Algebraische und kombinatorische Anwendungen in der Informatik

Skript

zur 2-std. Vorlesung im SS 2014

Prof. Dr. Peter Hauck

Arbeitsbereich Diskrete Mathematik

Fachbereich Informatik

Universität Tübingen

LaTeX-Fassung von Dr. Anni Neumann

Inhaltsverzeichnis

1 Abzählmethoden

1.1 Summen- und Produktregel

$S_1, \dots S_n$ seien endliche Mengen.

a) (**Summenregel**) Sind die endlichen Mengen $S_1, \dots, S_n$ paarweise disjunkt, so gilt $|S_1 \cup \dots \cup S_n| = \sum_{i=1}^{n} |S_i|$.

b) (**Produktregel**) Es gilt: $|S_1 \times \dots \times S_n| = \prod_{i=1}^{n} |S_i|$;
dabei setzen wir $S_1 \times \dots \times S_n := \{(s_1, \dots, s_n) : s_i \in S_i \text{ für } i = 1, \dots, n\}$.

1.2 Beispiele

a) Wählt man in 1.1 b) $S_1 = \dots = S_n = \{0, 1\}$, so erhält man:

Es gibt 2^n Wörter der Länge n über $\{0, 1\}$.

b) Wie viele Wörter x der Länge 3 aus dem Alphabet $\{a, b, \dots, z\}$ gibt es, die **mindestens** ein e enthalten?

Wir zeigen drei Möglichkeiten auf, diese Frage zu beantworten:

1. Möglichkeit: Wir setzen jeweils

$S_1 = \{x$: e steht an 1. Stelle von $x\}$, $S_2 = \{x$: erstes e steht an 2. Stelle von $x\}$, $S_3 = \{x$: erstes e steht an 3. Stelle von $x\}$.

Dann erhalten wir mit 1.1 b):

$|S_1| = 1 \cdot 26 \cdot 26 = 676$, $|S_2| = 25 \cdot 1 \cdot 26 = 650$ und $|S_3| = 25 \cdot 25 \cdot 1 = 625$.

Da S_1, S_2, S_3 paarweise disjunkt sind, ergibt sich mit 1.1 a) für die gesuchte Anzahl $|S_1 \cup S_2 \cup S_3| = |S_1| + |S_2| + |S_3| = 1951$.

2. Möglichkeit: Zunächst setzen wir

$T_i = \{x\text{: } x \text{ enthält } e \text{ genau } i\text{-mal}\}, i = 1, 2, 3$. Offensichtlich ist $T_3 = \{eee\},\ |T_3| = 1$. Weiter setzen wir im Folgenden $T_1^j = \{x \in T_1\text{: } e \text{ steht an der Stelle } j\}$ und $T_2^j = \{x \in T_2\text{: an der Stelle } j \text{ steht kein } e\}, j = 1, 2, 3$.

Es gilt $T_1 = T_1^1 \dot\cup T_1^2 \dot\cup T_1^3$ und $T_2 = T_2^1 \dot\cup T_2^2 \dot\cup T_2^3$. Daraus erhalten wir jeweils mit 1.1.a) $T_1 = 3 \cdot 25^2 = 1875$ bzw. $T_2 = 3 \cdot 25 = 75$. Dabei bedeutet $\dot\cup$ die disjunkte Vereinigung.

Die T_1, T_2, T_3 sind paarweise disjunkt, und damit erhalten wir analog wie in der 1.Möglichkeit wegen 1.1. a) für die gesuchte Anzahl $|T_1 \cup T_2 \cup T_3| = |T_1| + |T_2| + |T_3| = 1875 + 75 + 1 = 1951$.

3. Möglichkeit: Die gesuchte Anzahl kann man auch so bestimmen:

Anzahl aller Wörter der Länge 3 − Anzahl aller Wörter der Länge 3 ohne e = $26^3 - 25^3 = 1951$.

Dabei haben wir jeweils 1.1. b) angewandt.

Es stellt sich die Frage, inwieweit sich 1.1 a) verallgemeinern lässt, wenn die Mengen nicht paarweise disjunkt sind.
Es gilt folgende Aussage:

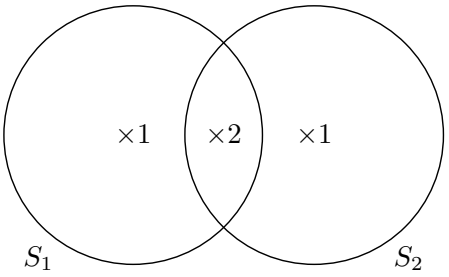
1.3 Einschließungs-Ausschließungsprinzip

$S_1, \dots S_n$ seien endliche Mengen. Dann gilt:

$$|S_1 \cup \dots \cup S_n| = \sum_{k=1}^{n} (-1)^{k+1} \sum_{1 \leqq i_1 < \dots < i_k \leqq n} |S_{i_1} \cap \dots \cap S_{i_k}|.$$

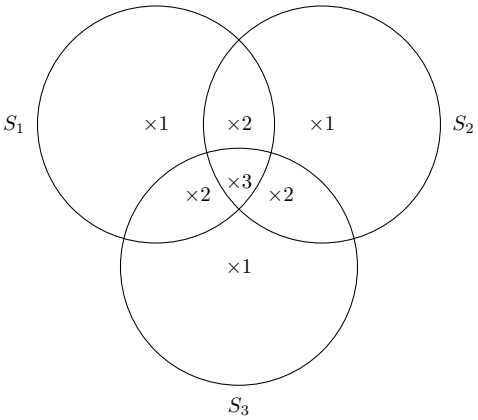
Veranschaulichung

Skizze für den Fall $n = 2$:



$$|S_1 \cup S_2| \quad = \quad |S_1| + |S_2| - |S_1 \cap S_2|$$

Skizze für den Fall $n = 3$:



$$\begin{aligned} |S_1 \cup S_2 \cup S_3| \quad &= \quad |S_1| + |S_2| + |S_3| \\ &- \quad |S_1 \cap S_2| - |S_1 \cap S_3| - |S_2 \cap S_3| \\ &+ \quad |S_1 \cap S_2 \cap S_3| \end{aligned}$$

Dieses Prinzip wird allgemein durch vollständige Induktion nach n bewiesen (vgl. z.B. [WHK], Kapitel 2.3, Satz 2.32)

1.4 Beispiel

Wie viele Wörter über $\{a, \ldots, z\}$ der Länge 10 gibt es, die nicht alle Vokale enthalten? Dazu betrachten wir die Menge S_a aller Wörter der Länge 10 ohne a; entsprechend definieren wir die Mengen S_e, S_i, S_o und S_u. Dann gilt offenbar $|S_a| = 25^{10}$ und Entsprechendes gilt auch für $|S_e|, |S_i|, |S_o|$ und $|S_u|$. Weiter gilt $|S_a \cap S_e| = 24^{10}$; Entsprechendes gilt auch für die anderen zehn derartiger Mengen $|S_a \cap S_i|, \ldots, |S_o \cap S_u|$. Ferner gilt $|S_a \cap S_e \cap S_i| = 23^{10}$ und Entsprechendes für die restlichen neun Kombinationen. Wegen $|S_a \cap S_e \cap S_i \cap S_o| = 22^{10}$ und den entsprechenden weiteren vier Kombinationen und

letztlich wegen $|S_a \cap S_e \cap S_i \cap S_o \cap S_u| = 21^{10}$ erhalten wir dann mit Hilfe von 1.3:

$|S_a \cup S_e \cup S_i \cup S_o \cup S_u| = 5 \cdot 25^{10} - 10 \cdot 24^{10} + 10 \cdot 23^{10} - 5 \cdot 22^{10} + 21^{10} = 140.948.727.706.936$

1.5 Prinzip des doppelten Abzählens

A, B seien endliche Mengen und $S \subseteq A \times B$.

Für jedes $a \in A$ sei $n_a = |\{b \in B\colon (a,b) \in S\}|$, d.h. n_a ist die Anzahl der Elemente b in B, für die das Paar (a,b) in S liegt. Und analog definieren wir für jedes $b \in B$ die Zahl $m_b = |\{a \in A\colon (a,b) \in S\}|$. Dann gilt:

$$\sum_{a \in A} n_a = \sum_{b \in B} m_b = |S|.$$

Speziell erhalten wir, falls $n = n_a$ unabhängig von a und $m = m_b$ unabhängig von b ist, die Gleichheit $|A| \cdot n = |B| \cdot m$.

Eine Anwendung ist die Berechnung von m, falls $|A|, |B|, n$ bekannt sind.

1.6 Beispiel

Gegeben ist ein Graph ohne Schleifen. Dann gilt:

Die Anzahl der Knoten (vertex, pl. vertices), von denen eine ungerade Anzahl von Kanten (edge) ausgeht, ist gerade, d.h. Jeder Graph ohne Schleifen hat eine gerade Anzahl von Knoten ungeraden Grades. (Handshaking-Lemma)

Zum Nachweis dieser Aussage bilden wir die Mengen $V =$ Menge aller Knoten und $E =$ Menge aller Kanten.

Wir setzen $S = \{(v,e)\colon v$ ist Knoten, e ist Kante, v ist Endknoten von $e\}$. Vom Knoten v gehen n_v Kanten aus (man nennt n_v auch den Grad von v), und jede Kante hat 2 Endknoten. Zunächst erhalten wir mit 1.5 die Gleichheit

$$(\circledast) \qquad |S| = \sum_{v \in V} n_v \underset{1.5}{=} \sum_{e \in E} 2 = 2 \cdot |E|.$$

Wir zerlegen die Indexmenge V disjunkt: $V = V_g \dot\cup V_u$, wobei V_g alle Knoten v von geradem Grad n_v umfasst; analog umfasst V_u alle Knoten ungeraden Grades.

Damit erhalten wir:

Weil $\sum_{v \in V_g} n_v$ und weil auch wegen ($\circledast$) die Summe $\sum_{v \in V_g} n_v + \sum_{v \in V_u} n_v$ gerade sind, ist dann $\sum_{v \in V_u} n_v$ gerade. Weil aber für $v \in V_u$ der jeweilige Grad n_v ungerade ist, ist dann zwingend $|V_u|$ gerade.

1.7 Schubfachprinzip

Werden m Objekte auf n "Schubfächer" verteilt, so enthält ein "Schubfach" mindestens $\lceil \frac{m}{n} \rceil$ Objekte.

Spezialfall:

Es sei $m > n$. Dann gilt: Mindestens ein Schubfach enthält mehr als ein Objekt.

Beweis:

Es gilt $\lceil \frac{m}{n} \rceil - 1 < \frac{m}{n}$. Angenommen, die Aussage ist falsch. Dann erhalten wir:
$m \leq n(\lceil \frac{m}{n} \rceil - 1) < n \cdot \frac{m}{n} = m$, ein Widerspruch.

1.8 Beispiele

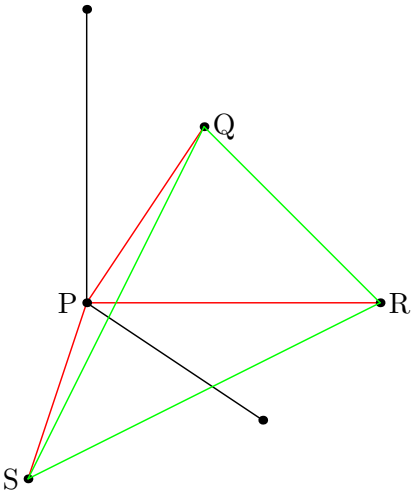
a) Gegeben sei ein Quadrat der Seitenlänge 70cm. Kann man 50 Punkte so im Quadrat verteilen, dass je 2 Punkte mindestens 15cm Abstand haben?

   Um diese Frage zu beantworten, gehen wir folgendermaßen vor: Wir teilen das gegebene Quadrat in 49 Teilquadrate der Seitenlänge 10cm. Nach 1.7 gibt es ein Teilquadrat, das mindestens 2 Punkte enthält. Größtmöglicher Abstand zweier Punkte in solch einem Quadrat ist die Länge der Diagonale, also $\sqrt{2} \cdot 10 = \sqrt{200} \approx 14,14$cm. Die Antwort lautet also nein!

   Schwieriger ist die Frage: Wie viele Punkte lassen sich unter den angegebenen Bedingungen im Quadrat unterbringen?

b) Gegeben sind 6 Punkte in der Ebene, keine 3 Punkte auf einer Geraden. Je 2 Punkte sind durch eine Kante verbunden, wir haben also einen vollständigen Graph. Die Kanten werden jeweils mit einer der Farben rot oder grün gefärbt.

Dann existiert ein einfarbiges Dreieck:



Vom Punkt P gehen 5 Kanten aus. Mindestens drei dieser Kanten PQ, PR, PS haben nach 1.7 dieselbe Farbe, z.B. rot. Ist eine der Kanten QR, QS, RS rot, z.B. QR, haben wir ein rotes Dreieck PQR. Ansonsten bilden dann Q, S, R ein grünes Dreieck.

Diese Aussage ist ein Spezialfall des **Satzes von Ramsey**:

Gegeben ist ein vollständiger Graph, d.h. alle Knoten sind durch Kanten verbunden. Die Kanten werden mit f Farben gefärbt. Wähle $r \in \mathbb{N}$. Dann gilt:

Ist die Anzahl der Knoten genügend groß, dann existieren r Knoten, deren sämtliche Verbindungskanten dieselbe Farbe haben.

In unserem obigen Fall ist $r = 3, f = 2$, und die Anzahl der Knoten ist mindestens 6 (6 ist die kleinstmögliche Zahl).

Elementare Abzählprobleme (vgl. Mathematik für Informatiker I)

Wiederholung:

Für $n, k \in \mathbb{N}_0 := \{0, 1, 2, \dots\}$ setzen wir:

- $[n]_0 = 1, [n]_k = n \cdot (n-1) \cdot \dots \cdot (n-k+1)$

  Insbesondere gilt: $[n]_k = 0$ für $k > n$)

- $n! := [n]_n = 1 \cdot 2 \cdot \dots \cdot n$; $0! = 1$

  $n!$ nennt man "n **Fakultät**"

- $\binom{n}{k} = \begin{cases} 0 & \text{für } k > n \\ \frac{n!}{k!(n-k)!} & \text{für } k \leq n \end{cases}$

$\binom{n}{k}$ nennt man **Binomialkoeffizient** und sagt "n über k".

- Wichtig sind die Beziehungen:
  $\binom{n}{k} = \binom{n}{n-k}$ für $k \leq n$
  $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ für $n, k \geq 1$.
  (Pascalsches Dreieck!)

1.9 Satz

a) Seien $k, n \in \mathbb{N} := \{1, 2, 3, \dots\}$.

Die Anzahl der Möglichkeiten aus einer Menge mit n Elementen k Elemente auszuwählen ist:

k aus n	ohne Berücksichtigung der Anordnung	mit Berücksichtigung der Anordnung
ohne Wiederholungen	$\binom{n}{k}$	$[n]_k$
mit Wiederholungen	$\binom{n+k-1}{k}$	n^k

b) Seien $k, n \in \mathbb{N}$.

Die Anzahl der geordneten n-Tupel $(x_1, \dots, x_n), x_i \in \mathbb{N}_0$, mit $\sum_{i=1}^{n} x_i = k$ ist $\binom{n+k-1}{k}$.

c) Seien $k, n \in \mathbb{N}$.

Die Anzahl der geordneten n-Tupel $(x_1, \dots, x_n), x_i \in \mathbb{N}$, mit $\sum_{i=1}^{n} x_i = k$ ist $\binom{k-1}{n-1}$.

d) Nach a) gilt: Die Anzahl aller k-elementigen Teilmengen einer Menge mit n Elementen ist $\binom{n}{k}$. Wir erinnern noch an den **Binomialsatz**: $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$.
Damit können wir dann die Anzahl aller Teilmengen einer Menge mit n Elementen bestimmen: $\sum_{k=0}^{n} \binom{n}{k} = (1+1)^n = 2^n$.

1.10 Beispiele

a) Wir betrachten das Poker-Spiel: Dieses Spiel hat je 13 Karten Karo, Herz, Pik und Kreuz. Wir bestimmen die Wahrscheinlichkeit fünf Karten von einer Sorte zu ziehen: Es gibt $\binom{52}{5}$ Möglichkeiten aus den 52 Pokerkarten 5 Karten zu ziehen, und es gibt für jede der 4 Kartensorten $\binom{13}{5}$ Möglichkeiten 5 Karten auszuwählen. Damit erhalten wir für die Wahrscheinlichkeit: $\frac{4 \cdot \binom{13}{5}}{\binom{52}{5}} = \frac{33}{16660} \approx 0,00198$.

b) Wir betrachten wieder das Poker-Spiel aus a). Wir bestimmen die Wahrscheinlichkeit 5 Karten mit genau 3 Assen zu erhalten. Wir erhalten: die Wahrscheinlichkeit ist: $\frac{\binom{4}{3}\binom{48}{2}}{\binom{52}{5}} = \frac{4512}{2598960} = 0,001736$.

c) Wie viele Möglichkeiten gibt es 12 identische Kugeln auf 8 unterschiedliche Kisten zu verteilen? Dabei dürfen Kisten auch leer bleiben.

Diese gesuchte Anzahl ist gleich der Anzahl aus 8 Kisten 12 Kugeln (mit Wiederholung und ohne Berücksichtigung der Anordnung) auszuwählen. Zur Bestimmung wenden wir 1.9a) an:

$$\binom{n+k-1}{k} \underset{\substack{n=8\\k=12}}{=} \binom{19}{12} = 50388 \underset{1.9b)}{=} |\{(x_1,\ldots,x_8)\colon x_i \geq 0, \sum_{i=1}^{8} x_i = 12\}|.$$

d) Wir betrachten nochmals die Aufgabenstellung wie in c), nur dass nun keine Kiste leer bleiben darf. Wir erhalten dann:

$$|\{(x_1,\ldots,x_8)\colon x_i \geq 1, \sum_{i=1}^{8} x_i = 12\}| \underset{1.9c)}{=} \binom{11}{7} = 330.$$

Wir können diese Aufgabe auch anders lösen:

Verteile je eine Kugel auf jede Kiste. Dann wende c) an mit $k = 4, n = 8$:

$\binom{11}{4} = \binom{11}{7} = 330$.

1.11 Satz

Die Anzahl der Möglichkeiten k unterschiedliche bzw. identische Objekte auf n unterschiedliche bzw. identische Kisten zu verteilen, wobei Kisten auch leer bleiben dürfen, ist:

	n identische Kisten	n unterschiedliche Kisten
k identische Objekte	$p_n(k)$	$\binom{n+k-1}{k}$
k unterschiedliche Objekte	$\sum_{i=1}^{n} S(k,i)$	n^k

Dazu definieren wir:

1.12 Definition

a) Sei A sei eine Menge. Unter einer (endlichen) **Partition** von A versteht man eine Auswahl von endlich vielen Teilmengen $A_1, \ldots, A_r$ von A mit den Eigenschaften: $A = A_1 \cup \cdots \cup A_r$, $A_i \neq \emptyset$ für $i = 1, \ldots r$ und $A_i \cap A_j = \emptyset$ für alle i, j mit $i \neq j$; dabei ist die Reihenfolge der Mengen $A_1, \ldots, A_r$ unerheblich.

Beispiel: Es sei A eine Menge mit $|A| = 4$ und es sei $r = 2$. Dann haben wir insgesamt 7 Partitionen mit zwei Teilmengen von A:

$\{1\} \cup \{2,3,4\}, \ldots, \{4\} \cup \{1,2,3\}$
$\{1,2\} \cup \{3,4\}, \{1,3\} \cup \{2,4\}, \{1,4\} \cup \{2,3\}$.

b) Die Anzahl aller Partitionen einer Menge mit k Elementen in r Teilmengen bezeichnen wir mit $S(k,r)$. Diese Zahlen nennt man auch **Stirlingzahlen 2.Art** (benannt nach James Stirling, 1692-1770).

Nach dem Beispiel in a) ist also $S(4,2) = 7$.

c) Die Anzahl aller Partitionen einer Menge mit k Elementen wird mit B_k bezeichnet und es gilt: $B_k = \sum_{i=1}^{k} S(k,i)$. Diese Zahlen B_k nennt man **Bell-Zahlen** (benannt nach E.T. Bell, 1883-1960).

d) $p_r(k)$ bezeichne die Anzahl der Zerlegungen von $k \in \mathbb{N}_0$ in r Summanden aus $\mathbb{N}_0$ ohne Berücksichtigung der Reihenfolge der Summanden. Es gilt also:

$p_r(k) = |\{(x_1, \ldots, x_r)\colon x_i \in \mathbb{N}_0, x_1 \geq x_2 \geq \cdots \geq x_r, \sum_{i=1}^{r} x_i = k\}|$.

Die Zahlen $p_r(k)$ nennt man **Partitionszahlen**.

Beispiele:

Es gilt $p_2(4) = 3$, denn wir haben folgende Zerlegungen von 4 in zwei Summanden: 4+0, 3+1, 2+2.

Es gilt $p_3(4) = 4$, denn in diesem Fall haben wir die folgenden Zerlegungen von 4 in drei Summanden: 4+0+0, 3+1+0, 2+2+0, 2+1+1.

Beweis von 1.11:

a) "k unterschiedliche Objekte auf n unterschiedliche Kisten verteilen":

Es gibt n Möglichkeiten für jedes Objekt, also insgesamt n^k Möglichkeiten

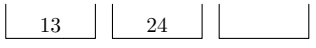
b) "k identische Objekte auf n unterschiedliche Kisten verteilen":

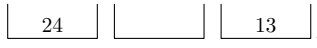
Wähle k Kisten mit Wiederholung, ohne Berücksichtigung der Anordnung (vgl. Beispiel 1.10c)). Nach 1.9.a) ist die Anzahl gegeben durch $\binom{n+k-1}{k}$.

c) "k unterschiedliche Objekte auf n identische Kisten verteilen":

Diese Anzahl ist die Anzahl aller Partitionen einer Menge mit k .Elementen in höchstens n Teilmengen und diese wiederum ist $\sum_{i=1}^{n} S(k, i)$.

Beispiel: $k = 4, n = 3$.



ist dasselbe wie


.

d) "k identische Objekte auf n identische Kisten verteilen":

Eine Kiste enthält x_1 Objekte, eine enthält x_2 Objekte, ... , eine Kiste enthält x_n Objekte, wobei $\sum_{i=1}^{n} x_i = k, x_i \in \mathbb{N}_0$. Also gilt:

Anzahl $= |\{(x_1, \ldots, x_n)\colon x_i \in \mathbb{N}_0, x_1 \geq x_2 \geq \cdots \geq x_n, \sum_{i=1}^{n} x_i = k\}| = p_n(k)$.

Beispiel: (7,4,2,0) als mit $n = 4, k = 13$:

ist dasselbe wie

1.13 Satz

Seien A, B Mengen mit $|A| = k, |B| = n$.

a) Die Anzahl aller Abbildungen $A \longrightarrow B$ ist n^k.

b) Die Anzahl aller injektiven Abbildungen $A \longrightarrow B$ ist $[n]_k$.

c) Die Anzahl aller bijektiven Abbildungen $A \longrightarrow B$ ist 0, falls $n \neq k$, sonst $n!$.

d) Die Anzahl aller surjektiven Abbildungen $A \longrightarrow B$ ist $S(k, n) \cdot n!$.

Beweis:

a) Wir wenden die Tabelle 1.9 auf den Fall "k aus n mit Wiederholung mit Berücksichtigung der Anordnung an" und erhalten dann, dass es n^k derartige Abbildungen gibt. Eine Skizze veranschaulicht noch einmal diese Situation:

$a_1, \ldots, a_k$ $\quad \leftarrow$ Elemente von A in fester Reihenfolge

$\downarrow \ \ldots \ \downarrow$

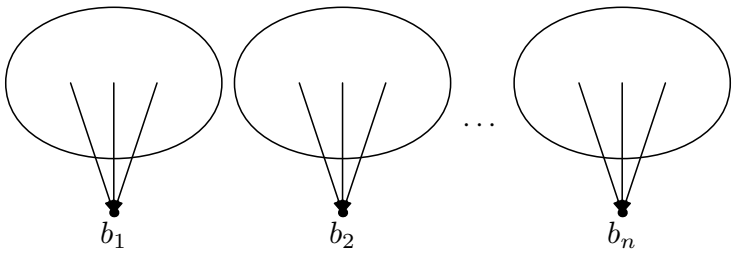
$b_1, \ldots, b_k$ $\quad \leftarrow$ jedem $a_i \in A$ kann ein beliebiges $b_i \in B$ zugeordnet werden.

b) Im Fall b) können wir analog wie in a) argumentieren, also hier im Fall "k aus n ohne Wiederholung mit Berücksichtigung der Anordnung", so dass es $[n]_k$ derartige Abbildungen gibt.

c) Für den Nachweis reicht die Bemerkung, dass $[n]_n = n!$.

d) Wir bilden Partitionen von A in n nicht-leere Teilmengen (nach 1.12 ist die Anzahl gleich $S(k,n)$). Die gewählten Teilmengen entsprechen den Urbildern der Elemente aus B.

Zur Veranschaulichung:



$\leftarrow$ Partitionen von A

$\leftarrow$ $n!$ Möglichkeiten der Zuordnung zu den Elementen von B.

Dann folgt die Behauptung.

Die Anzahl der surjektiven Abbildungen lässt sich explizit angeben, woraus dann mit 1.13.d) eine Formel für $S(k,n)$ folgt.

1.14 Satz

Seien A, B Mengen mit $|A| = k, |B| = n$. Dann gilt: Die Anzahl aller surjektiven Abbildungen $A \longrightarrow B$ ist gegeben durch die Formel $\sum_{j=0}^{n-1} (-1)^j \cdot \binom{n}{j} \cdot (n-j)^k$.

Beweis:

Sei $B = \{b_1, \ldots, b_n\}$. Wir setzen $T_i = \{\alpha : A \longrightarrow B\colon b_i \notin \alpha(A)\}, i = 1, \cdots, n$. S sei die Menge aller surjektiven Abbildungen $A \longrightarrow B$, T sei die Menge aller Abbildungen $A \longrightarrow B$. Dann gilt: $|S| = |T| - |T_1 \cup \cdots \cup T_n|$. Nach 1.13a) gilt $|T| = n^k$.

Mit Hilfe von 1.3 bestimmen wir im Folgenden $|T_1 \cup \cdots \cup T_n|$.

Nach 1.13.a) erhalten wir $|T_i| = (n-1)^k$ mit $\binom{n}{1} = n$ Möglichkeiten für i $(i = 1, \ldots n)$.

Für $i < j$ erhalten wir weiterhin $|T_i \cap T_j| = (n-2)^k$ mit $\binom{n}{2}$ Möglichkeiten für i, j.

Und für $i < j < k$ ist $|T_i \cap T_j \cap T_k| = (n-3)^k$ mit $\binom{n}{3}$ Möglichkeiten für $i, j, k, \ldots$ und

$|T_1 \cap \cdots \cap T_n| = (n-n)^k = 0$ für $\binom{n}{n} = 1$ Möglichkeit, jeweils nach 1.13.a).

Nach 1.3 gilt:

$$|T_1 \cup \cdots \cup T_n| = \sum_{j=1}^{n} (-1)^{j+1} \sum_{1 \le i_1 < \ldots i_j \le n} |T_{i_1} \cap \cdots \cap T_{i_j}| = \sum_{j=1}^{n} (-1)^{j+1} \binom{n}{j} (n-j)^k.$$

Und damit erhalten wir dann:

$$|S| = n^k - \sum_{j=1}^{n} (-1)^{j+1} \binom{n}{j} (n-j)^k = \sum_{j=0}^{n} (-1)^{j} \binom{n}{j} (n-j)^k = \sum_{j=0}^{n-1} (-1)^{j} \binom{n}{j} (n-j)^k.$$

1.15 Korollar

Es gilt $S(k,n) = \dfrac{1}{n!} \displaystyle\sum_{j=0}^{n-1} (-1)^j \binom{n}{j} (n-j)^k$.

Beweis: Folgt aus 1.13.d) und 1.14.

1.16 Korollar

a) $\displaystyle\sum_{j=0}^{n-1} (-1)^j \binom{n}{j} (n-j)^n = n!$

b) Ist $n > k$, so ist $\displaystyle\sum_{j=0}^{n-1} (-1)^j \binom{n}{j} (n-j)^k = 0$

Beweis: 1.13.d) und 1.14.

2 Lineare Rekursionen

2.1 Definition

a) Eine **Rekursion** ist ein unendliches System von Gleichungen der Form

($\star$) $x_n = f_n(x_{n-1}, \dots, x_1)$ für alle $n \geq n_0$.

Dabei sind f_n berechenbare Funktionen.

Bei Vorgabe von beliebigen **Anfangswerten** $x_1 = a_1, \dots, x_{n_0-1} = a_{n_0-1}$ sind $a_{n_0} = f_{n_0}(a_{n_0-1}, \dots, a_1), a_{n_0+1} = f_{n_0+1}(a_{n_0}, \dots, a_1) \dots$ festgelegt. Dabei nehmen wir an, dass die f_n überall definiert sind.

Jede Folge $(x_1, x_2, \dots)$ von Elementen aus $\mathbb{R}$ (oder $\mathbb{C}$), die ($\star$) erfüllt, heißt **Lösung** der Rekursion.

Beispiel:

$$x_n = x_{n-1}^2 + 2x_{n-2} \cdot x_1 \text{ für } n \geq 3$$
$$a_1 = 1,\ a_2 = 2 \text{ Anfangswerte}$$
$$a_3 = 6,\ a_4 = 40, \dots$$

b) Eine **Rekursion k-ter Ordnung** ist ein unendliches System von Gleichungen der Form:

$x_n = f_n(x_{n-1}, \dots, x_{n-k})$ für alle $n \geq k+1$.

c) Eine Rekursion heißt **linear von Ordnung k mit konstanten Koeffizienten**, falls gilt:

($\star\star$) $x_n = c_1 \cdot x_{n-1} + \dots + c_k \cdot x_{n-k} + g(n)$ für alle $n \geq k+1, c_i \in \mathbb{R}$ (oder $\mathbb{C}$) fest, d.h. unabhängig von n, $c_k \neq 0$.

Ist $g(n) = 0$ für alle $n \geq k+1$, so heißt die Rekursion **homogen**, sonst **inhomogen**.

2.2 Beispiele

a) **Fibonacci-Zahlen** (Leonardo von Pisa, $\sim$ 1170 - $\sim$ 1250, Liber abaci (Das Buch vom Abakus), Dezimalsystem)

Auf wie viele Arten kann $n \in \mathbb{N}$ als geordnete Summe von Einsen und Zweien geschrieben werden? Im Folgenden bezeichnen wir diese gesuchte Anzahl mit F_n.

1=1, also $F_1 = 1$

2=1+1, 2=2, also $F_2 = 2$

3=1+1+1+1, 3= 1+2, 3= 2+1, also $F_3 = 3$

4=1+1+1+1+1, 4=1+1+2, 4= 1+2+1, 4= 2+1+1, 4=2+2, also $F_n = 5$

Allgemein überlegen wir uns für $n \geq 3$, wie sich F_n aus F_{n-1} und F_{n-2} darstellen lässt:

Eine geordnete Summe für n endet entweder auf 1 oder 2. Endet sie auf 1, so steht davor eine geordnete Summe für $n-1$, endet sie auf 2, so steht davor eine geordnete Summe für $n-2$. Also

$$\text{für } n \geq 3 : \underbrace{\ldots}_{n-1} + 1, \qquad \underbrace{\ldots}_{n-2} + 2.$$

Diese Überlegung führt zur linearen homogenen Rekursion mit konstanten Koeffizienten der Ordnung 2: $F_n = F_{n-1} + F_{n-2}$ für alle $n \geq 3$ mit den Anfangsbedingungen $F_1 = 1$, $F_2 = 2$.

Ein anderes **Beispiel** führt auf ein ähnliches Ergebnis:

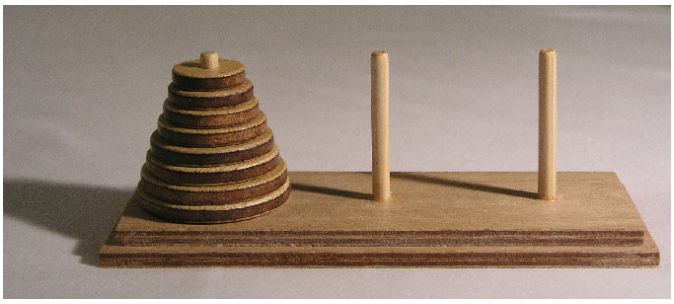
Wie viele 0-1-Folgen der Länge n gibt es ohne zwei aufeinanderfolgende Nullen?

Diese gesuchte Anzahl nennen wir a_n. Klar ist: $a_1 = 2, a_2 = 3$. Eine analoge Überlegung wie im obigen Beispiel liefert:

$$\text{für } n \geq 3 : \underbrace{\ldots}_{a_{n-1}} 1, \qquad \underbrace{\ldots}_{a_{n-2}} 10.$$

Diese Überlegung führt dann zur linearen homogenen Rekursion mit konstanten Koeffizienten der Ordnung 2: $a_n = a_{n-1} + a_{n-2}$ für alle $n \geq 3$ mit den Anfangsbedingungen $a_1 = 2, a_2 = 3$. Man sieht: $a_i = F_{i+1}, i = 1, \ldots$

b) Ein klassisches Beispiel sind die **Türme von Hanoi**.



© Wikipedia

Im Folgenden sind die Stäbe auf dem Bild von links nach rechts mit 1,2 und 3 nummeriert. Ziel ist es, den gesamten Turm effizient, d.h. mit der geringsten Zahl von Scheibenbewegungen, auf den Stab 2 umzulegen. Dabei ist die Regel zu beachten, dass nie eine größere Scheibe auf einer kleineren Schreibe zu liegen kommt, wobei immer nur genau eine Scheibe umgelegt werden darf. Dazu muss man effizient $n-1$ Scheiben vom Stab 1 auf den Stab 3 umlegen und dann die größte Scheibe vom Stab 1 auf den Stab 2 umlegen. Dann hat man wiederum effizient die $n-1$ Scheiben entsprechend der Regeln vom Stab 3 auf den Stab 2 umzulegen.

Bezeichnen wir mit a_n die Minimalanzahl von Scheibenbewegungen, um einen Turm mit n Scheiben von Stab 1 auf Stab 2 umzulegen, so erhalten wir eine lineare inhomogene Rekursion der Ordnung 1:

$$a_1 = 1, a_n = 2 \cdot a_{n-1} + 1 \text{ für alle } n \geq 2.$$

c) Als Verallgemeinerung von b): **Divide-and-Conquer-Algorithmen**:

Zerlege ein Problem der Größe n in a Teilprobleme der Größe $n-1$ (siehe z.B. b); hier ist $a = 2$). Dann ergibt sich für den Aufwand $T(n)$ die Rekursion:

$$T(n) = a \cdot T(n-1) + g(n).$$

Dabei beschreibt $g(n)$ den Aufwand für das Zerlegen in Teilprobleme und das Zusammensetzen der Lösungen.

Im Folgenden sei $K = \mathbb{R}$ oder $K = \mathbb{C}$.
Wir bezeichnen mit $K^{\mathbb{N}}$ den K-Vektorraum aller unendlichen Folgen $(b_1, b_2, \dots)$ mit $b_i \in K$. Gegeben ist die lineare Rekursion mit konstanten Koeffizienten der Ordnung k:

$$x_n = c_1 \cdot x_{n-1} + \cdots + c_k \cdot x_{n-k} + g(n) \text{ für alle } n > k.$$

Zu gegebenem $(a_1, \dots, a_n) \in K^k$ existiert genau eine Lösung $(x_1, x_2, \dots) \in K^{\mathbb{N}}$ mit $x_i = a_i, i = 1, \dots, k$.
Wir setzen:

$$l : \begin{cases} K^k \longrightarrow K^{\mathbb{N}} \\ a = (a_1, \dots, a_k) \longrightarrow l(a) = \text{Lösung mit Anfangswerten } a_1 \dots, a_k. \end{cases}$$

Beispiel:

$$x_n = 2 \cdot x_{n-1} + 3x_{n-3}, n > 3, k = 3, a = (0, 1, 2)$$
$$l(a) = (0, 1, 2, 4, 11, 28, 68, \dots)$$

2.3 Satz

a) Gegeben sei eine homogene lineare Rekursion (R_h) mit konstanten Koeffizienten der Ordnung k:

$$x_n = c_1 \cdot x_{n-1} + \cdots + c_k \cdot x_{n-k} \text{ für alle } n > k, c_k \neq 0.$$

Die Lösungen von (R_h) bilden einen Unterraum L von $K^{\mathbb{N}}$ der Dimension k; $l : K^k \longrightarrow L \subseteq K^{\mathbb{N}}$ ist ein Vektorraum-Isomorphismus.

b) Gegeben sei eine inhomogene lineare Rekursion (R) mit konstanten Koeffizienten der Ordnung k:

$$x_n = c_1 \cdot x_{n-1} + \cdots + c_k \cdot x_{n-k} + g(n) \text{ für alle } n > k, c_k \neq 0.$$

(R_h) sei die zugehörige homogene Rekursion.

Ist $b = (b_1, b_2, \dots)$ irgendeine spezielle Lösung von (R), so ist die Menge aller Lösungen von (R) gegeben durch $b + L = \{b_1 + y_1, b_2 + y_2, \dots : (y_1, y_2, \dots) \in L\}$, L Lösungsraum von (R_h).

Beweis: Wie in der Linearen Algebra für lineare Gleichungssysteme.

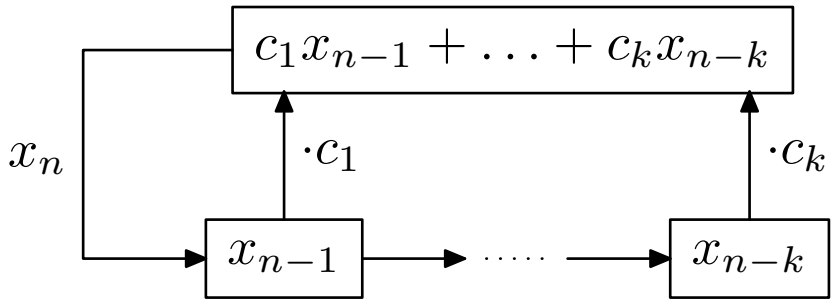
Wir gehen nun der Frage nach: Wie kommt man zu einer geschlossenen Beschreibung der Lösungen von (R_h)?
Wir betrachten zunächst homogene lineare Rekursionen:
Eine Basis von L bilden nach 2.3 z.B. $l(e_1), \dots l(e_k)$, wobei wie üblich $e_i = (0, \dots, \underset{\uparrow}{1}_{i}, \dots 0)$.
Wir erhalten dann z.B.

$$l(e_1) = (1, 0, \dots, \underset{\uparrow}{0}_{k}, \underset{\uparrow}{c_k}_{k+1}, c_1 c_k, c_1^2 c_k + c_2 c_k, \dots),$$

weil

$$\begin{aligned} x_{k+1} &= c_1 x_k + \cdots + c_k x_1 \\ x_{k+2} &= c_1 x_{k+1} + \ldots c_k x_2 \\ x_{k+3} &= c_1 x_{k+2} + \ldots c_k x_3 \\ &\vdots \end{aligned}$$

Das ist jedenfalls keine "schöne" Beschreibung. Deshalb suchen wir eine schönere Beschreibung, die es dann gestattet, jede Lösung $l(a)$ in geschlossener Form anzugeben. Die Idee dabei ist, dass man eine lineare homogene Rekursion interpretiert als lineares Schieberegister:



Das lineare Schieberegister kann man beschreiben durch eine lineare Abbildung α, die den Übergang von der Registerbelegung zum Zeitpunkt $n-k$ zur Registerbelegung zum Zeitpunkt $n-k+1$ angibt:

$$\alpha \begin{pmatrix} x_{n-k} \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} x_{n-k+1} \\ \vdots \\ x_{n-1} \\ \underbrace{c_1 x_{n-1} + \cdots + c_k x_{n-k}}_{x_n} \end{pmatrix}, \text{ also}$$

$$\alpha \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_{1+k} \end{pmatrix}, \alpha^2 \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} x_3 \\ \vdots \\ x_{2+k} \end{pmatrix}, \ldots, \alpha^m \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} x_{m+1} \\ \vdots \\ x_{m+k} \end{pmatrix}.$$

Bezüglich der kanonischen Basis $\mathcal{B} = (e_1, \ldots, e_k)$ von K^k hat α die folgende Darstellungsmatrix:

$$A_\alpha^{\mathcal{B}} = A = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & 1 \\ c_k & c_{k-1} & \cdots & c_2 & c_1 \end{pmatrix}, \text{ also}$$

$$A \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_{k-1} \\ x_k \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_k \\ c_1 x_k + \cdots + c_k x_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_k \\ x_{k+1} \end{pmatrix}$$

Wir bestimmen nun die Eigenwerte von A mit Hilfe des **charakteristischen Polynoms**

$$\det(tE_k - A) = \det \begin{pmatrix} t & -1 & & & & \\ & t & -1 & & & \\ & & t & & & \\ & & & \ddots & t & -1 \\ -c_k & -c_{k-1} & -c_{k-2} & \cdots & -c_2 & t - c_1 \end{pmatrix}$$

$$= \; t^k - c_1 t^{k-1} - \cdots - c_{k-1} t - c_k.$$

Die letzte Gleichheit erhält man durch Determinantenberechnung mittels Entwicklung nach der 1. Zeile und per Induktion.

Man berechnet nun die Lösungen der sogenannten **charakteristischen Gleichung der Rekursion**:

$$t^k - c_1 t^{k-1} - c_2 t^{k-2} - \cdots - c_{k-1} t - c_k = 0.$$

Sei d eine Lösung der charakteristischen Gleichung der Rekursion, also ein Eigenwert von A.

Sei $v = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix}$ ein zugehöriger Eigenvektor, d.h. $v \neq 0$ und $A \cdot v = d \cdot v$.

Es gilt dann:

$$\begin{pmatrix} v_2 \\ v_3 \\ \vdots \\ v_k \\ c_1 v_k + \cdots + c_k v_1 \end{pmatrix} = A \cdot v = d \cdot v = \begin{pmatrix} dv_1 \\ dv_2 \\ \vdots \\ dv_{k-1} \\ dv_k \end{pmatrix}.$$

Also gilt $v_2 = dv_1, \ldots, v_k = d^{k-1}v_1$. Der Eigenvektor v ist durch v_1 bestimmt: v_1 legt sämtliche Einträge $v_2, \ldots, v_k$ fest. Der Eigenraum zu d ist 1-dimensional.

Wir wählen jetzt $v_1 = 1$. Dann ist ein Eigenvektor zu d: $\begin{pmatrix} 1 \\ d \\ d^2 \\ \vdots \\ d^{k-1} \end{pmatrix} = v$.

Wir bestimmen nun im Folgenden $l(v)$, wobei wir aus Gründen der Einfachheit statt eigentlich v^t, t wie transponiert, v schreiben.

Wir erhalten die "schöne" Form

$$l(v) = (1, d, d^2, d^3, \ldots, d^{k-1}, d^k, d^{k+1}, \ldots),$$

denn es gilt:

$$\begin{pmatrix} v_2 \\ \vdots \\ v_{k+1} \end{pmatrix} = A \cdot v = d \cdot v = \begin{pmatrix} d \\ \vdots \\ d^k \end{pmatrix}, \begin{pmatrix} v_{m+1} \\ \vdots \\ v_{k+m} \end{pmatrix} = A^m \cdot v = d^m \cdot v = \begin{pmatrix} d^m \\ \vdots \\ d^{m+k-1} \end{pmatrix}.$$

Wir nehmen nun an: A hat k verschiedene Eigenwerte $d_1, \ldots, d_k \in K$, d.h. die charakteristische Gleichung der Rekursion hat k verschiedene Nullstellen in K. Die zugehörigen k Eigenvektoren

$v_1 = \begin{pmatrix} 1 \\ d_1 \\ \vdots \\ d_1^{k-1} \end{pmatrix}, \ldots, v_k = \begin{pmatrix} 1 \\ d_k \\ \vdots \\ d_k^{k-1} \end{pmatrix}$ bilden eine Basis von K^k. Und damit bilden die

$l(v_i) = w_i = (1, d_i, d_i^2, \ldots)$ eine Basis von L.

Gegeben seien jetzt die k Anfangswerte $a_1, \ldots a_k$; wie bestimmt man nun a_n?

Oder auch anders: Gegeben ist $a = (a_1, \ldots, a_k)$; wie sieht die zugehörige Lösungsfolge $l(a)$ aus?

Wir wissen: $l(a)$ ist Linearkombination der $w_1, \ldots, w_k$, d.h.

$$\begin{aligned} l(a) &= (a_1, \ldots, a_k, a_{k+1}, \ldots) \\ &= s_1 \cdot w_1 + \cdots + s_k \cdot w_k \\ &= (s_1, s_1 d_1, s_1 d_1^2, \ldots) + \cdots + (s_k, s_k d_k, s_k d_k^2, \ldots) \end{aligned}$$

für geeignete $s_1, s_2, \ldots, s_k \in K$.

Es folgt:

$$\begin{aligned} s_1 + s_2 + \cdots + s_k &= a_1 \\ s_1 d_1 + \cdots + s_k s_k &= a_2 \\ &\vdots \\ s_1 d_1^{k-1} + \cdots s_k d_k^{k-1} &= a_k \end{aligned}$$

Das ist ein LGS für die $s_1, \ldots, s_k$ mit Koeffizienten d_i^j. Wir lösen dieses LGS und dann gilt

$$a_n = s_1 d_1^{n-1} + \cdots + s_k d_k^{n-1} \text{ für alle } n.$$

Das ist die gesuchte *geschlossene Form* für a_n. Damit erhalten wir folgende wichtige Aussage:

2.4 Satz

Gegeben ist die homogene lineare Rekursion (R_h) durch

$$x_n = c_1 x_{n-1} + \cdots + c_k x_{n-k} \text{ für alle } n > k, \text{ mit } c_k \neq 0.$$

Dann gelten folgende Aussagen:

a) Ist $d \in \mathbb{C}$ eine Lösung der charakteristischen Gleichung $t^k - c_1 t^{k-1} - c_2 t^{k-2} - \cdots - c_{k-1} t - c_k = 0$, so ist $w = (1, d, d^2, d^3, \ldots)$ eine Lösung von (R_h).

b) Besitzt die charakteristische Gleichung k verschiedene Lösungen $d_1, \ldots, d_k \in \mathbb{C}$, so bilden die $w_i = (1, d_i, d_i^2, d_i^3, \cdots), i = 1, \ldots k$, eine Basis des Lösungsraums von (R_h).

c) Unter den Voraussetzungen in b) sei $a = (a_1, \ldots, a_k) \in K^k$ ein beliebiger Vektor

von Anfangswerten und $s_1, \dots s_k$ sei die Lösung des LGS

$$\begin{aligned} s_1 + s_2 + \cdots + s_k &= a_1 \\ d_1 s_1 + \cdots + d_k s_k &= a_2 \\ &\vdots \\ d_1^{k-1} s_1 + \cdots d_k^{k-1} s_k &= a_k, \end{aligned}$$

so ist

$$l(a) = (s_1 + \cdots s_n, s_1 d_1 \cdots + s_k d_k, \dots, s_1 d_1^{n-1} + \cdots + s_k d_k^{n-1}, \dots),$$
$$\uparrow$$
$$n$$

d.h.

$$a_n = s_1 d_1^{n-1} + \cdots + s_k d_k^{n-1} \text{ für alle } n \in \mathbb{N}.$$

2.5 Beispiele

a) Wir betrachten hier noch einmal die Fibonacci-Folge: $F_n = F_{n-1} + F_{n-2}, n \geq 3$, mit $F_1 = 1, F_2 = 2$. Die zugehörige charakteristische Gleichung $t^2 - t - 1 = 0$ hat die Nullstellen $d_{1,2} = \dfrac{1 \pm \sqrt{5}}{2}$. Wir bestimmen dann die s_1, s_2 aus

$$\begin{aligned} s_1 + s_2 &= 1 \\ s_1(\frac{1+\sqrt{5}}{2}) + s_2(\frac{1-\sqrt{5}}{2}) &= 2 \end{aligned}$$

Als Lösung erhalten wir dann:

$$\begin{aligned} s_1 &= \frac{1}{\sqrt{5}} \left(\frac{3+\sqrt{5}}{2} \right) = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^2 \\ s_2 &= \frac{1}{\sqrt{5}} \left(\frac{-3+\sqrt{5}}{2} \right) = -\frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^2 \end{aligned}$$

Damit erhält man dann wegen $F_n = s_1 \cdot d_1^{n-1} + s_2 \cdot d_2^{n-1}$ das Ergebnis:

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^{n+1}.$$

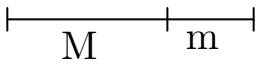
Bemerkung 1: Manchmal wird die Fibonacci-Folge auch mit den Anfangswerten $F_1 = F_2 = 1$ angegeben. Dann erhält man analog: $F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^n$.

Bemerkung 2: Wegen $-1 < \frac{1-\sqrt{5}}{2} < 0$ gilt $(\frac{1-\sqrt{5}}{2})^{n+1} \underset{n\to\infty}{\longrightarrow} 0$. Der zweite Term von F_n geht schnell gegen 0; F_n ist gleich der nächsten ganzen Zahl zu $\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} \approx \frac{1}{\sqrt{5}} \cdot 1,618^{n+1}$.

Bemerkung 3: $\frac{1+\sqrt{5}}{2}$ ist die Zahl des Goldenen Schnitts:

"Das Verhältnis des Gesamten zum Größeren ist gleich dem Verhältnis des Größeren zum Kleinen", in Formeln: $\frac{M+m}{M} = \frac{M}{m} = \frac{1+\sqrt{5}}{2}$.

Skizze:



b) Wir betrachten die lineare homogene Rekursion $x_n = 2x_{n-1} - x_{n-2} + 2x_{n-3}$, wobei $n \geq 4, x_1 = 0, x_2 = 4, x_3 = 10$.

Die zugehörige charakteristische Gleichung $t^3 - 2t^2 + t - 2 = 0$ hat eine Nullstelle $d_1 = 2$. Und wegen $t^3 - 2t^2 + t - 2 = (t-2)(t^2+1)$ erhalten wir dann als Nullstellen $d_1 = 2, d_2 = i, d_3 = -i$ (wobei $i^2 = -1$). Wir bestimmen dann s_1, s_2, s_3 mittels

$$s_1 + s_2 + s_3 = 0$$
$$2s_1 + is_2 - is_3 = 4$$
$$4s_1 - s_2 - s_3 = 10$$

Als Lösungen erhalten wir $s_1 = 2, s_2 = -1, s_3 = -1$, und damit erhalten wir

$$x_n = 2 \cdot 2^{n-1} - i^{n-1} - (-i)^{n-1} = \begin{cases} 2^n & \text{für } n \text{ gerade} \\ 2^n - 2 & \text{für } n \equiv 1 \mod 4 \\ 2^n + 2 & \text{für } n \equiv 3 \mod 4 \end{cases}$$

Wir geben jetzt noch an, wie sich die Aussage von 2.4b) verallgemeinern lässt, wenn die Nullstellen der charakteristischen Gleichung nicht notwendig verschieden sind:

2.6 Satz

Gegeben ist (R_h): $x_n = c_1 x_{n-1} + \cdots + c_k x_{n-k}, n > k, c_k \neq 0$.
Seien $d_1, \ldots d_s \in \mathbb{C}$ die verschiedenen Nullstellen der zugehörigen charakteristischen Gleichung $f(t) = 0$ mit Vielfachheiten $m_1, \ldots, m_s$, wobei $m_1 + \cdots + m_s = k$, d.h. für das charakteristische Polynom gilt: $f(t) = (t - d_1)^{m_1} \cdots (t - d_s)^{m_s}$.
Dann bilden

$$\begin{array}{c} w_{1,1}, \ldots, w_{1,m_1} \\ w_{2,1}, \ldots, w_{2,m_2} \\ \vdots \\ w_{s;1}, \ldots, w_{s,m_s} \end{array}$$

eine Basis des Lösungsraums von (R_h), wobei

$$w_{i,j} = (1, 2^{j-1} \cdot d_i, 3^{j-1} \cdot d_i^2, \ldots, \underset{\substack{\uparrow \\ n}}{n^{j-1} \cdot d_i^{n-1}}, \cdots)$$

für $i = 1, \ldots, s, j = 1, \ldots, m_i$.

Beweis: Siehe [A], Satz 3.1.

2.7 Beispiel

Gegeben sei $x_n = 3x_{n-2} - 2x_{n-3}, n \geq 4, x_1 = 1, x_2 = 2, x_3 = 6$.

Die zugehörige charakteristische Gleichung $t^3 - 3t + 2 = 0$ hat die Lösung $d_1 = 1$; die anderen erhalten wir aus $t^3 - 3t + 2 = (t-1)(t^2 + t - 2) = (t-1)^2(t+2)$: $d_1 = 1, m_1 = 2$, $d_2 = -2, m_2 = 1$. Wir erhalten dann:

$$w_{1,1} = (1, 1, 1, \dots, 1, \dots)$$

$$w_{1,2} = (1, 2, 3, \dots, n, \cdots)$$

$$w_{2,1} = (1, -2, 4, -8, \dots, \underset{\substack{\uparrow \\ n}}{(-2)^{n-1}}, \dots),$$

und es gilt $(x_n) = (x_1, x_2, \dots) = s_1 w_{1,1} + s_2 w_{1,2} + s_3 w_{2,1}$.

Wir bestimmen dann s_1, s_2, s_3 mittels

$$s_1 + s_2 + s_3 = 1$$

$$s_1 + 2s_2 - 2s_3 = 2$$

$$s_1 + 3s_2 + 4s_3 = 6.$$

Als Lösungen erhalten wir $s_1 = -\frac{4}{3}, s_2 = 2, s_3 = \frac{1}{3}$, und damit erhalten wir dann

$$x_n = -\frac{4}{3} + 2n + \frac{1}{3}(-2)^{n-1}.$$

Aus 2.6 folgt

2.8 Satz

Gegeben ist (R_h): $x_n = c_1 x_{n-1} + \cdots + c_k x_{n-k}, n > k, c_k \neq 0$.

Seien $d_1, \dots, d_s \in \mathbb{C}$ die verschiedenen Nullstellen der zugehörigen charakteristischen Gleichung $t^k - c_1 t^{k-1} - \cdots - c_k = 0$ mit Vielfachheiten $m_1, \dots, m_s$.

Sei o.B.d.A. $d := |d_1| = \cdots = |d_r| > |d_{r+1}| \geq \cdots \geq |d_s|$ und es sei $m = \max\{m_1, \ldots, m_s\}$.

Ist $(a_1, a_2, a_3, \ldots)$ eine Lösung von (R_h), so gilt $|a_n| = \mathcal{O}(n^{m-1} \cdot d^n)$.

(D.h. es existiert eine Konstante $C > 0$, so dass $|a_n| \leq C \cdot n^{m-1} \cdot d^n$ für alle n.)

Insbesondere gilt:

Ist $d < 1$, so ist $\lim\limits_{n\to\infty} a_n = 0$.

Ist $d = 1$, so wächst $|a_n|$ höchstens polynomiell.

Ist $d > 1$, so wächst $|a_n|$ höchstens exponentiell.

(Das genaue Wachstum hängt von den Anfangswerten ab.)

Wir betrachten im Folgenden inhomogene lineare Rekursionen von Typ:

$$(R) \quad x_n = c_1 x_{n-1} + \cdots + c_k x_{n-k} + a \cdot r^n$$

für alle $n > k$, $c_k \neq 0$, a, r fest, $a, r \neq 0$. (Ein wichtiger Spezialfall ist $r = 1$.)

Wir machen einen Ansatz für eine spezielle Lösung: $(d{\cdot}r, d{\cdot}r^2, d{\cdot}r^3, \ldots)$ für $d \neq 0$ geeignet:

Diese unendliche Folge ist eine Lösung genau dann, wenn:

$$\begin{aligned}
& d \cdot r^n = c_1 \cdot d \cdot r^{n-1} + \ldots c_k \cdot d \cdot r^{n-k} + ar^n \text{ für alle } n > k \\
\iff & \\
& \frac{(d-a)}{d} \cdot r^n = c_1 \cdot r^{n-1} + \ldots c_k \cdot r^{n-k} \text{ für alle } n > k \\
\iff & \\
& \frac{(d-a)}{d} \cdot r^k = c_1 \cdot r^{k-1} + \ldots c_{k-1} \cdot r + c_k \text{ für alle } n > k \\
\iff & \\
& \frac{a}{d} \cdot r^k = \underbrace{r^k - c_1 \cdot r^{k-1} - \cdots - c_{k-1} \cdot r - c_k}_{=:b} \text{für alle } n > k
\end{aligned}$$

Ist $b \neq 0$, so setze $d := \frac{a}{b} \cdot r^k$. Damit erhalten wir

2.9 Satz

Gegeben sei die lineare inhomogene Rekursion (R) wie oben. Wir setzen voraus, dass r keine Nullstelle der zugehörigen charakteristischen Gleichung der zu (R) gehörenden linearen homogenen Rekursion (R_h) ist, d.h. es gilt $b := r^k - c_1 r^{k-1} - \cdots - c_{k-1} r - c_k \neq 0$. Wir setzen $d := \frac{a}{b} \cdot r^k$.
Dann ist $(d \cdot r, d \cdot r^2, d \cdot r^3, \dots)$ eine spezielle Lösung von (R).

2.10 Beispiel

Wir betrachten noch einmal die Türme von Hanoi (vgl. 2.2b)): Die zugehörige Rekursion lautet: $a_n = 2 \cdot a_{n-1} + 1$ für alle $n \geq 2$, wobei $a_1 = 1$. Diese Rekursion ist vom Typ 2.9 mit $a = r = 1$. Die zugehörige charakteristische Gleichung ist $t - 2 = 0$ mit der einzigen Nullstelle $t_1 = 2$. Die Lösung der zugehörigen linearen homogenen Rekursion ist $(s, 2s, 2^2 s, \dots)$: $s \in K\}$. $r = 1$ ist keine Nullstelle der charakteristischen Gleichung; es ist $b := 1 - 2 = -1$ und $d = -1$.
Nach 2.9 ist $(-1, -1, \dots)$ eine spezielle Lösung von (R). Und somit erhalten wir nach 2.3b) die allgemeine Lösung $(s - 1, 2s - 1, 2s^2 - 1, \dots)$. Wir bestimmen nun s so, dass $s - 1 = a_1 = 1$, also ist $s = 2$. Dann erhalten wir schlussendlich die geschlossene Form $(a_1, a_2, \dots) = (1, 3, 7, \dots, \underset{\substack{\uparrow \\ n}}{2^n - 1}, \dots)$ d.h. $a_n = 2^n - 1$.

Falls r eine Nullstelle der charakteristischen Gleichung der zugehörigen linearen homogenen Rekursion ist, so kann man auch eine spezielle Lösung angeben. Dazu vergleiche man das Skript [H], 3.10.
Der einfachste Fall ist der, dass r eine einfache Nullstelle der charakteristischen Gleichung ist. Dann setzt man $b := (k+1)r^k - k \cdot c_1 r^{k-1} - \cdots - 2c_{k-1} r - c_k$. Dann ist $b \neq 0$ und mit $d := \frac{a}{b} \cdot r^k$ ist $(dr, 2dr^2, 3dr^3, \dots, ndr^n, \dots)$ eine spezielle Lösung von (R). Dazu schauen

wir uns noch ein Beispiel an:

2.11 Beispiel

Gegeben ist die lineare inhomogene Rekursion $a_n = 4a_{n-2} + 3 \cdot 2^n$, $n \geq 3$, $a_1 = 3, a_2 = 8$. Die charakteristische Gleichung ist $t^2 - 4 = (t+2)(t-2) = 0$. Der Lösungsraum der zugehörigen linearen homogenen Rekursion ist $L = \langle (1, 2, 2^2, \dots), (1, -2, (-2)^2, \dots) \rangle$, $r = 2$ ist einfache Nullstelle von $t^2 - 4$. Wir setzen $b := 3 \cdot 2^2 - 4 = 8$, $d := \frac{a}{b} r^k = \frac{3}{8} \cdot 2^2 = \frac{3}{2}$. Dann ist eine spezielle Lösung gegeben durch $(3, 2 \cdot 3 \cdot 2, 3 \cdot 3 \cdot 2^2, 4 \cdot 3 \cdot 2^3, \dots, n \cdot 3 \cdot \overset{\uparrow}{2}^{n-1}, \dots)$. n Als allgemeine Lösung erhalten wir dann: $n \cdot 3 \cdot 2^{n-1} + s_1 \cdot s^{n-1} + s_2 \cdot (-2)^{n-1}$ mit $s_1, s_2 \in K$. s_1, s_2 erhalten wir als Lösung des LGS:

$$3 = a_1 = 3 + s_1 + s_2$$
$$8 = a_2 = 12 + 2s_1 - 2s_2.$$

Wir erhalten $s_1 = -1, s_2 = 1$ und damit gilt für a_n:

$$a_n = \begin{cases} 3n \cdot 2^{n-1} & \text{für } n \text{ ungerade} \\ 3n \cdot 2^{n-1} - 2 \cdot 2^{n-1} = (3n-2) \cdot 2^{n-1} & \text{für } n \text{ gerade.} \end{cases}$$

Ebenso kann man die Lösungen für lineare inhomogene Rekursionen vom Typ

$$x_n = c_1 \cdot x_{n-1} + \cdots + c_k \cdot x_{n-k} + a \cdot n^l$$

für alle $n > k$, $c_k \neq 0$, $l \in \mathbb{N}_0$, $a \neq 0$, bestimmen (vgl. Skript, [H], 3.12).

Wir wollen nun das Wachstum von Rekursionsfolgen bestimmen, die bei Komplexitätsabschätzungen von Divide-and-Conquer-Algorithmen auftreten:

$$x_n = a \cdot x_{n-1} + g(n)$$
$$\text{oder } x_n = a \cdot x_{\frac{n}{b}} + g(n) \quad (\text{dabei: } \frac{n}{b} \text{ steht für } \left\lceil \frac{n}{b} \right\rceil \text{ oder } \left\lfloor \frac{n}{b} \right\rfloor).$$

(Man beachte, dass die letztgenannte Rekursion keine lineare Rekursion fester Ordnung ist.)

Für den ersten Typ benötigen wir den folgenden Satz:

2.12 Satz

Gegeben sei die lineare Rekursion $x_n = f(n-1) \cdot x_{n-1} + g(n)$ für alle $n \geq 2$, wobei $f, g \in K^{\mathbb{N}}$. Gegeben sei der Anfangswert $a_1 =: g(1)$.

Dann gilt für die zugehörige Lösung $(a_1, a_2, \dots)$:

$$a_n = \sum_{i=1}^{n} \left(g(i) \prod_{j=i}^{n-1} f(j) \right), \text{ wobei } \prod_{j=n}^{n-1} f(j) := 1.$$

Beweis: Vollständige Induktion über n:

IA: Für $n = 1$ gilt: $g(1) \cdot \prod_{j=1}^{0} f(j) = g(1) = a_1$.

IS: $n \longrightarrow n+1$: Es gilt:

$$\begin{aligned} a_{n+1} &= f(n) \cdot a_n + g(n+1) \\ &= f(n) \left(\sum_{i=1}^{n} \left(g(i) \prod_{j=i}^{n-1} f(j) \right) \right) + g(n+1) \\ &= \sum_{i=1}^{n} \left(g(i) \prod_{j=i}^{n} f(j) \right) + g(n+1) \\ &= \sum_{i=1}^{n+1} \left(g(i) \prod_{j=i}^{n} f(j) \right) \end{aligned}$$

Wir behandeln ein Beispiel zu diesem Satz:

2.13 Beispiel

Gegeben sei die lineare inhomogene Rekursion $x_n = 2^{n-1} \cdot x_{n-1} + 2^{\frac{n(n-1)}{2}}$ mit $a_1 = 1$. Wir untersuchen also jetzt eine lineare inhomogene Rekursion der Ordnung 1, die keine konstanten Koeffizienten besitzt: Es ist $f(n) = 2^n, g(n) = 2^{\frac{n(n-1)}{2}}$ für alle $n \in \mathbb{N}$ (beachte: $g(1) = 1 = a_1$).
Wir erhalten dann mit 2.12:

$$\begin{aligned} a_n &= \sum_{i=1}^{n} 2^{\frac{i(i-1)}{2}} \prod_{j=i}^{n-1} 2^j = \sum_{i=1}^{n} 2^{\frac{i(i-1)}{2}} \frac{\sum_{j=1}^{n-1} 2^j}{\sum_{j=1}^{i-1} 2^j} \\ &= \sum_{i=1}^{n} 2^{\frac{i(i-1)}{2}} \frac{2^{\sum_{j=1}^{n-1} j}}{2^{\sum_{j=n}^{i-1} j}} = \sum_{i=1}^{n} 2^{\frac{i(i-1)}{2}} \frac{2^{\frac{n(n-1)}{2}}}{2^{\frac{i(i-1)}{2}}} \\ &= n \cdot 2^{\frac{n(n-1)}{2}}. \end{aligned}$$

Im Folgenden führen wir einige wichtige Bezeichnungen bezüglich des Wachstums ein:

2.14 Definition

(Dazu vergleiche man auch 2.8.)
Gegeben ist eine Folge $(a_n)_{n\in\mathbb{N}}$ mit $a_n \geq 0$ für alle n. Wir setzen dann:

$\mathcal{O}(a_n) := \{(b_n)$: es exstieren $C > 0, n_1 \in \mathbb{N}$ mit $0 \leq b_n \leq C \cdot a_n$ für alle $n \geq n_1\}$

$\Omega(a_n) := \{(b_n)$: es existieren $D > 0, n_1 \in \mathbb{N}$ mit $b_n \geq D \cdot a_n$ für alle $n \geq n_1\}$

$\Theta(a_n) := \mathcal{O}(a_n) \cap \Omega(a_n)$

$= \{(b_n)$: es existieren $C_1, C_2 > 0, n_1 \in \mathbb{N}$ mit $C_1 \cdot a_n \leq b_n \leq C_2 \cdot a_n$ für alle $n \geq n_1\}$.

Beispiele:

$(\log(n)) \in \mathcal{O}(n^a)$ für alle $a > 0$,

$(n^a) \in \mathcal{O}(b^n)$ für alle $a > 0, b > 1$,

$(n^2 + 5n + 1) \in \Theta(n^2)$ wegen $n^2 \leq n^2 + 5n + 1 \leq 7n^2$.

2.15 Satz

Gegeben sei eine lineare Rekursion

$$(R) \quad x_n = a \cdot x_{n-1} + g(n) \text{ für alle } n \geq 2,$$

wobei $a > 1$. Der Anfangswert sei gegeben durch $a_1 =: g(1)$. Dabei ist der homogene Fall gegeben durch $g(n) = 0$ für alle $n \geq 2$.
Sei nun $(a_1, a_2, \dots)$ Lösung von (R) mit dem Anfangswert a_1. Dann gelten folgende Aussagen:

a) Ist $(|g(n)|) \in \mathcal{O}(a^{n(1-\epsilon)})$ für ein $\epsilon > 0$, so gilt $(|a_n|) \in \mathcal{O}(a^n)$.

   Ist überdies $g(n) \geq 0$ für alle n, $g \neq 0$, d.h. ist g nicht die Nullfunktion, so ist $a_n \geq 0$ für alle n und $(a_n) \in \Theta(a^n)$.

b) Ist $(|g(n)|) \in \Theta(a^n)$, so ist $(|a_n|) \in \mathcal{O}(n \cdot a^n)$.

   Existiert überdies ein $m_1 \in \mathbb{N}$ mit $g(n) \geq 0$ für alle $n \geq m_1$, so ist $a_n \geq 0$ für alle $n \geq m_2$ für ein $m_2 \in \mathbb{N}$ und $(a_n) \in \Theta(n \cdot a^n)$.

   Allgemeiner gilt:

   Ist $(|g(n)|) \in \Theta(n^k (\log n)^l \cdot a^n)$, $k, l \in \mathbb{N}_0$, so gelten $(|a_n|) \in \mathcal{O}(n^{k+1} (\log n)^l \cdot a^n)$ und $(a_n) \in \Theta(n^{k+1} (\log n)^l \cdot a^n)$, falls $g(n) \geq 0$ für alle $n \geq m_1$.

c) Existiert $c \in \mathbb{R}$ mit $0 < c < 1$ und $n_1 \geq 2$ mit $0 < a \cdot g(n-1) \leq c \cdot g(n)$ für alle $n \geq n_1$, so ist $(a_n) \in \Theta(g(n))$.

Bemerkung:

Im Fall c) gilt (zur Vereinfachung sei $n_1 = 2$):
$g(n) \geq \left(\frac{a}{c}\right) g(n-1) \geq \left(\frac{a}{c}\right)^{n-1} g(1) = \left(\frac{a}{c}\right)^{n} \cdot \frac{c}{a} \cdot g(1) = a^{n(1+\epsilon)} \frac{c}{a} \cdot g(1)$ mit $\epsilon := -\log_a c > 0$.
Also ist $(g(n)) \in \Omega(a^{n(1+\epsilon)})$ für ein $\epsilon > 0$. Dazu vergleiche man den Fall a).

Beweis:

Nach 2.12 ist $a_n = \sum_{i=1}^{n} g(i) a^{n-i}$.
a) Sei o.B.d.A. $|g(n)| \leq c \cdot a^{n(1-\epsilon)}$ für alle n. Dann ist:

$$\begin{aligned} |a_n| &\leq c \cdot \sum_{i=1}^{n} a^{i(1-\epsilon)a^{n-i}} \leq c \cdot a^n \sum_{i=0}^{n} a^{-i\epsilon} \\ &= c \cdot a^n \frac{1 - a^{-(n+1)\epsilon}}{1 - a^{-\epsilon}} \\ &\underset{\substack{\epsilon>0 \\ a>1}}{\leq} \left(c \cdot \frac{1}{1 - a^{-\epsilon}} \right) \cdot a^n. \end{aligned}$$

Wir erhalten also $|a_n| \in \mathcal{O}(a^n)$.
Ist $g(n) \geq 0$ für alle n und $g(n_1) > 0$, so ist $a_n = \sum_{i=1}^{n} g(i) a^{n-i} \geq g(n_1) a^{n-n_1} = \underbrace{\frac{g(n_1)}{a^{n_1}}}_{konst>0} a^n$
für alle $n \geq n_1$, also $(a_n) \in \Theta(a^n)$.

Zu b) und c) siehe das Skript [H].

Zu b) betrachte man das Beispiel: $x_n = a x_{n-1} + a^n, a_1 = a$. Dann ergibt sich mittels 2.12
$a_n = \sum_{i=1}^{n} a^i a^{n-i} = n \cdot a^n$. Oder auch anschaulich:
$a_1 = a, a_2 = a^2 + a^2 = 2a^2, a_3 = a \cdot a_2 + a^3 = 3a^3, \ldots$

2.16 Bemerkung

Wenn bei einem Divide-and-Conquer-Algorithmus ein Problem der Größe n in a Teilprobleme der Größe $n-1$ aufgeteilt wird, die rekursiv gelöst werden, so ist die Komplexität des Algorithmus gegeben durch $T(n) = a \cdot T(n-1) + g(n)$. Dabei beschreibt $g(n)$ den

Aufwand für das Zerlegen in Teilprobleme und das Zusammensetzen der Lösungen. Die Komplexität wird nach 2.15 durch die Teilprobleme bestimmt, falls $g(n)$ nicht zu stark wächst, d.h. $(g(n)) \in \mathcal{O}(a^{n(1-\epsilon)})$.
Die Komplexität wird von $g(n)$ dominiert, falls (i.W.) $(g(n)) \in \Omega(a^{n(1+\epsilon)})$.

2.17 Satz

Es seien $a, b \in \mathbb{R}$, wobei $a, b > 1$. Im Folgenden steht $\frac{n}{b}$ abkürzend für $\lfloor \frac{n}{b} \rfloor$ oder $\lceil \frac{n}{b} \rceil$.
Gegeben sei die Rekursion

$$(R) \quad x_n = a \cdot x\left(\frac{n}{b}\right) + g(n) \text{ für alle } n \geq b.$$

Dabei setzen wir voraus, dass $g(n)$ monoton wachsend, nicht-negativ und nicht die Nullfunktion ist. Die Anfangswerte sind $a_i =: g(i), i = 1, \ldots, \lceil b \rceil - 1$ und $(a_1, a_2, \ldots)$ sei die zugehörige Lösung von (R). Dann gelten folgende Aussagen:

a) Ist $(g(n)) \in \mathcal{O}(n^{\log_b(a)-\epsilon})$ für ein $\epsilon > 0$, so ist $(a_n) \in \Theta(n^{\log_b(a)})$.

b) Ist $(g(n)) \in \Theta(n^{\log_b(a)})$, so ist $(a_n) \in \Theta(\log n \cdot n^{\log_b(a)})$.

c) Existiert $0 < c < 1$ mit $a \cdot g\left(\lceil \frac{n}{b} \rceil\right) \leq c \cdot g(n)$ für alle $n \geq n_0$, so ist $(a_n) \in \Theta(g(n))$.

Bemerkung: Ähnlich wie in 2.15c) kann man zeigen, dass in c) $(g(n)) \in \Omega(n^{\log_b(a)+\epsilon})$ gilt, wobei $\epsilon = -\log_b(c) > 0$.
Beweisidee: Angenommen $b \in \mathbb{N}$.
Ist $n = b^m$, $m \in \mathbb{N}_0$, so erhält man mit den Substitutionen $x'_m := x(n), g'(m) := g(n)$ die Rekursion

$$(R) \quad x'_m = a \cdot x'_{m-1} + g'(m) \text{ vom Typ 2.15.}$$

Dabei beachte man $a^m = a^{\log_b(n)} = a^{log_a(n) \cdot \log_b(a)} = n^{\log_b(a)}$.

Wenn man nur die natürliche Zahlen n vom Typ $n = b^m$ betrachtet, erhält man 2.17 aus 2.15.

Für beliebiges n benötigt man, dass $g(n)$ monoton wächst.

Spezialfälle:

Der wichtigste Fall im Satz 2.17 ist $a = b > 1$:

- Ist $(g(n)) \in \mathcal{O}(n^{1-\epsilon})$ für ein $\epsilon > 0$, so ist $(a_n) \in \Theta(n)$.

- Ist $(g(n)) \in \mathcal{O}(n)$, so ist $(a_n) \in \Theta(n \cdot \log(n))$.

2.18 Beispiel

Wir betrachten das Sortierverfahren **Mergesort**:

Diese Verfahren arbeitet nach dem Prinzip "Divide and Conquer". Die zu sortierende Datei wird als Liste L mit n Elementen betrachtet. Man zerlegt diese Liste in zwei gleich große Teile L', L'' und wendet dann Mergesort rekursiv auf L' und L'' an. Geliefert werden dann sortierte Listen $\widetilde{L}'$ und $\widetilde{L}''$. Man vergleicht die beiden sortierten Listen und bildet dann durch "Mischen" die gesamte sortierte Liste. Die maximale Anzahl an Vergleichen ergibt sich zu

$$V(n) = 2 \cdot V\left(\frac{n}{2}\right) + (n-1) \text{ mit } V(2) = 1.$$

Nach 2.17 b) ergibt sich $(V(n)) \in \Theta(n \cdot \log(n))$.

3 Geordnete Mengen

3.1 Definition

a) M sei eine Menge und $\preceq$ sei eine Relation auf M. $\preceq$ heißt **Ordnungsrelation** (oder auch **partielle Ordnung**), falls gilt:

(1) $a \preceq a$ für alle $a \in M$ (Reflexivität)

(2) Falls $a \preceq b$ und $b \preceq a$, so ist $a = b$ (Antisymmetrie)

(3) Falls $a \preceq b$ und $b \preceq c$, so gilt auch $a \preceq c$ (Transitivität)

Das Paar $(M, \preceq)$ heißt (partiell) **geordnete Menge**.

Wir führen noch die folgende Schreibweise ein: $a \prec b$, falls $a \preceq b$ und $a \neq b$.

b) Sei $(M, \preceq)$ eine geordnete Menge, $a, b \in M$ und $a \preceq b$.

Dann heißt $[a, b] := \{c \in M : a \preceq c \text{ und } c \preceq b\}$ **Intervall**.

c) Sei $(M, \preceq)$ geordnete Menge, $a, b \in M$.

a heißt **Vorgänger** von b, falls $a \prec b$ und falls kein $c \in M$ existiert mit $a \prec c \prec b$.

Wir benutzen für diesen Sachverhalt die Schreibweise: $a \prec\!\cdot\, b$.

d) $(M, \preceq)$ heißt **total geordnete Menge** oder **Kette**, falls $(M, \preceq)$ geordnet und folgende zusätzliche Bedingung erfüllt ist:

(4) Für alle $a, b \in M$ ist $a \preceq b$ oder $b \preceq a$.

3.2 Beispiele

a) $(\mathbb{R}, \leq)$ ist total geordnet; kein Element besitzt einen Vorgänger.

$(\mathbb{Z}, \leq)$ ist total geordnet; jedes Element a besitzt Vorgänger $a - 1$.

b) $(\mathbb{N}, |)$ ist geordnet, aber nicht total geordnet.

c) A sei eine Menge, $(\mathcal{P}(A), \subseteq)$ ist geordnet, aber nicht total geordnet, falls $|A| \geq 2$.

d) Sei $M := \{0,1\}^n$, also die Menge der $0-1$-Folgen der Länge n. Wir definieren die sog. **lexikographische Ordnung** auf M:

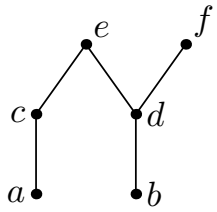
$$\begin{aligned}&(a_1,\ldots,a_n) \leq (b_1,\ldots,b_n)\\ :\Leftrightarrow &(a_1,\ldots,a_n) = (b_1,\ldots,b_n) \text{ oder}\\ &\text{es existiert } k \in \mathbb{N}_0, 0 \leq k < n, \text{ mit } a_1 = b_1,\ldots,a_k = b_k, a_{k+1} = 0, b_{k+1} = 1.\end{aligned}$$

Diese Ordnung ist eine totale Ordnung.

Eine endliche geordnete Menge $(M, \preceq)$ kann man durch ein **Hasse-Diagramm** veranschaulichen (Helmut Hasse, 1898-1979, Zahlentheoretiker). Wir repräsentieren die Ordnung durch einen **gerichteten Graphen**: Dabei ist jede Ecke ein Element von M; wir haben eine **gerichtete Kante zwischen** a **und** b genau dann wenn gilt $a \prec\cdot\, b$.. Dabei lässt man den Richtungspfeil weg; die Richtung wird dadurch zum Ausdruck gebracht, dass sich im Hasse-Diagramm b oberhalb von a befindet. Die Richtung ist also "von unten nach oben".

3.3 Beispiele

a) Sei $M = \{a,b,c,d,e,f\}$; vorgegeben ist folgendes Hasse-Diagramm:



$a \preceq e, a \not\preceq d$

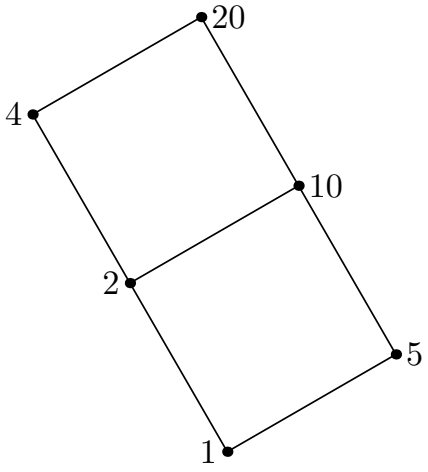
$[a,e] = \{a,c,e\}$

Vorgänger von e sind c und d.

b) $M = \{1,2,3,4,5\}$ mit der üblichen $\leq$-Relation. Das Hasse-Diagramm sieht dann folgendermaßen aus:

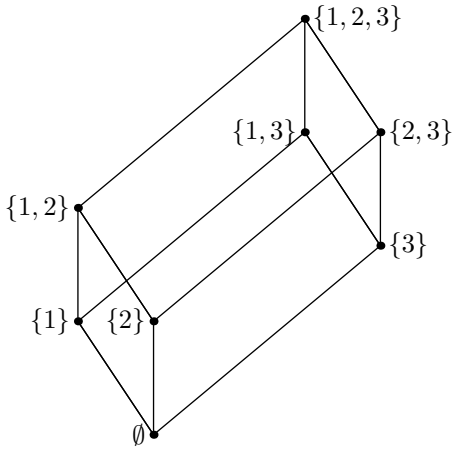
c) $M = \{1, 2, 4, 5, 10, 20\} =: \mathbb{T}_{20}$ mit der Teilerrelation, also die Menge aller Teiler von 20.

Das Hasse-Diagramm hat dann folgendes Aussehen:



d) $A = \{1, 2, 3\}$, $M := \mathcal{P}(A)$, geordnet bezüglich $\subseteq$.

Als Hasse-Diagramm erhalten wir dann:



3.4 Definition

Sei $(M, \preceq)$ eine geordnete Menge. Dann bezeichnet man die Menge

$$A(M, \preceq) := \{f : M \times M \to \mathbb{C} : f(a, b) = 0 \text{ für alle } a, b \in M \text{ mit } a \not\preceq b\}$$

als **Inzidenzalgebra** zu $(M, \preceq)$.

Die Inzidenzalgebra einer Ordnung wurde 1964 von Gian-Carlo Rota zur Untersuchung kombinatorischer Sachverhalte eingeführt.

Ist M eine endliche Menge, so kann man jedes Element $f \in A(M, \preceq)$ als Matrix auffassen:

$$f \longleftrightarrow \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{array} \begin{array}{c} \begin{array}{cccccc} a_1 & a_2 & \dots & a_j & \dots & a_n \end{array} \\ \begin{pmatrix} & & & \vdots & & \\ & & & \vdots & & \\ & & & \vdots & & \\ \dots & \dots & \dots & f(a_i, a_j) & & \\ & & & & & \\ & & & & & \end{pmatrix} \end{array}$$

3.5 Bemerkungen

$A(M, \preceq)$ ist abgeschlossen unter Addition und Multiplikation von Funktionen, die folgendermaßen definiert sind:

Für alle $f, g \in A(M, \preceq)$ definieren wir:

$$(f + g)(a, b) := f(a, b) + g(a, b)$$
$$(f \cdot g)(a, b) := \sum_{c \in M} f(a, c) \cdot g(c, b) \text{ (Faltungsprodukt).}$$

Ist $a \preceq b$, so gilt

$$(f \cdot g)(a, b) = \sum_{c \in [a,b]} f(a, c) \cdot g(c, b).$$

Zum Nachweis der Abgeschlossenheit von $A(M, \preceq)$ bezüglich Multiplikation ist folgendes zu bemerken:

Seien $f, g \in A(M, \preceq)$ und $a, b \in M$ mit $a \not\preceq b$. Ist $c \in M$, so gilt dann $a \not\preceq c$ oder $c \not\preceq b$ wegen der Transitivität von $\preceq$, d.h. $f(a, c) \cdot g(c, b) = 0$. Also sind alle Summenden gleich Null, d.h. $(f \cdot g)(a, b) = 0$. Daher gilt $f \cdot g \in A(M, \preceq)$.

$A(M, \preceq)$ besitzt auch ein **neutrales Element** $\delta \in A(M, \preceq)$ bezüglich der Multiplikation, wobei

$$\delta(a, b) := \begin{cases} 1 & \text{für } a = b \\ 0 & \text{für } a \neq b \end{cases}$$

δ entspricht für endliches M der **Einheitsmatrix**.

Dem Leser sei als Übung überlassen nachzuweisen, dass δ neutrales Element ist.

3.6 Zeta-Funktion

Die **Zeta-Funktion** $\zeta \in A(M, \preceq)$ wurde von Rota eingeführt und ist definiert durch:

$$\zeta(a,b) := \begin{cases} 1 & \text{für } a \preceq b \\ 0 & \text{sonst.} \end{cases}$$

Diese Zeta-Funktion beschreibt die Ordnung $\preceq$ auf M.

Beispiel: Wir berechnen die Zeta-Funktion von Beispiel 3.3c) und bestimmen dazu die zugehörige Matrix:

$$\zeta \longleftrightarrow \begin{array}{c} \\ 1 \\ 2 \\ 4 \\ 5 \\ 10 \\ 20 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 4 & 5 & 10 & 20 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

3.7 Satz und Definition

Sei $(M, \preceq)$ endliche geordnete Menge.

a) ζ ist bezüglich der Multiplikation invertierbar, d.h. es existiert $\mu \in A(M, \preceq)$ mit $\mu \cdot \zeta = \zeta \cdot \mu = \delta$.

   μ nennt man die **Möbius-Funktion** zu $(M, \preceq)$ (benannt nach August Ferdinand Möbius, 1790-1868, Mathematiker und Astronom, der diese Funktion in einem zahlentheoretischen Spezialfall (siehe 3.8 b)) betrachtet hat).

D.h. für alle $a, b \in M$ mit $a \preceq b$ gilt:

$$\sum_{c \in [a,b]} \mu(a,c) = \sum_{c \in [a,b]} \mu(c,b) = \begin{cases} 1 & \text{für } a = b \\ 0 & \text{für } a \neq b. \end{cases}$$

b) μ ist rekursiv definiert durch:

$$\mu(a,a) = 1 \text{ für alle } a \in M$$
$$\mu(a,b) = 0, \text{ falls } a \not\preceq b$$
$$\mu(a,b) = -\sum_{\substack{c \in [a,b] \\ c \neq b}} \mu(a,c), \text{ falls } a \prec b.$$

$\mu(a,b)$ hängt nur von der Relation $\preceq$ auf $[a,b]$ ab, nicht von $\preceq$ auf ganz M.

c) Es ist $\mu(a,b) \in \mathbb{Z}$ für alle $a, b \in M$ und es gilt $\mu(a,b) = -1$, falls $a \prec\cdot\, b$.

Beweis: a) Die Gleichung in a) ist äquivalent zu $\mu = \zeta^{-1}$, denn für $a \preceq b$ ist $\zeta(a,c) = \zeta(c,b) = 1$ für alle $c \in [a,b]$.

b) Folgt aus a).

c) Folgt aus b).

3.8 Beispiele

a) $(M, \preceq)$ sei total geordnet und endlich.

Also $M = \{a_1, a_2, \ldots, a_n\}$ mit $a_1 \prec a_2 \prec \cdots \prec a_n$.

Es gilt für $a, b \in M$:

$$\mu(a,b) = \begin{cases} 1 & \text{für } a = b \\ -1 & \text{für } a \prec\cdot\, b \\ 0 & \text{sonst.} \end{cases}$$

Beweis:

Wegen 3.7 b) und 3.7 c) bleibt zu zeigen: $\mu(a,b) = 0$, falls $a \neq b$, a kein Vorgänger von b.

Da M total geordnet ist, gilt $a \prec b$ oder $b \prec a$.

Falls $b \prec a$, so ist $a \not\leq b$, also $\mu(a,b) = 0$.

Sei also $a \prec b$; dann existieren $b_1, \ldots, b_k \in M$ mit $a = b_0 \prec\cdot\, b_1 \prec\cdot \cdots \prec\cdot\, b_k = b$, wobei $k \geq 2$. Wir erhalten dann:

$$\begin{aligned}\mu(a,b) &\underset{\substack{3.7b)\\ M \text{ total geordnet}}}{=} -\sum_{i=0}^{k-1} \mu(a,b_i)\\ &= -\mu(a,b_{k-1}) - \sum_{i=0}^{k-2} \mu(a,b_i)\\ &\underset{\substack{3.7b)\\ k\geq 2}}{=} -\mu(a,b_{k-1}) + \mu(a,b_{k-1}) = 0.\end{aligned}$$

Im Folgenden werden wir noch die Matrixdarstellungen der Funktionen ζ und μ angeben:

$$\zeta \longleftrightarrow \begin{array}{c} \\ a_1 \\ \vdots \\ \\ \vdots \\ a_n \end{array} \begin{array}{c} \begin{array}{ccccc} a_1 & \ldots & \ldots & \ldots & a_n \end{array} \\ \begin{pmatrix} 1 & \ldots & \ldots & 1 & 1 \\ & 1 & \ddots & \ldots & 1 \\ & & 1 & \ddots & 1 \\ & \mathbf{0} & & 1 & 1 \\ & & & & 1 \end{pmatrix} \end{array} \quad \text{bzw. } \mu \longleftrightarrow \begin{array}{c} \\ a_1 \\ \vdots \\ \\ \vdots \\ a_n \end{array} \begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & \ldots & \ldots & a_n \end{array} \\ \begin{pmatrix} 1 & -1 & 0 & \ldots & 0 \\ & 1 & -1 & \ddots & 0 \\ & & 1 & \ddots & 0 \\ & \mathbf{0} & & 1 & -1 \\ & & & & 1 \end{pmatrix} \end{array}$$

Die ζ entsprechende Matrix ist eine obere Dreiecksmatrix mit lauter Einsen. Es ist zu bemerken, dass man natürlich die μ entsprechende Matrix auch durch Matrixinversion erhalten kann.

b) Wir betrachten nochmals $(\mathbb{T}_{20}, |)$ (vgl. das Beispiel in 3.6) und bestimmen im Folgenden die zu ζ und μ zugehörigen Matrizen: Wir erhalten zunächst für ζ:

$$\zeta \longleftrightarrow \begin{array}{c} \\ 1 \\ 2 \\ 4 \\ 5 \\ 10 \\ 20 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 4 & 5 & 10 & 20 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Im Folgenden berechnen wir nun die μ zugehörige Matrix:

Zunächst gilt: $\mu(a,a) = 1$ für alle $a \in M$, und es gilt $\mu(a,b) = 0$, falls $a \nmid b$.

Weiter gilt:

$$\mu(1,2) = \mu(1,5) = \mu(2,4) = \mu(2,10) = \mu(5,10) = \mu(4,20) = \mu(10,20) = -1$$

(vgl. auch das Hasse-Diagramm 3.3c).

$$\mu(1,4) = \mu(5,20) = 0, \text{ denn } [1,4] \text{ und } [5,20] \text{ sind Ketten.}$$

$$\mu(1,10) = -\mu(1,1) - \mu(1,2) - \mu(1,5) = -1 + 1 + 1 = 1 \text{ wegen 3.7b)}$$

$$\mu(1,20) = \underbrace{-\mu(1,1) - \mu(1,2) - \mu(1,5) - \mu(1,10)}_{=0} - \mu(1,4) = -\mu(1,4) = 0, \text{ wieder wegen 3.7b)}$$

$$\mu(2,20) = -\mu(2,2) - \mu(2,4) - \mu(2,10) = -1 + 1 + 1 = 1, \text{ auch wieder wegen 3.7b)}$$

Wir erhalten also für die μ zugeordnete Matrix die Darstellung:

$$\mu \longleftrightarrow \begin{array}{c} \\ 1 \\ 2 \\ 4 \\ 5 \\ 10 \\ 20 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 4 & 5 & 10 & 20 \end{array} \\ \begin{pmatrix} 1 & -1 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Allgemein für $(\mathbb{T}_n, |)$ mit $n \in \mathbb{N}$:

Sind $a, b \in \mathbb{T}_n$, so gilt:

$$\mu(a,b) = \begin{cases} (-1)^r & \text{falls } a|b \text{ und falls } \frac{b}{a} \text{ Produkt von } r \text{ verschiedenen Primzahlen} \\ 0 & \text{sonst.} \end{cases}$$

Bemerkung:

In der Zahlentheorie setzt man

$$\mu(m) := \mu(1,m) = \mu(a,b) \text{ für alle } a, b \text{ mit } \frac{b}{a} = m.$$

μ ist die **klassische Möbiusfunktion der Zahlentheorie**.

3.9 Satz (Möbiusinversion)

Sei $(M, \preceq)$ eine geordnete Menge, $f : M \longrightarrow \mathbb{C}$ eine beliebige komplexwertige Funktion. Dann gelten folgende Aussagen:

a) Die komplexwertige Funktion $g : M \longrightarrow \mathbb{C}$ sei definiert durch

$$g(x) = \sum_{\substack{y \in M \\ y \preceq x}} f(y).$$

Wir bemerken, dass die Funktion g eine Akkumulierung darstellt. Dann ist die Funktion f mittels der Funktion g und der Möbiusfunktion μ darstellbar:

$$f(x) = \sum_{\substack{y \in M \\ y \preceq x}} g(y)\mu(y,x) \text{ für alle } x \in M.$$

b) Sei $h : M \longrightarrow \mathbb{C}$ definiert durch

$$h(x) = \sum_{\substack{y \in M \\ y \succeq x}} f(y)$$

Dann ist

$$f(x) = \sum_{\substack{y \in M \\ y \succeq x}} h(y)\mu(x, y).$$

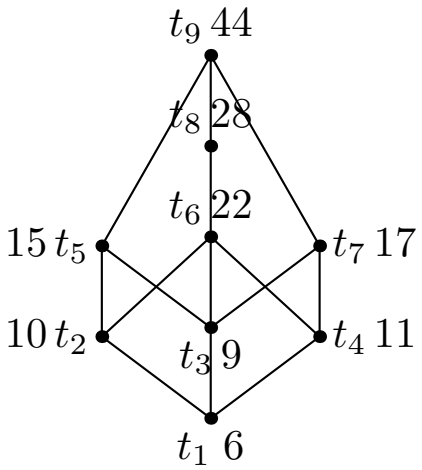
Beweis: a) Sei $x \in M$. Dann gilt:

$$\begin{aligned}
\sum_{\substack{y \in M \\ y \preceq x}} g(y)\mu(y, x) &= \sum_{\substack{y \in M \\ y \preceq x}} \left(\sum_{\substack{z \in M \\ z \preceq y}} f(z) \right) \cdot \mu(y, x) \\
&= \sum_{\substack{y \in M \\ y \preceq x}} \sum_{\substack{z \in M \\ z \preceq y}} f(z) \cdot \mu(y, x) \\
&\underset{\substack{\text{Ist } z \not\preceq y, \\ \text{so } \zeta(z,y)=0}}{=} \sum_{\substack{y \in M \\ y \preceq x}} \sum_{z \in M} f(z) \cdot \zeta(z, y) \cdot \mu(y, x) \\
&= \sum_{z \in M} f(z) \sum_{\substack{y \in M \\ y \preceq x}} \zeta(z, y)\mu(y, x) \\
&= \sum_{z \in M} f(z)\delta(z, x) = f(x)
\end{aligned}$$

b) Der Beweis erfolgt analog.

3.10 Beispiel

Zur Lösung einer Aufgabe sei die Lösung gewisser Teilaufgaben notwendig, für deren Lösung wieder die Lösungen gewisser Teilaufgaben notwendig sind usw. Dies führt zu einer partiellen Ordnung auf der Menge der (Teil-)Aufgaben $t_1, \ldots, t_n$. Das folgende Hasse-Diagramm stelle eine solche dar:



Bekannt sei die Zeit $g(t_i)$ zur Lösung der Teilaufgabe t_i **einschließlich** der Zeit für die dazu benötigten Teilaufgaben (im obigen Hasse-Diagramm neben t_i eingetragen).
Wie groß ist die "reine" Zeit $f(t_i)$ zur Lösung der Teilaufgabe t_i **ohne** die Lösungszeit der benötigten Teilaufgaben?
Wir bestimmen z.B. $f(t_6)$: Jeweils mittels 3.9 und 3.7(c) erhalten wir:

$$f(t_6) = g(t_6)\mu(t_6, t_6) + g(t_4)\mu(t_4, t_6) + g(t_3)\mu(t_3, t_6) + g(t_2)\mu(t_2, t_6) + g(t_1)\mu(t_1, t_6),$$

wobei

$$\begin{aligned}
\mu(t_6, t_6) &= 1 \\
\mu(t_4, t_6) &= \mu(t_3, t_6) = \mu(t_2, t_6) = -1 \\
\mu(t_1, t_6) &= -\mu(t_1, t_1) - \mu(t_1, t_2) - \mu(t_1, t_3) - \mu(t_1, t_4) \\
&= -1 + 1 + 1 + 1 = 2.
\end{aligned}$$

Also erhalten wir insgesamt:

$$f(t_6) = 22 \cdot 1 - 11 \cdot 1 - 9 \cdot 1 - 10 \cdot 1 + 6 \cdot 2 = 4.$$

Man beachte die Ähnlichkeit zum Einschließungs-Ausschließungsprinzip.

3.11 Definition

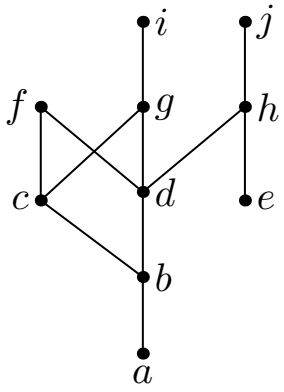
Sei $(M, \preceq)$ eine geordnete Menge. Wir definieren:

a) $K \subseteq M$ heißt eine **Kette**, wenn $(K, \preceq)$ total geordnet ist (vgl. 3.1(d)).

b) $A \subseteq M$ heißt eine **Antikette**, manchmal auch **Sperner-System** genannt, wenn für alle $a, a' \in A$ Folgendes gilt:

$$\text{Ist } a \neq a', \text{ so ist } a \not\preceq a' \text{ und } a' \not\preceq a.$$

Beispiel: Sei $(M, \preceq)$ gegeben durch folgendes Hasse-Diagramm:



Wir sehen: $\{a, b, c, g, i\}$ ist eine Kette, ebenfalls wie $\{a, d, j\}$. $\{f, g, j\}$ ist eine Antikette.

3.12 Bemerkungen

a) Sei K eine Kette und A eine Antikette in $(M, \preceq)$. Dann gilt $|A \cap K| \leq 1$.

b) Aus a) folgt: Hat M eine Kette K mit $|K| = k$, so kann M nicht mit weniger als k Antiketten überdeckt werden. (Überdeckung von M durch Teilmengen $M_1, M_2, \ldots, M_r$ bedeutet: $M = \bigcup_{i=1}^{r} M_i$. Man beachte: Man kann die Vereinigung disjunkt wählen (bei höchstens so großer Anzahl nicht-leerer M_i)).

c) Hat M eine Antikette A, $|A| = l$, so kann M nicht mit weniger als l Ketten überdeckt werden (folgt aus a).

Beispiel von oben:

Eine längste Kette K hat die Länge $|K| = \mathbf{5}$, z.B. die oben angegebene.

Mit wie vielen Antiketten lässt sich M überdecken?

Es reichen die **5** Antiketten $\{a, e\}, \{b\}, \{c, d\}, \{f, g, h\}, \{i, j\}$.

Eine längste Antikette A hat die Länge $|A| = \mathbf{3}$, z.B. die oben angegebene.

Mit wie vielen Ketten lässt sich jetzt M überdecken?

Es reichen **3** Ketten: $\{a, b, c, f\}, \{d, g, i\}, \{e, h, j\}$.

Diese Ergebnisse aus dem Beispiel sind kein Zufall. Es gilt immer:

"Die maximale Länge einer Kette (Antikette) = Maximalanzahl von Antiketten (Ketten), die zu einer Überdeckung von M benötigt werden".

Eine dieser beiden Aussagen ist einfach zu beweisen:

3.13 Satz

Sei $(M, \preceq)$ eine endliche geordnete Menge.

Hat die längste Kette von M die Größe k, so kann M mit k Antiketten überdeckt werden.

Beweis:

Wir definieren die **Höhe** $h(x)$ von $x \in M$ als die Maximalzahl der Elemente $\neq x$ einer Kette, deren größtes Element x ist. Also: $h(x) = 0$, falls x keinen Vorgänger hat.

Wir setzen

$$A_i = \{x \in M\colon h(x) = i\} \text{ für } i = 0, 1, \ldots.$$

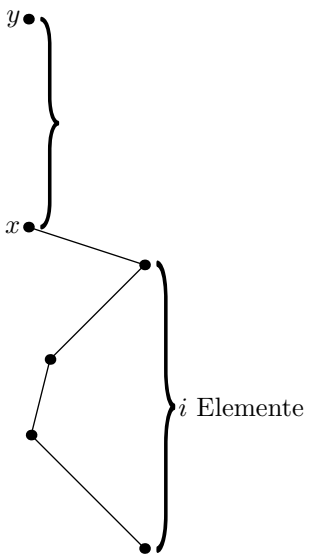
Nach Voraussetzung ist dann $A_i = \emptyset$ für $i \geq k$.

Da die Höhe für jedes Element aus M eindeutig festgelegt ist, erhalten wir unmittelbar:

$$M = A_0 \dot\cup \ldots \dot\cup A_{k-1},$$

d.h. M ist die disjunkte Vereinigung der A_i. Und jedes A_i ist Antikette, denn:

Angenommen es existieren $x, y \in A_i$, $x \neq y$, $x \prec y$. Dann haben wir die folgende Situation:



also gilt $h(y) \geq i + 1$, ein Widerspruch zu $y \in A_i$.

Bemerkung: So wurde auch im Beispiel eine Überdeckung durch Antiketten konstruiert.

3.14 Satz von Dilworth (1950)

(Robert P. Dilworth, 1914-1993, amerikanischer Mathematiker)

Sei $(M, \preceq)$ eine endliche geordnete Menge. Hat die größte Antikette von M die Größe l, so kann M durch l Ketten überdeckt werden.

Bemerkung: Dieser Satz wurde vorher schon von Gallai und Milgram entdeckt, aber nicht publiziert.

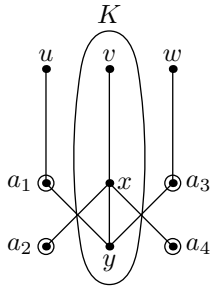
Beweis: (Tverberg, 1967)

Wir beweisen den Satz von Dilworth mittels vollständiger Induktion nach $|M|$:

Für $|M| = 1$ ist die Behauptung klar.

Sei K eine maximale Kette in M. Enthält jede Antikette in $(M \setminus K, \preceq)$ höchstens $l - 1$ Elemente, so ist die Behauptung per Induktionsvoraussetzung bewiesen.

Angenommen wir haben eine Antikette $\{a_1, \ldots, a_l\}$ der Länge l in $(M \setminus K, \preceq)$.

[Dazu ein Beispiel zur Anschauung: 

]

Dieses Beispiel wird im Folgenden immer wieder zur Veranschaulichung betrachtet!

Wir definieren für das Folgende:

$$M^- = \{x \in M\text{: es existiert } i \text{ mit } x \preceq a_i\} \supseteq \{a_1, \ldots, a_l\}$$
$$M^+ = \{x \in M\text{: es existiert } i \text{ mit } x \succeq a_i\} \supseteq \{a_1, \ldots, a_l\}.$$

[Das obige Beispiel ergibt:

$$M^- = \{a_1, a_2, a_3, a_4, y\}$$
$$M^+ = \{a_1, a_2, a_3, a_4, x, u, v, w\} \quad]$$

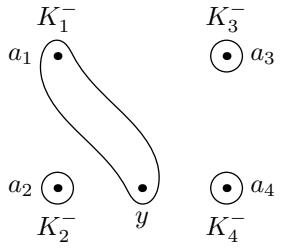
Es ist $M = M^- \cup M^+$, denn angenommen es existiert $a \in M$ mit $a \notin M^- \cup M^+$, so ist $\{a_1, \ldots, a_l, a\}$ Antikette, ein Widerspruch zur Maximalität.

Da K eine maximale Kette ist, liegt das größte Element von K nicht in M^-:

Denn angenommen doch: Sei v das größte Element von K und $v \preceq a_i$ für ein geeignetes i; dann gilt $v \prec a_i$, da $a_i \notin K$. Dann ist aber K verlängerbar durch a_i, ein Widerspruch zur Maximalität.

Also gilt $|M^-| < |M|$ und M^- enthält Antikette mit l Elementen. Per Induktion gilt dann $M^- = K_1^- \cup \cdots \cup K_l^-$, K_i^- Ketten, $a_i \in K_i^-$, $i = 1, \ldots, l$.

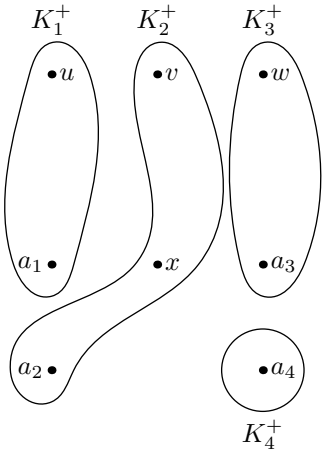
[Das Beispiel ergibt hier:



]

Wir zeigen, dass die a_i die maximalen Elemente der Ketten $K_i^-, i = 1, \ldots, l$ sind:
Denn angenommen es existiert $k \in M^-$ mit $k \succ a_i$ für ein i. Nach Definition von M^- existiert a_j mit $k \preceq a_j$. Dann ist $a_i \prec a_j$, ein Widerspruch, also gilt die Behauptung.
Wenn wir für M^+ dual vorgehen, erhalten wir analog $M^+ = K_1^+ \cup \cdots \cup K_l^+$, K_i^+ Ketten, a_i minimale Elemente von K_i^+, $i = 1, \ldots, l$.

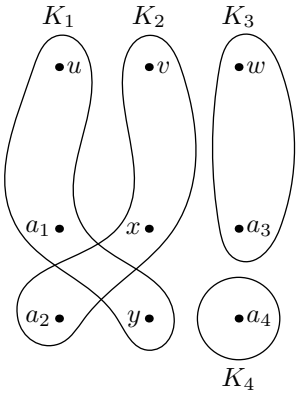
[Das Beispiel ergibt hier:



]

Dann gilt: $K_i = K_i^- \cup K_i^+$ Kette, $M = M^- \cup M^+ = \bigcup_{i=1}^{l} K_i$.

[Auch hier unser Beispiel:



]

Bemerkungen:

Der Satz von Dilworth ist im Wesentlichen äquivalent zu einer Reihe anderer wichtiger Sätze der Kombinatorik. Wir zeigen hier nur, wie sich ein wichtiger graphentheoretischer Satz, der **Satz von König**, aus dem Satz von Dilworth ergibt. Zur Vorbereitung führen wir einige Begrifflichkeiten ein:

3.15 Definition

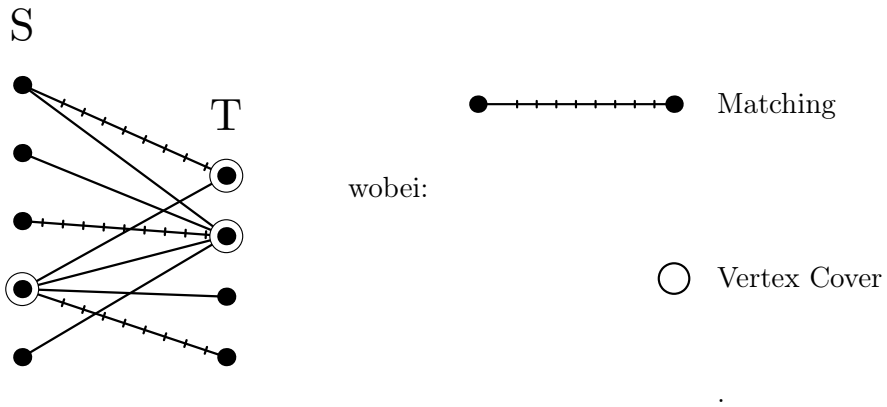
Sei $G = (E, K)$ ein Graph, E die Eckenmenge, K die Kantenmenge des Graphen.
Dabei setzen wir voraus, dass G ein **einfacher** Graph ist, d.h. er habe keine Mehrfachkanten und keine Schleifen; deshalb kann man K mit einer Teilmenge von $\mathcal{P}_2(E)$ identifizieren.

a) G heißt **bipartit**, falls es eine disjunkte Zerlegung $E = S \dot{\cup} T$ gibt, so dass jede Kante einen Endpunkt in S und einen Endpunkt in T hat.

b) $V \subseteq E$ heißt **Vertex Cover** (überdeckende Punktmenge), falls jede Kante mindestens einen Endpunkt in V hat.

c) Ein **Matching** ist eine Teilmenge $L \subseteq K$, so dass keine zwei Kanten in L einen gemeinsamen Endpunkt haben.

Beispiel



.

Klar ist: Wenn m die Größe eines Matchings und p die Größe eines Vertex Covers ist, so ist $m \leq p$.

3.16 Satz von König (1916)

(Dénes König (1884-1944), ungarischer Mathematiker)

Sei $G = (E, K)$ ein endlicher bipartiter Graph. Dann ist die maximale Größe $m(G)$ eines Matchings gleich der minimalen Größe $p(G)$ eines Vertex Cover.

Beweis:

Klar ist: $m(G) \leq p(G)$.

Es ist $E = S \dot{\cup} T$; G ist bzgl. S, T bipartit.

Wir definieren auf E eine Relation $\preceq$ durch

$$e \preceq e \text{ für alle } e \in E$$

$$s \preceq t :\Longleftrightarrow \text{ es gibt eine Kante zwischen } s \text{ und } t, \ (s \in S, t \in T)$$

(ansonsten keine $\preceq$ -Beziehung)

Die Relation $\preceq$ ist eine Ordnungsrelation.

Sei $V \subseteq S \dot{\cup} T$ ein Vertex Cover. Dann ist $E \setminus V$ Antikette bzgl. $\preceq$; diese Aussage ist klar. Es gilt aber auch die Umkehrung: Denn sei A eine Antikette; dann ist zu zeigen, dass $E \setminus A$ ein Vertex Cover ist. Angenommen es existiert eine Kante (s, t) mit $s, t \notin E \setminus A$. Dann folgt $s, t \in A$, ein Widerspruch, da $s \preceq t$.

Wir erhalten also

$$|S| + |T| - p(G) = \text{ maximale Größe einer Antikette}$$

$$\underset{3.15}{=} \text{ minimale Anzahl von disjunkten Ketten, die } E \text{ überdecken.}$$

In $(E, \preceq)$ gibt es nur 1- oder 2-elementige Ketten.

Es mögen in der Zerlegung k_1 1-elementige und k_2 2-elementige Ketten vorkommen. Dann

gelten:

$$k_1 + k_2 = |S| + |T| - p(G)$$
$$k_1 + 2k_2 = |S| + |T|.$$

Und damit gilt $k_2 = p(G)$. Die 2-elementigen Ketten bilden ein Matching von G. Also gilt $m(G) \geq p(G)$. Und damit erhalten wir wie gewünscht wegen $m(G) \leq p(G)$ die Gleichheit $m(G) = p(G)$.

4 Scheduling

(Dieses Kapitel orientiert sich bis 4.11 an [I], Kap. VI.)

Das Gebiet "Scheduling"(= Ablaufplanung) ist ein Spezialgebiet der Kombinatorischen Optimierung.

Allgemeines Problem:

Gegeben ist eine Menge von Jobs (auch Aufgaben, Tätigkeiten etc. genannt) und eine Menge von Maschinen (Prozessoren), auf denen diese Jobs (oder auch abhängig von der Maschine nur gewisse Jobs) ausgeführt werden können.

Die Zuordnung der Jobs auf die Maschinen soll unter Einhaltung gewisser Randbedingungen bzgl. gewisser Ziele optimiert werden.

Zunächst beschäftigen wir uns mit den **Job-Scheduling-Problemen**, später dann kurz noch mit den sogenannten **Intervall-Scheduling-Problemen**.

Für die Job-Scheduling-Probleme gehen wir von **folgender Situation** aus (welche nicht die allgemeinste ist!):

4.1 Definition

a) Gegeben ist eine Menge J von **Jobs**; diese bilden eine (partiell) geordnete Menge $(J, \preceq)$, wobei

$$i \prec j :\Longleftrightarrow \text{ Der Job } j \text{ kann begonnen werden, wenn der Job } i \text{ beendet ist.}$$

Jeder Job kann auf einer von m **Maschinen** durchgeführt werden. Auf jeder Maschine benötigt ein Job j dieselbe Dauer $d(j)$. Dabei setzen wir voraus: Jeder Job muss ohne Unterbrechung durchgeführt werden.

b) Ein **Ablaufplan (Schedule)** ordnet jedem Job j einen **Anfangszeitpunkt** $x(j) \geq 0$ und eine Maschine $m(j) \in \{1, \ldots, m\}$ zu. **Endzeitpunkt** für den Job j ist dann $y(j) = x(j) + d(j)$.

Folgende **Bedingungen** sollen erfüllt sein:

- Eine Maschine kann nicht mehrere Jobs gleichzeitig bearbeiten, d.h. frühestens zum Endzeitpunkt $y(j)$ kann auf der Maschine $m(j)$ ein neuer Job begonnen werden.
- Es ist die sog. **Vorgängerrelation** einzuhalten:

  Ist $i \prec j$, so ist $y(i) \leq x(j)$; dabei ist $\leq$ die normale Kleiner-Gleich-Relation.

4.2 Job-Scheduling-Problem

Beim Job-Scheduling-Problem geht es darum, einen **optimalen Ablaufplan** zu finden. Dabei kann Optimalität bzgl. unterschiedlicher Ziele definiert sein.
Z.B.

(1) Minimiere die **Ablaufzeit** $T := \max\{y(j) : j \in J\}$.

(2) Halte für jeden Job j einen Fertigstellungstermin ein bzw. minimiere die Gesamtverspätung.

(3) Minimiere die Anzahl der verspäteten Jobs.

(4) Minimiere die maximale Verspätung.

(5) Minimiere die Leerzeiten der Maschinen.

etc.
Wir befassen uns hier zunächst mit (1):

$$\text{Minimiere } T := \max\{y(j) : j \in J\}.$$

Für dem Fall $m = 1$ (1-Maschinenplan) ist das Problem trivial. Wir ordnen die Jobs vollständig (als Kette) bzgl. einer vollständigen Ordnung $\leq$, so dass gilt: Ist $i \preceq j$, so ist

$i \leq j$. Dann führen wir die Jobs entsprechend der vollständigen Ordnung ohne Leerzeiten der Maschine hintereinander aus.
Dass eine solche Ordnung $\leq$ möglich ist, folgt aus:

4.3 Satz (Topologisches Sortieren)

Sei $(M, \preceq)$ eine endliche partiell geordnete Menge. Dann existiert eine vollständige Ordnung $\leq$ auf M, die eine Verfeinerung von $\preceq$ ist, d.h. es gilt: Ist $i \preceq j$, so ist auch $i \leq j$.
Beweis:
Da M endlich ist, existiert ein Element $i_0 \in M$, das keine Vorgänger besitzt, also die Höhe 0 hat. Wähle i_0 als kleinstes Element bezüglich $\leq$ und wende Induktion auf $(M \setminus \{i_0\}, \preceq)$ an.

Man kann zeigen, dass 4.3 auch für beliebige partiell geordnete (unendliche) Mengen gilt, wenn man das Auswahlaxiom zulässt.

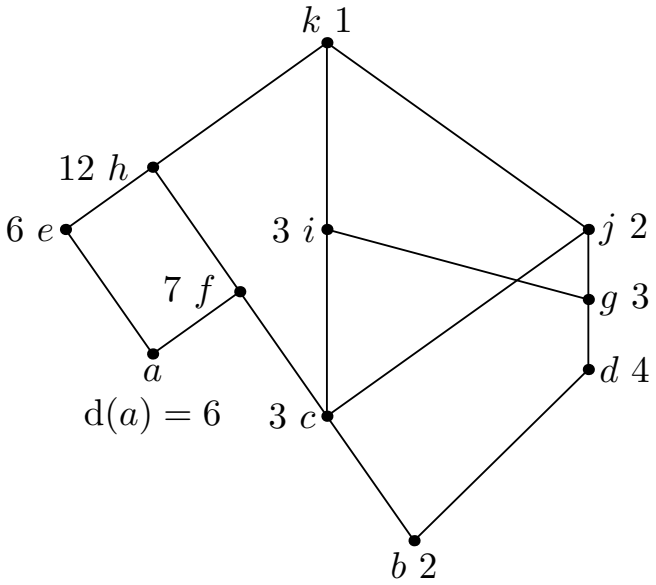
Wir werden im Folgenden $\boldsymbol{m \geq 2}$ annehmen. Dann ist das Job-Scheduling-Problem 4.2(1) $\mathcal{NP}$-hart, d.h. es ist nicht zu erwarten polynomielle Algorithmen zu finden.
Wir geben einen Algorithmus an, der schnell ist, aber im Allgemeinen nicht eine optimale Lösung liefert:

4.4 Listenplan-Algorithmus

Gegeben sei eine Liste L (d.h. eine total geordnete Menge) der Jobs (jeder Job tritt genau einmal in der Liste auf). Dabei muss L nicht eine Verfeinerung von $\preceq$ sein!
Der zugehörige Listenplan (dies ist ein Ablaufplan) wird folgendermaßen erstellt:
Auf frühestmöglichen noch freien Anfangszeitpunkt im Plan (d.h. frühestmöglicher Zeitpunkt einer Maschine, die noch frei ist) wird der erste noch in der Liste befindliche Job j gesetzt, der zu diesem Zeitpunkt eingeplant werden kann (d.h. alle Jobs i mit $i \prec j$ sind

abgeschlossen). Dieser Job j wird gestrichen. Wiederhole diesen Vorgang bis alle Jobs im Plan untergebracht sind.

4.5 Beispiel

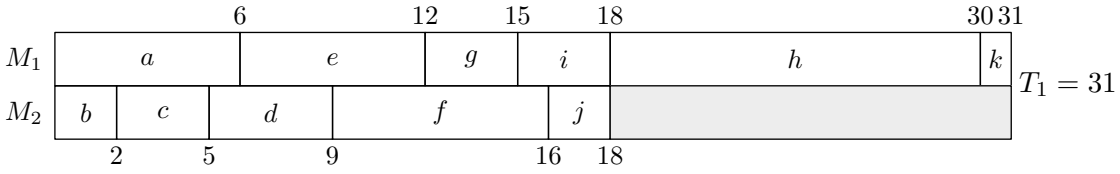


Die Zahlen neben den Jobbezeichnungen geben die Dauern des Jobs an.

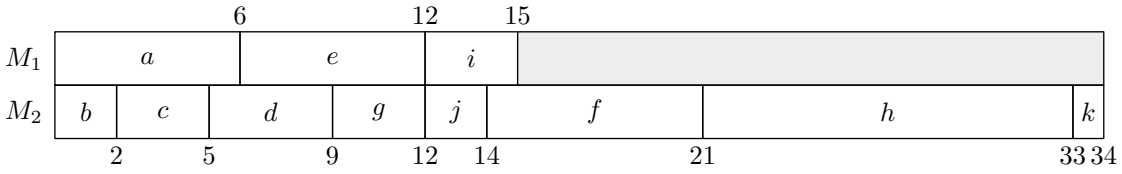
$m = 2$ **(2-Maschinenplan)**:

Gegeben ist die Liste $L_1 = (a\ b\ c\ d\ e\ f\ g\ i\ j\ h\ k)$.

Dann erhalten wir das folgende sogenannte **Gantt-Diagramm**, wobei ein Gantt-Diagramm oder Balkenplan ein nach dem Unternehmensberater Henry L. Gantt (1861–1919) benanntes Instrument des Projektmanagements ist, das die zeitliche Abfolge von Aktivitäten grafisch in Form von Balken auf einer Zeitachse darstellt:

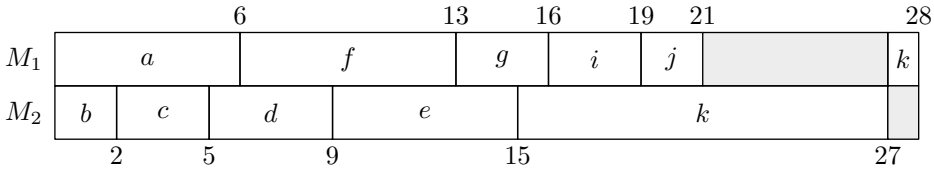


Liste $L_2 = (g\ i\ a\ c\ h\ b\ k\ j\ e\ d\ f)$



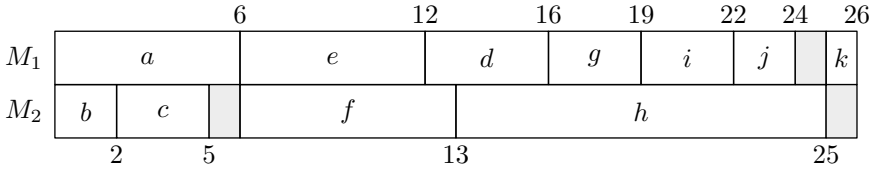
$T_2 = 34$

Liste $L_3 = (a\ b\ c\ d\ f\ e\ g\ h\ i\ j\ k)$



$T_3 = 28$

Optimaler Ablaufplan:



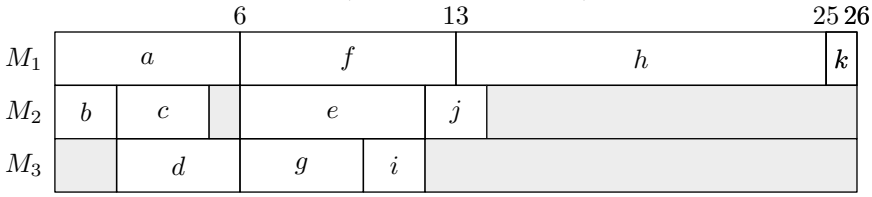
$T_{\text{opt}} = 26$

(Es geht nicht besser, da die Jobs a, f, h, k nacheinander abgearbeitet werden müssen und deren Gesamtdauer beträgt schon 26 Zeiteinheiten.)

Dieser Plan ist als Listenplan nicht zu erhalten, da dort eine Maschine nie eine Pause machen darf, wenn es noch einen mögliche Job gibt. Nach b, c kann man immer d ausführen!

$m = 3$ **(3-Maschinenplan)**:

Gegeben ist die Liste $L = (a\ b\ c\ d\ f\ e\ h\ i\ j\ k)$



Optimale Zeit, aber 2 der 3 Maschinen sind schlecht ausgelastet.

Nach 4.5 können Listenpläne im Allgemeinen nicht optimal sein (sie sind aber schnell konstruierbar). Es gilt jedoch:

4.6 Satz von Graham (1966)

Gegeben sind m Maschinen und eine partiell geordnete Jobmenge $(J, \preceq)$ mit der Ausführungsdauer $d(j)$ für jeden Job $j \in J$.
Für die Ablaufzeit T_{opt} eines optimalen Plans (d.h. minimaler Laufzeit) und die Ablaufzeit T eines beliebigen Listenplans gilt

$$\frac{T}{T_{opt}} \leq 2 - \frac{1}{m}.$$

(D.h. z.B. für $m = 2$: Der schlechteste Listenplan hat höchstens um 50% höhere Laufzeit als ein optimaler Plan.)

Beweis:

Sei c_1 ein Job ganz rechts im Listenplan. Unter allen Vorgängern von c_1 sei c_2 derjenige, der am spätesten beendet wird. Dann darf zwischen $y(c_2)$ und $x(c_1)$ keine Maschine pausieren (sonst hätte man diese nach $y(c_2)$ mit c_1 beauftragen können, denn alle Vorgänger von c_1 sind zu diesem Zeitpunkt abgeschlossen).
Führe dasselbe Argument mit c_2 durch etc.
Das liefert eine Kette K (bezüglich $\preceq$) von Jobs $c_1 \prec c_{k-1} \cdots \prec c_2 \prec c_1$, so dass immer, wenn eine Maschine pausiert, einer der Jobs aus K durchgeführt wird.
Sei $d(K)$ die Summe der Dauern der Jobs in K. Die Summe der Pausenzeiten des Plans sei T_0. Da höchstens $m - 1$ Maschinen gleichzeitig pausieren können, gilt:

$$(*) \qquad T_0 \leq (m-1) \cdot d(K).$$

Da K eine Kette ist, können auch in einem optimalen Plan nie zwei Jobs aus K gleichzeitig

durchgeführt werden. Also

$$(**) \qquad d(K) \leq T_{opt}.$$

Sei $d(J) := \sum_{c \in J} d(c)$. Dann erhalten wir:

$$\begin{aligned} T &= \frac{1}{m} \underbrace{(d(J) + T_0)}_{\substack{\text{Gesamtlaufzeit aller Maschinen; alle Maschinen laufen einschließlich Pausen gleich lang.}}} \\ &\underset{(*) \text{ u. } (**)}{\leq} \frac{1}{m}(d(J) + (m-1) \cdot T_{opt}) \\ &\leq \frac{1}{m}(m \cdot T_{opt} + (m-1) \cdot T_{opt}), \text{ da } T_{opt} \geq \frac{d(J)}{m} \\ &= (2 - \frac{1}{m}) T_{opt}. \end{aligned}$$

Bemerkung: Die Schranke in 4.6 ist scharf für jedes m.

4.7 Bemerkung

Wie oben gesagt, sind keine polynomiellen Algorithmen für das Job-Scheduling-Problem 4.2(1) für m Maschinen bekannt, falls $m \geq 2$.
Sind alle Jobdauern $d(i)$ alle gleich, gibt es für $m = 2$ einen $\mathcal{O}(n^2)$-Algorithmus von Coffman und Graham (1972), der den optimalen Ablaufplan berechnet (siehe [I], VI.4).

Wir betrachten jetzt Einmaschinenprobleme. Das Job-Scheduling-Problem ist dann nach 4.3 trivial. Aber es gibt hier auch nicht-triviale Probleme, z.B.:

4.8 Situation

Gegeben sei die Situation 4.1 mit einer Maschine ($m = 1$). Zusätzlich sei für jeden Job i ein Fertigstellungstermin $f(i)$ vorgegeben, der eingehalten werden soll.
Gesucht wird ein Ablaufplan, der die **maximale Verspätung** $\max\{y(i) - f(i)\colon i \in J\}$ minimiert.
Dazu formulieren wir

4.9 Algorithmus von Lawler (1973)

Gegeben sei die Situation 4.8.
Folgender Algorithmus berechnet, in umgekehrter Reihenfolge, eine Liste $i_1, \ldots, i_n$ der Jobs (wobei gilt: Ist $i_j \preceq i_k$, so $j \leq k$ mit der normalen Ordnung $\leq$ auf $\mathbb{N}$), mit kleinstmöglicher maximaler Verspätung.
Algorithmus:

(1) Berechne $n(i) := |\{j \in J : j \succ i\}|$ für alle $i \in J$.

(2) Sei $n = |J|$. Setze $t := n$.

(3) Wähle unter den Jobs i mit $n(i) = 0$ einen mit maximalen Wert $f(i)$, etwa den Job l.

(4) Setze $i_t = l$.

(5) Falls $t = 1$, STOP.

Sonst setze $n(i) := n(i) - 1$ für alle $i \preceq l$ und $t := t - 1$. Mache weiter mit (3).

Man beachte:

1) Schon verwendete Jobs erhalten den Wert -1 während der Durchführung des Algorithmus.

2) Es existiert zu jedem Zeitpunkt ein Job j mit $n(j) = 0$, da die Anzahl der Jobs endlich ist.

3) Der Ablaufplan $i_1, \ldots, i_n$ ist tatsächlich auf nur einer Maschine durchführbar:

   Gegeben sei i_k. Ist $i_k \succ i_r$ für ein $r > k$ (d.h. i_r wird für i_k benötigt, steht aber in der Liste hinter i_k)?

   Nein, wegen (3) und (5):
   Denn falls $n(l) = 0$, so wird der Job l für keinen noch nicht verwendeten Job mehr benötigt. Ist $i_k \succ i_r$, so ist $n(i_r) \neq 0$, kann also nicht im Algorithmus hinter i_k in der Liste eingebaut worden sein.

4.10 Beispiel

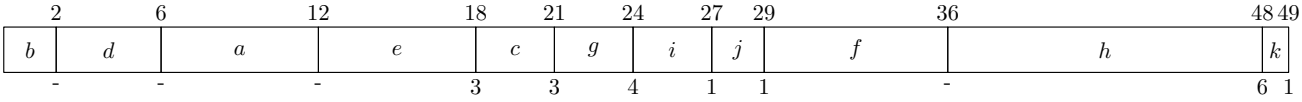
Wir verwenden Beispiel 4.5, wobei die Fertigstellungstermine gegeben sind durch die Tabelle:

Job u	a	b	c	d	e	f	g	h	i	j	k
$f(u)$	12	20	18	8	15	40	20	42	28	30	48

Man beachte: $\sum_{u \in J} d(u) = 49$. Deswegen wird es Verspätungen geben, da k als letzter Job durchgeführt wird.

Nun zur Visualisierung der Durchführung des Algorithmus von Lawler:

Job u	$n(u)$											
a	4	3	2	1	1	1	1	1	0←	-1	-1	-1
b	8	7	6	5	4	3	2	1	1	1	0←	-1
c	5	4	3	2	1	0	0←	-1	-1	-1	-1	-1
d	4	3	3	3	2	1	0	0	0	0←	-1	-1
e	2	1	0	0	0	0	0	0←	-1	-1	-1	-1
f	2	1	0←	-1	-1	-1	-1	-1	-1	-1	-1	-1
g	3	2	2	2	1	0←	-1	-1	-1	-1	-1	-1
h	1	0←	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
i	1	0	0	0	0←	-1	-1	-1	-1	-1	-1	-1
j	1	0	0	0←	-1	-1	-1	-1	-1	-1	-1	-1
k	0←	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1



Verspätung:

Die kleinste maximale Verspätung beträgt 6 (Zeiteinheiten).

4.11 Satz

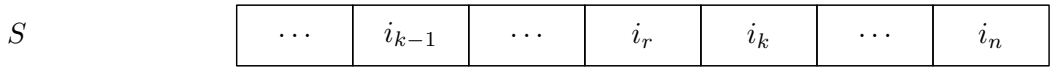
Der Algorithmus 4.9 berechnet einen Ablaufplan (für eine Maschine) mit kleinstmöglicher maximaler Verspätung in $\mathcal{O}(n^2)$. Insbesondere berechnet der Algorithmus einen Plan ohne Verspätung, wenn es einen solchen gibt.

Beweis:

Sei $R : i_1, \ldots, i_n$ der vom Algorithmus berechnete Ablaufplan, $S : j_i, \ldots, j_n$ ein optimaler Plan, der an möglichst vielen Stellen rechts (ununterbrochen) mit R übereinstimmt.

Angenommen $R \neq S$. Sei $i_k = j_k, \cdots, i_n = j_n, i_{k-1} \neq j_{k-1}$.

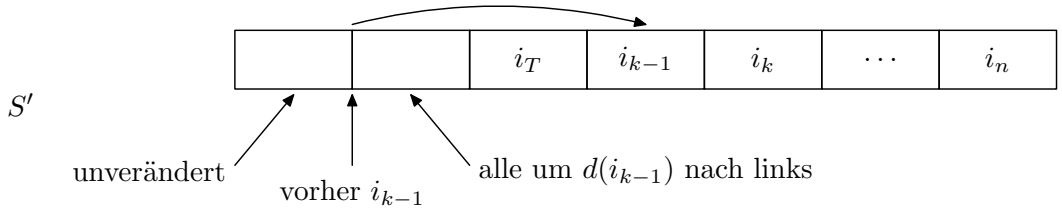
Sei $j_{k-1} = i_r$:



i_r und i_{k-1} können beide unmittelbar vor i_k eingeplant werden (da R, S zulässige Pläne), d.h. $n(i_r) = n(i_{k-1}) = 0$ bei dieser Stelle des Algorithmus.

Nach Wahl von i_{k-1} in 4.9 ist $f(i_r) \leq f(i_{k-1})$.

Wandle nun S um zu S':



Die Verspätung von i_{k-1} ist nicht größer als vorher die Verspätung von i_r, da $f(i_{k-1}) \geq f(i_r)$. Alle anderen Jobs erhalten die gleiche oder eine geringere Verspätung. Also ist auch S' optimal, stimmt aber an den Stellen $k-1, \ldots, n$ mit R überein. Das ist ein Widerspruch zur Wahl von S. Also erhalten wir $R = S$. Die Komplexität $\mathcal{O}(n^2)$ ist klar; die Vorgängerrelation $\preceq$ ist als Matrix der Zeta-Funktion gegeben.

Wir kommen jetzt zum sog. **Intervall-Scheduling**:

4.12 Definition

Gegeben sind identische Maschinen und n voneinander unabhängige Jobs. Jede Maschine kann jeden Job bearbeiten, aber nie mehr als einen gleichzeitig.

Jeder Job i muss in einem festen Zeitintervall (s_i, t_i) durchgeführt werden; $d(i) = t_i - s_i$ ist die **Dauer des Jobs**.

Intervall-Scheduling-Problem: Bestimmung der Minimalanzahl von Maschinen, die benötigt werden, um alle Jobs durchzuführen.

Ein Beispiel: Ferienhäuser, Buchungszeiträume von Gästen.
$\widehat{=}$ Maschinen, $\widehat{=}$ Jobs

4.13 Bemerkung

Gegeben ist die Situation 4.12. Wir definieren eine partielle Ordnung $\preceq$ auf der Menge der Jobs durch: $i \prec j :\Leftrightarrow t_i \leq s_j$. Durch $\preceq$ wird J eine partiell geordnete Menge. Jede Kette in $(J, \preceq)$ kann auf einer Maschine durchgeführt werden (und nur Ketten können auf einer Maschine durchgeführt werden).

Zur **Lösung des Invervall-Scheduling-Problems**:

Die Minimalzahl von Ketten, die $(J, \preceq)$ überdecken

$=$ Maximalgröße von Antiketten in $(J, \preceq)$ (wegen 3.14 (Satz von Dilworth))

$=$ Maximalanzahl von Jobs, deren Zeitintervalle sich paarweise überschneiden.

Bemerkung:

Es gibt polynomielle Algorithmen für das Überdeckungsproblem von partiell geordneten Mengen mit einer Minimalanzahl von Ketten (Ford, Fulkerson (1962)).

In der speziellen Situation des Intervall-Scheduling gibt es einen $\mathcal{O}(n \cdot \log(n))$-Algorithmus:

4.14 Algorithmus von Gupta, Lee, Leung (1979)

(1) Ordne die $2n$ Zahlen $s_1, \ldots, s_n, t_1, \ldots, t_n$ in aufsteigender Reihenfolge $u_1 \leq u_2 \leq \cdots \leq u_{2n}$.

Falls $u_i = u_{i+1} = \cdots = u_{i+k}$, so zunächst t_j's, dann s_l's.

(2) Bilde 'Stapel' von Maschinen (höchstens n) und Liste l der Länge n.

(3) Setze $m = 0$ und counter $= 0$.

(4) Für $i = 1, \ldots, 2n$:

Ist $u_i = s_k$ für ein k, so setze oberste Maschine des Stapels auf $l[k]$. (Diese Maschine erledigt jetzt Job k).

Setze:

$$\text{counter} = \text{counter} + 1$$

$$m = \max(counter, m).$$

Ist $u_i = t_j$ für ein j, so setze Maschine auf $l[j]$ zurück auf den Stapel (oben) (Maschine hat Job j erledigt, ist jetzt wieder frei).

Setze dann:

$$\text{counter} = \text{counter} - 1.$$

Ausgabe: m=Minimalanzahl der benötigten Maschinen.

4.15 Satz

Der Algorithmus 4.14 ist korrekt und hat die Komplexität $\mathcal{O}(n \cdot \log(n))$.

Beweis:

$$\begin{aligned} m &= \text{maximale Anzahl gleichzeitig arbeitender Maschinen} \\ &= \text{maximale Anzahl paarweise überlappender Intervalle} \\ &= \text{maximale Größe einer Antikette der Jobs} \\ &\underset{\text{s.v.}}{=} \text{Minimalanzahl benötigter Maschinen.} \end{aligned}$$

Komplexität: (1) liefert $\mathcal{O}(n \cdot \log((n))$, der Rest geht mit $\mathcal{O}(n)$.

Beachte: Der Algorithmus liefert auch eine Zuordnung der Jobs zu den minimal vielen benötigten Maschinen.

4.16 Beispiel

Gegeben sind die Intervalle:

(0,4)	(1,7)	(2,5)	(4,9)	(6,12)	(8,10)	(11,13)
$\widehat{=}$	$\widehat{=}$	$\widehat{=}$	$\widehat{=}$	$\widehat{=}$	$\widehat{=}$	$\widehat{=}$
(s_1, t_1)	(s_2, t_2)	(s_3, t_3)	(s_4, t_4)	(s_5, t_5)	(s_6, t_6)	(s_7, t_7)

Wir ordnen die Intervallgrenzen gemäß des Algorithmus, berechnen counter und bestimmen m:

Ordnung:	s_1=0,	s_2=1,	s_3=2,	$t_1 = 4$,	s_4=4,	t_3=5,	s_5=6
counter:	1	2	3	2	3	2	3
m	1	2	3	3	3	3	3
Ordnung:	t_2=7,	s_6=8,	t_4=9,	t_6=10,	s_7=11,	t_5=12,	t_7=13
counter:	2	3	2	1	2	1	0
m	3	3	3	3	3	3	3

Für die Liste l der Länge 7 erhalten wir dann schlussendlich:

l:	m_1	m_2	m_3	m_1	m_3	m_2	m_2
	1	2	3	4	5	6	7

Es werden also **mindestens** 3 Maschinen benötigt.

5 Literatur

[A] Aigner, M., Diskrete Mathematik, Vieweg, 1999.

[C] Cameron, P.J., Combinatorics: Topics, Techniques, Algorithms. Cambridge University Press, 1994.

[DKR] Diekert, V., Kufleitner M., Rosenberger G., Elemente der diskreten Mathematik - Zahlen und Zählen, Graphen und Verbände. de Gruyter, 2013.

[H] Hauck, P., Kombinatorische Methoden in der Informatik. Vorlesungsskript (dm.inf.uni-tuebingen.de/skripte/Kombinatorik/KombinatorikSS2008.pdf).

[I] Ihringer, T., Diskrete Mathematik. Heldermann, 2002.

[M] Martin, G.E., Counting. The Art of Ennumerative Combinatories. Springer, 2009.

[MN] Matousek, J., Nesetril J., Diskrete Mathematik. Springer 2002.

[WHK] Wolff, M.P.H., Hauck P., Küchlin W., Mathematik für Informatik und BioInformatik. Springer, 2004.