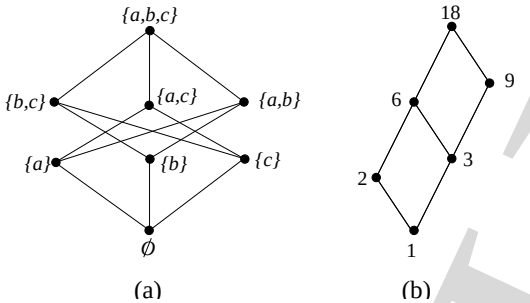


Bild 1.5: Zwei partielle Ordnungen.

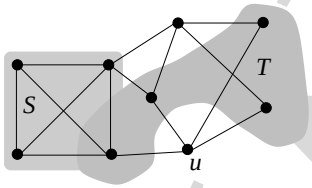


Bild 1.6: Eine Clique $S$ der Größe 4 und eine unabhängige Menge $T$ der Größe 4; der Knoten $u$ hat vier Nachbarn, also hat Grad 4.

## 1.2 Graphen

Binäre Relationen auf *endlichen* Mengen nennt man auch »Graphen«. Solche Relationen sind anschaulich und vielseitig anwendbar. Es gibt Punkte (*Knoten*) und Linien[2] (*Kanten*) zwischen einigen dieser Punkte (Bild 1.6). Graphen sind so vielseitig, weil die Idee so einfach ist: Es gibt eine Menge $V$ von Objekten (die Knoten), und zwei Knoten sind entweder verbunden oder nicht. Graphen sind die wichtigsten Objekte der Informatik überhaupt.

Ein *gerichteter* Graph $G = (V, E)$ ist gegeben durch die Menge $V$ seiner Knoten und die Menge $E \subseteq V \times V$ seiner Kanten. Man sagt, dass die Kante $e = (u, v)$ die Knoten $u$ und $v$ verbindet; die Knoten $u, v$ selbst sind *Endknoten* von $e$. Zwei Knoten, die in einem Graphen durch eine Kante verbunden sind, heißen *adjazent* oder *benachbart*. Die Anzahl aller Nachbarn von $u$ nennt man als *Grad* von $u$ (engl. Grad = degree). In dem Fall von *ungerichteten* Graphen nimmt man zusätzlich an, dass die Relation $E$ antireflexiv ($(v, v) \notin E$ für alle $v \in V$) und symmetrisch (wenn $(u, v) \in E$, dann auch $(v, u) \in E$) ist. In diesem Fall sind also Kanten 2-elementige Knotenmengen $\{u, v\}$ mit $u \neq v$. In einem ungerichteten Graphen sind die Kanten mit Linien und in gerichteten Graphen sind sie mit gerichteten Pfeilen verbunden. Einen *Teilgraphen* erhält man, indem man einige Kanten und/oder einige Knoten (mit zu ihnen inzidenten Kanten) entfernt.

In der Graphentheorie untersucht man verschiedene Charakteristiken von Graphen. Die meisten Fakten dieser Theorie werden in weiteren Informatik-Vorlesungen vorgestellt, spätestens dann, wenn man zu Graphenalgorithmen kommt. Daher beschränken wir uns hier nur auf einige Begriffe, die wir in diesem Buch benutzen werden.

2 Die Linien müssen nicht unbedingt gerade sein und können sich schneiden.

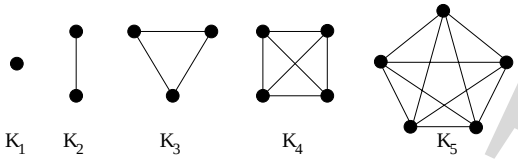


Bild 1.7: Cliquen $K_s$ mit $s = 1,2,3,4,5$ Knoten.

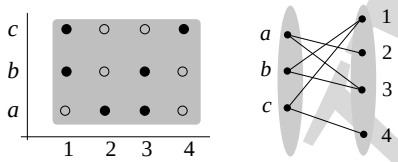


Bild 1.8: Eine binäre Relation und ihre Darstellung als ein bipartiter Graph.

Eine *Clique* in einem Graphen $G = (V, E)$ ist eine Teilmenge $S \subseteq V$ von Knoten, so dass zwischen je zwei Knoten in $S$ eine Kante liegt. In der Soziologie werden Beziehungsgraphen behandelt, das heißt Knoten symbolisieren Personen und Kanten enge Beziehungen. Damit wird die Bezeichnung »Clique« anschaulich klar. Cliquen mit $s$ Knoten bezeichnet man auch durch $K_s$ (Bild 1.7).

Das Gegenteil von Clique ist *unabhängige Menge* – das ist eine Teilmenge $T \subseteq V$ der Knoten, so dass zwischen keinen zwei Knoten in $T$ eine Kante liegt (siehe Bild 1.6).

Eine interessante Klasse von Graphen sind die *bipartiten Graphen*. Diese Graphen $G = (V, E)$ haben die Eigenschaft, dass es eine Zerlegung der Knotenmenge $V$ in zwei disjunkte Teilmengen $A$ und $B$ gibt, so dass für jede Kante der eine Endknoten zu $A$ gehört und der andere zu $B$. Bipartite Graphen haben eine große Bedeutung, liefern sie doch unmittelbar eine Veranschaulichung der binären Relationen. Tatsächlich können nämlich die Elemente einer *beliebigen* Relation $R \subseteq A \times B$ als Kanten von Knoten aus $A$ nach Knoten aus $B$ aufgefasst werden (Bild 1.8).

Graphen werden gewöhnlich mit Hilfe geometrischer Diagramme dargestellt. Oft aber wird zwischen einem Graphen und einem diesen Graphen darstellenden Diagramm nicht deutlich unterschieden. Es muss aber ausdrücklich davor gewarnt werden, Graphen und Diagramme gleichzusetzen. Spezielle geometrische Darstellungen können das Vorhandensein von Eigenschaften suggerieren, die der dargestellte Graph als eine Struktur, die lediglich aus einer Knotenmenge und einer Relation über dieser Menge besteht, gar nicht besitzen kann. Zum Beispiel kann ein Kreis der Länge 5 als 5-zackiger Stern dargestellt werden (siehe Bild 1.9). Um verschiedene aber demselben Graphen entsprechende Diagramme als ein Objekt zu betrachten, benutzt man den Begriff der »Isomorphie«: Zwei Graphen sind *isomorph*, falls sie sich nur bis auf die Nummerierung ihrer Knoten unterscheiden.

Ein *Weg* (engl. walk) von Knoten $u$ nach Knoten $v$ in einem Graphen ist eine Folge $u_0, \ldots, u_r$ von (nicht notwendig verschiedenen) jeweils benachbarten Knoten mit $u = u_0$ und $v = u_r$. Die *Länge* dieses Weges ist die Anzahl $r-1$ der Kanten, die er erhält. Beachte, dass i. A. ein Weg sowohl einen Knoten wie auch eine Kante *mehrmals* durchlaufen darf!

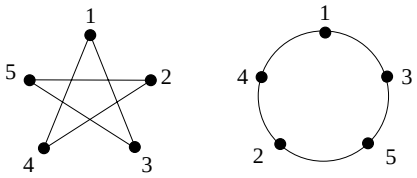


Bild 1.9: Diese zwei Graphen sind isomorph, sie stellen also dieselbe Relation zwischen Knoten dar.

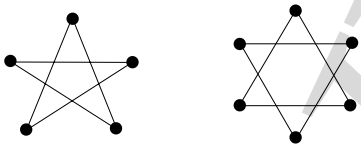

Bild 1.10: Der 5-eckige Stern ist zusammenhängend, der Davidstern aber nicht.

Ein Weg ist *einfach*, falls er keinen Knoten mehr als einmal durchläuft. Ein einfacher Weg mit $u_0 = u_r$ heißt *Kreis* oder *Zyklus*. Ein Graph $G = (V, E)$ heißt *zusammenhängend*, wenn es für zwei beliebige Knoten $u, v \in V$ mindestens einen Weg von $u$ nach $v$ gibt (Bild 1.10). Ein Graph heißt *zyklenfrei*, wenn er keine Kreise enthält.

Ein *Baum*, ist ein ungerichteter, zyklenfreier und zusammenhängender Graph. Ein typischer Baum ist in Bild 1.11 dargestellt und Bild 1.12 stellt alle nicht-isomorphen Bäume mit 5 Knoten dar.

In verschiedenen Anwendungsgebieten der Informatik spielen die Bäume eine herausragende Rolle. Mit diesen Graphen lassen sich zum Beispiel beschreiben: Syntaxbäume (siehe Bild 1.13), Datenstrukturen (Heaps, binäre Suchbäume, AVL-Bäum), Abstammungsbäume (Dateiorganisation in Unix Filesytem), usw.

Um ein reelles Problem mit Hilfe von Graphen zu lösen, muss man zunächst dieses Problem als ein graphentheoretisches Problem *modellieren*. Das ist ein wichtiger Schritt in jeder Anwendung. Wir wollen nun diesen Schritt an einem einfachen Beispiel demonstrieren.

**Beispiel 1.2: Ein Modellierungsbeispiel**

Unser Dekanat will einen Zeitplan für Klausuren erstellen. Das Ziel ist, alle Klausuren in so kürzer Zeit wie möglich zu organisieren. Man könnte alle Klausuren an einem Tag parallel durchführen. Das Problem ist dabei, dass einige Studierende an mehreren Klausuren teilnehmen wollen.

Dieses Problem kann man als ein *Färbungsproblem* für Graphen betrachten. Man nimmt für jede Klausur einen Knoten. Man verbindet zwei Knoten $u$ und $v$, falls es mindestens einen Studenten gibt, der an beiden Klausuren $u$ und $v$ teilnehmen will, und damit dürfen die beiden Klausuren nicht am selben Tag organisiert werden (Kanten entsprechen also Überschneidungen, die zu vermeiden sind).

Für jeden möglichen Klausurtag nimmt man eine Farbe z.B. gelb für Montag, grün für Dienstag, rot für Mittwoch, usw. Wir gehen hierbei davon aus, dass an jedem Tag beliebig viele Hörsäle zur Verfügung stehen.

Nun probiert man die Knoten mit diesen Farben so zu färben, dass *keine zwei*

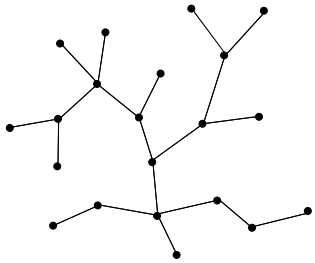

Bild 1.11: Ein typischer Baum.

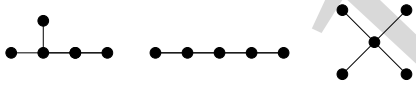

Bild 1.12: Alle nicht-isomorphen Bäume mit 5 Knoten.

*benachbarten Knoten dieselbe Farbe tragen*; man sagt dann, dass die Färbung *legal* ist. Da alle Klausuren in möglichst kurzer Zeit abgewickelt sein sollen, versucht das Dekanat den »Klausurgraphen« mit möglichst wenigen Farben zu färben.

Im Allgemeinen ist eine Färbung der Knoten eines Graphen $G = (V, E)$ *legal*, falls keine zwei adjazenten Knoten dieselbe Farbe tragen. Die kleinste Anzahl der Farben, für die es eine solche Färbung gibt, bezeichnet man mit $\chi(G)$ und nennt sie *chromatische Zahl* von $G$.

So kann man zum Beispiel jeden bipartiten Graphen mit nur 2 Farben färben. Der vollständige Graph $K_n$ benötigt aber bereits $n$ Farben.

Das Problem, den gegebenen Graphen $G$ mit $\chi(G)$ Farben zu färben, ist i. A. sehr schwer. Ist man aber mit »etwas« mehr als $\chi(G)$ Farben zufrieden, dann kann man den folgenden (sehr einfachen aber in der Praxis oft nützlichen) »gierigen« (engl. »greedy«) Algorithmus anwenden. Zuerst fixieren wir eine beliebige Reihenfolge der Knoten $v_1, \ldots, v_n$. Dann färbt der Algorithmus die Knoten gemäß dieser Reihenfolge:

1. Knoten $v_1$ erhält die Farbe 1.

2. Wenn $v_1, \ldots, v_k$ gefärbt sind, dann erhält $v_{k+1}$ die *kleinste* natürliche Zahl $j$ als Farbe, die noch nicht an einen Nachbarn von $v_{k+1}$ vergeben wurde.

Man kann sich leicht überzeugen, dass dieser Algorithmus für jeden Graphen eine legale Färbung mit höchstens $d + 1$ Farben findet, wobei $d$ der maximale Grad eines Knotens in $G$ ist. Angenommen, es sind bereits $d + 1$ Farben benutzt und der Algorithmus soll den Knoten $v$ färben. Wenn $v$ (laut Algorithmus) die Farbe $d + 2$ erhalten sollte, dann sollte $v$ mit $d + 1$ Knoten benachbart sein. Dies ist aber unmöglich, da jeder Knoten den Grad höchstens $d$ hat.

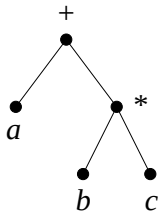


Bild 1.13: Ein Syntaxbaum für den Ausdruck $a + b * c$.

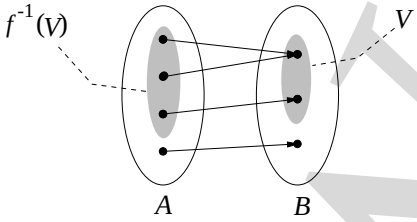


Bild 1.14: Das Urbild von $V$ unter $f$.

## 1.3 Abbildungen (Funktionen)

Eine Relation $f \subseteq A \times B$ derart, dass für jedes $a \in A$ genau ein Element $b \in B$ mit $(a, b) \in f$ gibt, heißt *Abbildung* (oder *Funktion*) von $A$ nach $B$; $A$ ist der *Definitionsbereich* der Abbildung und $B$ ihr *Bildbereich* (oder *Wertebereich*). Bei einer Abbildung wird also jedem Element $a$ aus $A$ in einer eindeutigen Weise das Element $b = f(a)$ in $B$ zugeordnet.

Eine Abbildung können wir uns als »black box« $x \mapsto f(x)$ vorstellen, in die wir etwas hineinstecken und dafür etwas Neues herausbekommen. Beispiel: $x \mapsto x^2$ ergibt das Quadrat einer Zahl, $A \mapsto |A|$ die Mächtigkeit einer Menge und $x \mapsto |x|$ den Betrag einer Zahl. Bei einer Funktion ist es wichtig zu wissen, was man hineinstecken darf und aus welchem Bereich das Ergebnis ist. Wir schreiben

$$f : A \to B \quad \text{oder} \quad A \xrightarrow{f} B$$

um anzuzeigen, dass die Funktion $f$ Eingaben aus der Menge $A$ akzeptiert und die Ausgabe $f(x)$ zu der Menge $B$ gehört. Die Menge aller Abbildungen von $A$ nach $B$ bezeichnet man oft mit $B^A$. Für $U \subseteq A$ heißt

$$f(U) = \{f(a) \colon a \in U\}$$

*Bild* von $U$ unter $f$. Für $V \subseteq B$ heißt

$$f^{-1}(V) = \{a \in A \colon f(a) \in V\}$$

*Urbild* von $V$ unter $f$ (siehe Bild 1.14). Für $b \in B$ setzt man

$$f^{-1}(b) = \{a \in A \colon f(a) = b\}\,.$$