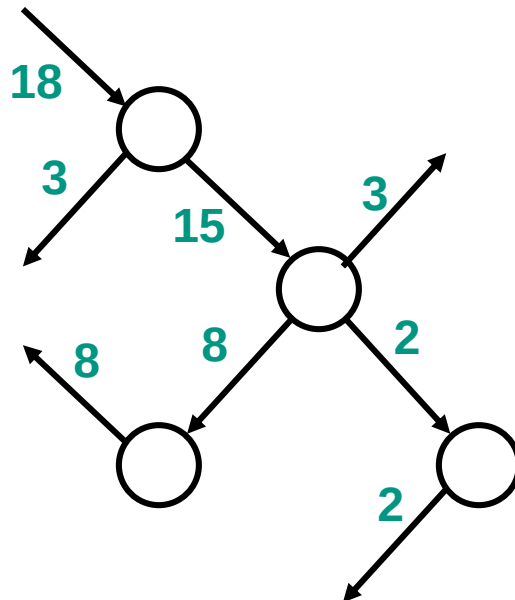
- Analog zum Scotland Yard Spiel:
 - es macht einen **Unterschied**, ob **zwei Knoten** mit einer Bus-, Taxi- oder U-Bahn-Verbindung verbunden sind
- Man kann jeder **Kante** noch **zusätzliche Attribute** zuordnen
- In vielen Fällen genügt jedoch die **Zuordnung** einer reellen Zahl, die als **Gewicht** bezeichnet wird
 - dann spricht man von einem **gewichteten Graphen**
- **Gewichtete Graphen:**
 - dienen zur **Modellierung** von:
 - Netzwerkflüssen,
 - Transportprozessen, und
 - Planungsstrategien

Gewichtete Graphen

Beispiele: Gewichtete Graphen

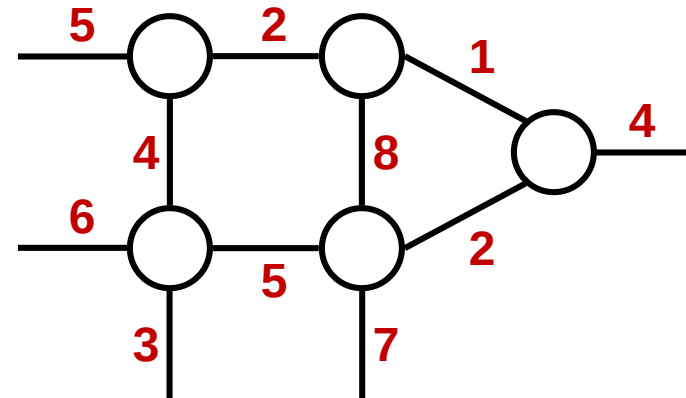
Netzwerkfluss

Gewicht: **Durchsatz**



Transportprozess

Gewicht: **Entfernung**



Beispiel: Planungsstrategie

- **Produktionsabläufe** sind mit **gewichteten Graphen** modellierbar:
 - die **gerichteten Kanten** werden als Arbeitsschritt aufgefasst
 - als **Gewicht** wird die jeweils benötigte Dauer verwendet
- mehrere ankommende **Kanten** an **einem Knoten** markieren
 - **Arbeitsschritte**, die **vorher abgearbeitet sein müssen**, bevor die Arbeitsschritte der abgehenden Kanten begonnen werden können
- Also: **Knoten** sind **Produktionszustände**
 - für deren Erreichen ist **minimale Arbeitsdauer** zu berechnen
- Gesucht: die **Dauer** des **kompletten Produktionsablaufs**

The graph consists of 12 nodes and 18 directed edges. The edges are labeled with numbers. The final node on the right is labeled with the sum of the numbers on the edges leading to it.

Edges and their labels:

- Node 1 (top-left) to Node 2 (top-middle): 2
- Node 1 to Node 3 (top-right): 8
- Node 2 to Node 4 (middle-left): 1
- Node 2 to Node 5 (middle-middle): 5
- Node 3 to Node 6 (middle-right): 7
- Node 3 to Node 7 (bottom-left): 3
- Node 4 to Node 8 (bottom-middle): 3
- Node 5 to Node 9 (bottom-right): 2
- Node 6 to Node 10 (bottom-left): 4
- Node 6 to Node 11 (bottom-middle): 4
- Node 7 to Node 12 (bottom-right): 1
- Node 8 to Node 13 (bottom-left): 5
- Node 9 to Node 14 (bottom-middle): 6
- Node 10 to Node 15 (bottom-right): 2
- Node 11 to Node 16 (bottom-left): 8
- Node 12 to Node 17 (bottom-right): 7
- Node 13 to Node 18 (bottom-left): 4
- Node 14 to Node 19 (bottom-middle): 3
- Node 15 to Node 20 (bottom-right): 4
- Node 16 to Node 21 (bottom-left): 9
- Node 17 to Node 22 (bottom-right): 2

The final node on the right is labeled with the sum of the numbers on the edges leading to it: $\Sigma = ???$

Beispiel: Planungsstrategie

Vorgehensweise:

→ Berechnung der Dauer des kompletten Produktionsablaufs

- zur Lösung des Problems berechnet man jeweils das **Maximum** der **Gewichtssummen** sämtlicher Wege zu einem Knoten
- die **maximale Summe** steht für die **Gesamtzeit** des Auftrags
- **Optimalerweise:**
 - man wählt die **Knotenreihenfolge** bzw. **Kantenfolge** so, dass die **Gewichtssummen** der **Vorgängerknoten** schon berechnet sind
- Die Wege mit **maximalen Gewichtssummen** zum Endknoten ergeben einen (maximal) **aufspannenden Graphen**

Beispiel: Planungsstrategie

