Formale Grundlagen der Informatik

WiSe 2019/20

Universität Trier

Henning Fernau
Stefan Hoffmann
Universität Trier
fernau@uni-trier.de

Formale Grundlagen der Informatik Gesamtübersicht

1. Rechnen: Gesetze und Regeln
2. Modellieren und Formalisieren:
   Keine Angst vor Formalismen . . . und etwas Logik
3. Warum stimmt das eigentlich ? Logik und Beweisverfahren
4. Herangehen an Aufgaben aus Informatik und Mathematik

Organisatorisches für MO, MI, DO, FR

Vorlesung
8.30 bis 12.00 im F 55 auf Campus II (mit Pause)

Gruppenarbeit und Sprechstunde
12-14 Uhr: Gelegenheit zur Gruppenarbeit zu ausgewählten Übungen
Wir haben auch den H 13 für Sie reserviert
parallel dazu: 13:30-14:15 Uhr **Sprechstunde**: Wir sind im F 55 für Sie da.

Übung 14-16 Uhr, d.h.: 14.15-15.45 im F 55
(Stefan Hoffmann / Alexander Urbanek)

Am Abend überdenken Sie bitte den Stoff; sie erhalten auch Gelegenheit zu weiteren Übungen.
Fragen sind auch vor den Vorlesungen möglich (ab 8:15 Uhr).

Achtung: Die Mensa schließt während der "vorlesungsfreien Zeit" um 13.30.

Formalismen

eine “sprachlose Kunst”

Algorithm 3.7.1 Let us call a nonterminal A **erasable** if $A \Rightarrow^* e$. It is easy to see that a nonterminal A is erasable if and only

1. There is a rule $A \to e$ in G, or
2. There is a rule $A \to B_1B_2 \dots B_n$ in G, where all nonterminals $B_1, B_2, \dots, B_n$ are erasable.

Using the above characterization of the erasable nonterminals, apply the following procedure to find the set E of all erasable nonterminals:

1. Set $E := \emptyset$.
2. Add to E all nonterminals in the left-hand sides of rules satisfying 1.
3. While there is a rule $A \to B_1B_2 \dots B_n \in R$ with $B_1, B_2, \dots, B_n \in E$ and $A \notin E$, add A to E.

Now, for every rule $A \to \alpha$ in G, Algorithm 3.7.1 adds to R all rules obtained by eliminating one or more erasable nonterminals in α. For example, if G contains the rule $A \to BCD$ and C and D are erasable, add the rules

$$\begin{aligned} A &\to B \\ A &\to BC \\ A &\to BD. \end{aligned}$$

Finally, eliminate from R all rules of the type $A \to e$ and terminate.

End Algorithm

The result of applying Algorithm 3.7.1 to G is a grammar G_1 that does not generate e. Other than that, any derivation in G can obviously be simulated by a derivation in G_1. Thus, $L(G) - \{e\} = L(G_1)$.

Algorithm 3.7.2 operates similarly to Algorithm 3.7.1, using nonterminals instead of e.

Algorithm 3.7.2 For any nonterminal A in G, let $NT(A)$ be the set $\{B | B \in NT, A \Rightarrow^* B\}$ of all nonterminals derivable from A. Find the set $NT(A)$ using a procedure similar to the one used by Algorithm 3.7.1 to find E:

1. Set $NT(A) := \{A\}$.
2. While there is a rule $B \to C$ with $B \in NT(A)$ and $C \notin NT(A)$, add C to $NT(A)$.

For every pair (A, B) such that $B \in NT(A)$ and every rule $B \to \alpha$ with $\alpha \notin NT$, add the rule $A \to \alpha$ to R. For example, if G contains the rules $B \to CD$ and $NT(A) = \{A, B\}$, then the rule $A \to CD$ is added to R. Finally, eliminate all rules of the type $A \to B$ from R and terminate.

End Algorithm

It is clear that any derivation in G can be simulated by a derivation in the grammar obtained by application of Algorithm 3.7.2. If G contains the rules $A \to B$ and $B \to \alpha$, then replacing them by $A \to \alpha$ creates the same opportunities as their combination.

Algorithm 3.7.3 This algorithm converts any rule $A \to \alpha$ with $|\alpha| > 1$ to a group of rules $B \to C_1C_2$, where C_1 and C_2 are nonterminals. Given any rule $A \to B_0B_1 \dots B_k$, where $B_i \in (\Sigma \bigcup NT), 1 \le i \le k$, the algorithm converts it to the rules

$$\begin{aligned} A &\to X_{B_0}X_{B_1B_2\dots B_k} \\ X_{B_1B_2\dots B_k} &\to X_{B_1}X_{B_2B_3\dots B_k} \\ X_{B_2B_3\dots B_k} &\to X_{B_2}X_{B_3B_4\dots B_k} \\ &\vdots \\ X_{B_{k-1}B_k} &\to X_{B_{k-1}B_k} \end{aligned}$$

where $X_{B_0}, X_{B_1}, \dots, X_{B_k}, X_{B_1B_2\dots B_k}, X_{B_2B_3\dots B_k}, \dots, X_{B_{k-1}B_k}$ are new nonterminals. Now, if B_i is a nonterminal, then the algorithm uses just B_i instead of X_{B_i}. If B_i is a terminal, the algorithm adds the rule $X_{B_i} \to B_i$ to the set of rules.

This step is described by the following example. The rule $A \to BabDBE$ is replaced by

$$\begin{aligned} A &\to BX_{abDBE} \\ X_{abDBE} &\to X_aX_{bDBE} \\ X_{bDBE} &\to X_bX_{DBE} \\ X_{DBE} &\to DX_{BE} \\ X_{BE} &\to BE \\ X_a &\to a \\ X_b &\to b. \end{aligned}$$

The new nonterminals are used in the above set of rules for $A \to BabDBE$ only. Thus, the new set of rules is equivalent to the given rule. A string w can be derived from $A \to BabDBE$ if and only if it can be derived from the set of rules replacing it. The obtained set of rules satisfies the required condition, so terminate.

End Algorithm

To complete the proof of the theorem, we consecutively apply Algorithms 3.7.1, 3.7.2, and 3.7.3 to the grammar G. The grammar G_1 obtained by application of

Formalismen

eine “sprachlose Kunst”

no determinista N cuyos estados son los propios elementos. Hay una transición de $A \to \alpha\cdot X\beta$ a $A \to \alpha X\cdot\beta$ etiquetada con X, y hay una transición de $A \to \alpha\cdot B\beta$ a $B \to \cdot\gamma$ etiquetada con ϵ. Entonces, *cerradura*(I) para el conjunto de elementos (estados de N) I es exactamente la *cerradura*-ϵ de un conjunto de estados de AFN definida en la sección 3.6. Por tanto, *ir_a*(I, X) da la transición desde I con el símbolo X en el AFD construido a partir de N por la construcción de subconjuntos. Así considerado, el procedimiento *elementos*(G') de la figura 4.34 es simplemente la propia construcción de subconjuntos aplicada al AFN N construido a partir de G', como ya se ha descrito.

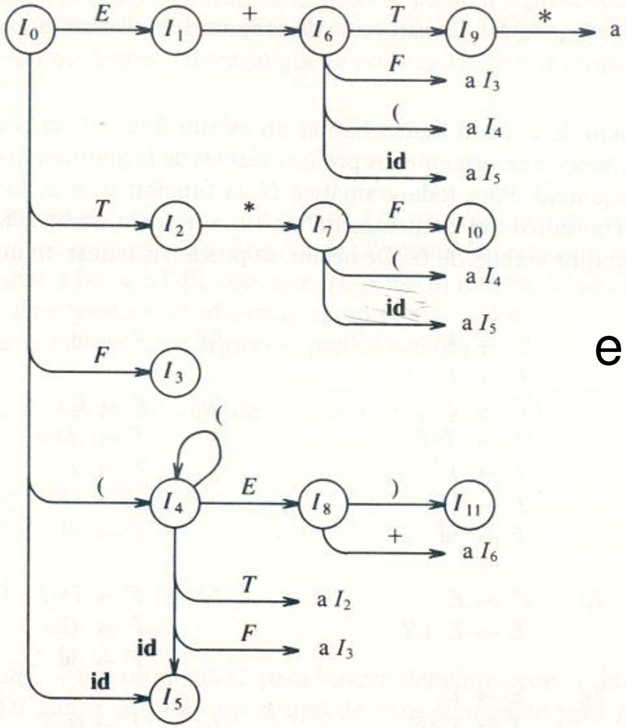


Fig. 4.36. Diagrama de transiciones del AFD D para los prefijos viables.

Elementos válidos. Se dice que un elemento $A \to \beta_1\cdot\beta_2$ es *válido* para un prefijo viable $\alpha\beta_1$ si existe una derivación $S' \overset{*}{\underset{md}{\Rightarrow}} \alpha A w \underset{md}{\Rightarrow} \alpha\beta_1\beta_2 w$. En general, un elemento será válido para muchos prefijos viables. El hecho de que $A \to \beta_1\cdot\beta_2$ sea válido para $\alpha\beta_1$ informa sobre si desplazar o reducir cuando se encuentre $\alpha\beta_1$ en la pila de análisis sintáctico. En concreto, si $\beta_2 \neq \epsilon$, entonces indica que aún no se ha desplazado el mango hacia la pila, así que el movimiento debe ser desplazar. Si $\beta_2 = \epsilon$, entonces parece que $A \to \beta_1$ es el mango, y se debe reducir mediante esta producción. Por

supuesto, dos elementos válidos pueden indicar dos cosas distintas para el mismo prefijo válido. Se pueden resolver algunos de estos conflictos observando el siguiente símbolo de entrada y otros se pueden resolver con los métodos de la siguiente sección, pero no hay que suponer que todos los conflictos de acciones de análisis sintáctico se pueden resolver utilizando el método LR para construir una tabla de análisis sintáctico para una gramática arbitraria.

El conjunto de elementos válidos para cada prefijo viable que pueda aparecer en la pila de un analizador LR es fácil de calcular. De hecho, un teorema básico de la teoría del análisis sintáctico LR es que el conjunto de elementos válidos para un prefijo viable γ es exactamente el conjunto de elementos alcanzados desde el estado inicial a lo largo de un camino etiquetado con γ en el AFD construido a partir de la colección canónica de conjuntos de elementos con transiciones dadas por *ir_a*. En resumen, el conjunto de elementos válidos abarca toda la información útil que se puede extraer de la pila. Aunque aquí no se demostrará, se dará un ejemplo de este teorema.

Ejemplo 4.37. Considérese de nuevo la gramática (4.19), cuyos conjuntos de elementos y función *ir_a* se exhiben en las figuras 4.35 y 4.36. Evidentemente, la cadena $E + T *$ es un prefijo viable de (4.19). El autómata de la figura 4.36 se encontrará en el estado I_7 después de haber leído $E + T *$. El estado I_7 contiene los elementos

$$T \to T * \cdot F$$
$$F \to \cdot(E)$$
$$F \to \cdot\mathbf{id}$$

que son precisamente los elementos válidos para $E + T *$. Para comprobarlo, considérense las tres siguientes derivaciones por la derecha

$$\begin{array}{lll} E' \Rightarrow E & E' \Rightarrow E & E' \Rightarrow E \\ \quad \Rightarrow E + T & \quad \Rightarrow E + T & \quad \Rightarrow E + T \\ \quad \Rightarrow E + T{*}F & \quad \Rightarrow E + T{*}F & \quad \Rightarrow E + T{*}F \\ & \quad \Rightarrow E + T{*}(E) & \quad \Rightarrow E + T{*}\mathbf{id} \end{array}$$

La primera derivación muestra la validez de $T \to T * \cdot F$; la segunda, la validez de $F \to \cdot(E)$, y la tercera, la validez de $F \to \cdot\mathbf{id}$ para el prefijo viable $E + T *$. Se puede demostrar que no hay otros elementos válidos para $E + T *$; se deja la prueba al lector interesado. □

Tablas de análisis sintáctico SLR

A continuación se muestra cómo construir las funciones de acción e *ir_a* del análisis sintáctico SLR a partir del autómata finito determinista que reconoce prefijos viables. Este algoritmo no producirá únicamente tablas de acciones definidas para todas las gramáticas, pero funcionará correctamente con muchas gramáticas para lenguajes de programación. Dada una gramática, G, se aumenta G para producir G',

Formalismen

eine “sprachlose Kunst”

linearmente indipendenti se e soltanto se tutti gli elementi diagonali sono diversi da zero:

$$\begin{pmatrix} a_{11} & 0 & 0 & \cdot & 0 \\ a_{21} & a_{22} & 0 & \cdot & 0 \\ a_{31} & a_{32} & a_{33} & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdot & a_{nn} \end{pmatrix}$$

Dim. Pensiamo se e come si può ottenere la colonna nulla come combinazione lineare delle colonne della matrice.

Supponiamo $a_{kk} \neq 0$ per $k = 1, 2, \dots, n$. Essendo $a_{11} \neq 0$, una combinazione lineare delle nostre colonne la quale produca la n-pla nulla deve avere il primo coefficiente, λ_1, nullo: vedasi la prima riga della matrice. Dovendo poi essere $\lambda_1 a_{21} + \lambda_2 a_{22} = 0$, dal risultato ottenuto $(\lambda_1 = 0)$ e dall'ipotesi fatta $(a_{22} \neq 0)$ segue $\lambda_2 = 0$. E così proseguendo si giunge a constatare l'indipendenza lineare delle nostre n colonne.

Supponiamo poi che, per un certo valore di k, sia $a_{kk} = 0$. Se è $k = n$, fra le nostre n-ple c'è quella nulla, sicché esse sono linearmente dipendenti. Sia $k < n$; proviamo che già le colonne dalla k-ima alla n-ima sono linearmente dipendenti: e invero, in ciascuna di esse ai primi k posti c'è 0. I posti rimanenti sono in numero di $n - k$, e le colonne in questione sono in numero di $n - k + 1$. Dunque, tenuto conto del fatto che non ci possono essere più di $n - k$ $(n - k)$-ple linearmente indipendenti, resta provata la nostra tesi.

Esempio. Applicando il criterio ora stabilito e il risultato del n. 10, vogliamo decidere se le tre terne (che scriviamo in colonna)

$$\begin{array}{c:c:c} 5 & 1 & 3 \\ 1 & 0 & 1 \\ 4 & 7 & 2 \end{array}$$

sono linearmente dipendenti o meno.

Sottraendo dalla II la I moltiplicata per 1/5, e proseguendo con operazioni analoghe otteniamo successivamente

$$\begin{array}{c:c:c} 5 & 0 & 3 \\ 1 & -\frac{1}{5} & 1 \\ 4 & \frac{31}{5} & 2 \end{array} \qquad \begin{array}{c:c:c} 5 & 0 & 0 \\ 1 & -\frac{1}{5} & \frac{2}{5} \\ 4 & \frac{31}{5} & -\frac{2}{5} \end{array} \qquad \begin{array}{c:c:c} 5 & 0 & 0 \\ 1 & -1 & 1 \\ 4 & 31 & -1 \end{array} \qquad \begin{array}{c:c:c} 5 & 0 & 0 \\ 1 & -1 & 0 \\ 4 & 31 & 30 \end{array}.$$

Concludiamo che le tre terne sono linearmente indipendenti.

14. Applicazioni lineari

Def. Se E, E' sono due spazi vettoriali su un medesimo corpo K, diremo che un'applicazione f di E in E' è **lineare** se

$$f(\boldsymbol{u} + \boldsymbol{v}) = f(\boldsymbol{u}) + f(\boldsymbol{v}) \tag{1}$$

$$f(c\boldsymbol{u}) = cf(\boldsymbol{u}) \tag{2}$$

per $\boldsymbol{u}, \boldsymbol{v} \in E$, $c \in K$ arbitrari.

Se la f è lineare, si ha anche

$$f\left(\sum_1^r{}_i\, c_i \boldsymbol{u}_i\right) = \sum_1^r{}_i\, c_i f(\boldsymbol{u}_i)$$

cioè ogni combinazione lineare di vettori di E viene portata nella combinazione lineare analoga, ossia con gli stessi coefficienti, dei vettori corrispondenti di E'.

Si ha necessariamente (dalla (2), per $c = 0$)

$$f(\boldsymbol{0}) = \boldsymbol{0}.$$

Le applicazioni lineari si chiamano anche **omomorfismi.** Sono notevoli i seguenti casi particolari:

f biunivoca fra E ed $f(E)$ ossia iniettiva **(monomorfismo)**;

$f(E) = E'$ ossia suriettiva **(epimorfismo).**

Le applicazioni lineari soddisfacenti alle due prime condizioni, ossia biiettive (biunivoche fra E ed E') sono gli **isomorfismi.**

Se è $E = E'$, l'applicazione è detta **endomorfismo**; un endomorfismo biiettivo, ossia un isomorfismo di uno spazio con se stesso è detto anche **automorfismo**; tale è l'applicazione identica i_E; un altro esempio già visto (n. 1) è la f definita da $f(\boldsymbol{u}) = \lambda \boldsymbol{u}$ con $\lambda \neq 0$. Due endomorfismi di uno spazio vettoriale sono in ogni caso componibili. Nel caso di automorfismi, il composto di due di essi è evidentemente un automorfismo; inoltre, essendo che ogni automorfismo ammette un'applicazione inversa che è ancora un automorfismo, si ha che *gli automorfismi di uno spazio vettoriale costituiscono gruppo.* Gli automorfismi più su menzionati come esempio ne formano un sottogruppo, isomorfo al gruppo moltiplicativo $K - \{0\}$.

Tutte le applicazioni fra spazi vettoriali con le quali avremo a che fare saranno lineari; perciò l'aggettivo potrà venire spesso sottinteso.

Teorema. *Per ogni $f: E \to E'$, l'insieme $f(E)$ è un sottospazio lineare di E' (spazio immagine di E).*

Formalismen

eine “sprachlose Kunst”

Нам осталось объединить накопленные факты и установить следующий результат о борелевском исчислении, который в частном случае — рассмотренный «персонально» для функции $f(z)=z$ — оборачивается классической спектральной теоремой.

7.51. Теорема. *Пусть a — нормальный оператор в H, Ω — компакт в $\mathbb{C}$, содержащий его спектр. Тогда*

(I) («спектральная теорема для нормального оператора») *существует, и притом единственное, разложение единицы E на Ω такое, что $a=\int_\Omega \lambda\, dE_\lambda$;*

(II) *при этом борелевское исчисление на Ω от a может быть задано равенством $f(a)=\int_\Omega f(\lambda)\,dE_\lambda$; $f\in B(\Omega)$.*

◁ Положим $E(\Delta)$ равным значению борелевской функции χ_Δ от a. Тогда из теоремы 7.37 (I) и предложения 7.45 следует, что E — спектральная мера, задающая операторы $f(a)$ с помощью указанного в формулировке равенства; отсюда, в частности, $a=\int_\Omega \lambda\, dE_\lambda$. Далее, из вида борелевского исчисления в 7.37 и из предложени... очевидным образом вытекает, что для любых $x, y\in H$ и борелевского Δ $E_{x,y}(\Delta)=\int_\Omega \chi_\Delta(t)\,d\mu_{x,y}(t)=$ ... $\mu_{x,y}$ — (заведомо регулярная) мера из предложения 7.34; это означает, что E — разложение единицы.

Остается доказать единственность в (I). Пусть $F\colon \Delta\mapsto F(\Delta)$ — разложение единицы на Ω такое, что $a=\int_\Omega \lambda\, dF_\lambda$. Тогда отображение $f\to\int_\Omega f(\lambda)\,dF_\lambda$ является, на основании предложения 7.50, борелевским исчислением на Ω от a. В силу единственности последнего (7.32), это означает, что $\int_\Omega f(\lambda)\,dF_\lambda=\int_\Omega f(\lambda)\,dE_\lambda$ для всех $f\in B(\Omega)$. Взяв любое борелевское Δ и положив $f:=\chi_\Delta$, мы получаем, что $F(\Delta)=E(\Delta)$. ▷

7.52. Задача (ср. 7.45). Пусть Ω — произвольный компакт, φ — непрерывный *-гомоморфизм между полинормированными алгебрами $B(\Omega)$; $\underline{m}$, и $\mathscr{B}(H)$; $\underline{w}$. Показать, что $E\colon \Delta\mapsto\chi_\Delta$ — разложение единицы на Ω.

Разложение единицы, фигурирующее в 7.51, называется *разложением единицы оператора a* (на Ω). Покажем, что оно фактически не зависит от выбора $\Omega\supseteq \mathrm{Sp}\,a$.

7.53. Предложение. *Пусть E^0, соответственно E, — разложение единицы оператора a на $\mathrm{Sp}\,a$, соответственно Ω. Тогда для любого борелевского $\Delta\subseteq\Omega$ $E(\Delta)=E^0(\Delta\cap\mathrm{Sp}\,a)$. Как следствие, $E(\Delta)=0$ при $\Delta\subseteq\Omega\setminus\mathrm{Sp}\,a$.*

◁ Очевидно, $E'\colon \Delta\to E^0(\Delta\cap\mathrm{Sp}\,a)$ есть разложение единицы на Ω такое, что $\int_\Omega \lambda\, dE'_\lambda=\int_\Omega \lambda\, dE^0_\lambda=a$. Поэтому нужное равенство следует из свойства единственности в теореме 7.51 (I). ▷

В заключение настоятельно рекомендуем читателю — если ему не приходилось делать этого раньше — найти в явном виде разложение единицы по крайней мере для следующих операторов: (I) ортопроектора; (II) оператора поточечного умножения на ограниченную последовательность $\{\lambda_n\in\mathbb{C};\ n\in\mathbb{N}\}$ в l_2; (III) оператора $g(t)\mapsto tg(t)$ в $L^2[0,1]$.

7.3. Описание C^*-алгебр как операторных алгебр. Теперь, вооружившись полным знанием коммутативных C^*-алгебр, мы переходим к основным результатам об общих C^*-алгебрах.

7.54. Лемма. *Пусть Ω — компакт, B — унитальная ... банахова алгебра с мультипликативной нормой, $\varkappa\colon C(\Omega)\to B$ — инъективный непрерывный*[1]) *уни... ...физм. Тогда для любой $f\in C(\Omega)$ $\|\varkappa(f)\|\geqslant\|f\|_0$.*

◁ Напомнив о функторе спектра $\underline{\Omega}$ (см. п. 1.2), положим $\alpha:=\underline{\Omega}(\varkappa)\colon \underline{\Omega}(B)\to\Omega$ и $\Delta:=\mathrm{Im}\,\alpha\subseteq\Omega$. Возьмем $t\in\underline{\Omega}(B)$ и $s=\alpha(t)$; тогда для любой $f\in C(\Omega)$ $f(s)=f(\alpha(t))=\widehat{\varkappa(f)}(t)$. Ввиду теоремы 2.11 (I) это означает, что $|f(s)|\leqslant\|\varkappa(f)\|$.

Согласно определению равномерной нормы $\|\cdot\|_0$, нам осталось показать, что $\Delta=\Omega$. Так как α — непрерывное отображение между компактами, Δ замкнуто. Поэтому, если $\Delta\neq\Omega$, то из теоремы Урысона (см. п. 0.0.3°) следует, что в $C(\Omega)$ существуют функции g и h такие, что $g|_\Delta=1$, $h\neq0$ и $gh=0$. Поскольку для любого $t\in\underline{\Omega}(B)$ $\widehat{\varkappa(g)}(t)=g(\alpha(t))=1$, то, согласно следствию 1.2, $\varkappa(g)$ обратим в B. Но тогда из $\varkappa(g)\varkappa(h)=\varkappa(gh)=0$ вытекает, что $\varkappa(h)=0$, а это противоречит инъективности $\varkappa$. ▷

[1]) Предположение о непрерывности $\varkappa$ здесь и в следующей теореме могло бы быть отброшено; см. замечание 1.16.

Formalismen

eine “sprachlose Kunst”

nach Bearbeitung der ersten Spalten von $\left(\begin{smallmatrix} A & B \\ C & D\end{smallmatrix}\right)$ die Matrix

$$\begin{pmatrix} A^+ & A^*B \\ CA^* & D \sqcup CA^*B \end{pmatrix}.$$

Man vergleiche dies mit dem Zwischenzustand

$$\begin{pmatrix} A^{-1} & -A^{-1}B \\ +CA^{-1} & D-CA^{-1}B\end{pmatrix}\begin{pmatrix}U\\Y\end{pmatrix}=\begin{pmatrix}X\\V\end{pmatrix}$$

eines Gauss-Jordan-Algorithmus mit Diagonal-Pivotwahl, angewendet auf das Gleichungssystem

$$\begin{pmatrix}A&B\\C&D\end{pmatrix}\begin{pmatrix}X\\Y\end{pmatrix}=\begin{pmatrix}U\\V\end{pmatrix}.$$

Übungen

3.2.1 Man zeige: Jedes in I enthaltene R ist symmetrisch und jede asymmetrische Relation ist irreflexiv.

3.2.2 Man beweise, daß eine Relation R genau dann Äquivalenz ist, wenn sie reflexiv ist und $RR^{\mathrm{T}} \subset R$ erfüllt.

3.2.3 Man zeige, daß für jede homogene Relation R gilt

$$\inf\{H \mid R \sqcup RH \subset H\} = \inf\{H \mid R \sqcup RH = H\},$$
$$\inf\{H \mid S \sqcup RH \subset H\} = R^*S.$$

3.2.4 Man beweise folgendes: Eine in der Äquivalenzrelation S enthaltene Relation Q und eine beliebige Relation R erfüllen stets $Q(R \sqcap S) = QR \sqcap S$. (Man vergleiche dieses Resultat mit dem modularen Gesetz der Verbandstheorie.)

3.2.5 Für jedes R ist $\overline{R^{\mathrm{T}}\overline{R}} = R \cdot\!\cdot R$ reflexiv und transitiv; siehe (2.3.8).

3.2.6 Man beweise mit dem Schubfachprinzip, daß für jede boolesche $n\times n$-Matrix R

i) $R^n \subset (I \sqcup R)^{n-1}$.

ii) $R^* = \sup_{0\le i<n} R^i$.

iii) $R^+ = \sup_{0<i\le n} R^i$.

iv) $(I \sqcup R)(I \sqcup R^2)(I \sqcup R^4)(I \sqcup R^8)\dots(I \sqcup R^{2^{\lfloor \log n \rfloor}}) = R^*$.

3.3 Extrema, Schranken und Grenzen

Wenn eine Ordnung vorgelegt ist, verlangen typische Aufgabenstellungen, die Ordnung in bezug auf eine Teilmenge der Punkte zu studieren und maximale oder größte Punkte usw. zu ermitteln. Nun betrachten wir aber Ordnungen auf zwei verschiedenen Ebenen, einerseits auf der Meta-Ebene der zumeist umgangssprachlichen Formulierung die Ordnung $R \subset S$ *von* Relationen und andererseits auf der Objekt-Ebene eine Ordnung E *als* Relation. Mit der nun folgenden relationenalgebraischen Untersuchung von Schranken, Grenzen und Extrema klären wir zugleich die Grundlagen unserer bisherigen Arbeitsweise; das ist ein logisch und semantisch interessanter zusätzlicher Aspekt. Diese Überlegungen sind für das Verständnis der weiteren Kapitel nicht unbedingt erforderlich.

Zunächst interessieren wir uns für die „extremalen“ Elemente einer Teilmenge. Das sind die maximalen Elemente, welche keine echten Nachfolger in der Teilmenge haben, wie auch die minimalen Elemente, die keine echten Vorgänger in der Teilmenge besitzen.

3.3.1 Definition. Von einer gegebenen Ordnung betrachten wir die irreflexive Version C, dazu eine Teilmenge t und einen Punkt x. Es heiße

i) x **maximales Element** von t $:\Longleftrightarrow$ $x \subset t \subset \overline{C}^{\mathrm{T}}x$

$\Longleftrightarrow$ $x \subset t$ und $xt^{\mathrm{T}} \subset \overline{C}$

$\Longleftrightarrow$ Der Punkt x gehört zur Menge t und ist nicht kleiner als irgendein Punkt aus t.

$\mathrm{Max}(t) := t \sqcap \overline{Ct}$ Menge der Maxima von t.

ii) ...imales Element von t $:\Longleftrightarrow$ $x \subset t \subset \overline{C}x$

$\Longleftrightarrow$ $x \subset t$ und $tx^{\mathrm{T}} \subset \overline{C}$

$\Longleftrightarrow$ Der Punkt x gehört zur Menge t und ist nicht größer als irgendein Punkt aus t.

$\mathrm{Min}(t) := t \sqcap \overline{C^{\mathrm{T}}t}$ Menge der Minima von t.

Wenn über die Relation C Unklarheit herrschen könnte, werden wir $\mathrm{Max}_C(t)$ anstelle von $\mathrm{Max}(t)$ verwenden. □

Die ersten beiden Varianten der Definition sind nach (2.4.4.i) äquivalent. Ferner ist $x \subset \mathrm{Max}(t) = t \sqcap \overline{Ct}$ äquivalent zu $x \subset t$ und $x \subset \overline{Ct}$. Letzteres ist weiter gleichbedeutend mit $Ct \subset \overline{x}$ und $xt^{\mathrm{T}} \subset \overline{C}$, so daß $\mathrm{Max}(t)$ in der Tat die Zusammenfassung aller maximalen Punkte darstellt.

Maximalität eines Punktes x der Menge t kann als „innere“ Eigenschaft von t, versehen mit der „auf t eingeschränkten“ Ordnungsrelation, verstanden werden, die nachfolgend erklärten Begriffe der Schranken und Grenzen hingegen nicht! Maxima existieren bekanntlich nicht für jede Menge; insbesondere kann die leere Menge niemals maximale Elemente enthalten. Andererseits gibt es für eine Menge u. U. mehrere Maxima. Eine Existenzaussage erwähnen wir als Folge von (6.3.2).

Statt für eine Teilmenge t kann man auch für eine Relation X mit $\mathrm{Max}(X) := X \sqcap \overline{CX}$ „spaltenweise“ Maxima bilden. Viele der folgenden Aussagen bleiben erhalten, und es ergeben sich interessante Identitäten wie ...er anderem Max...

Die Definition der Maxima und Minima verlangt zunächst keine speziellen Eigenschaften der Relation C, insbesondere nicht, daß C eine Striktordnung ist. In Abb. 3.3.1 – die Pfeile sind konventionsgemäß aufwärts gerichtet – liegt eine Relation E zugrunde, die nicht transitiv, also keine Ordnung ist. Auch in

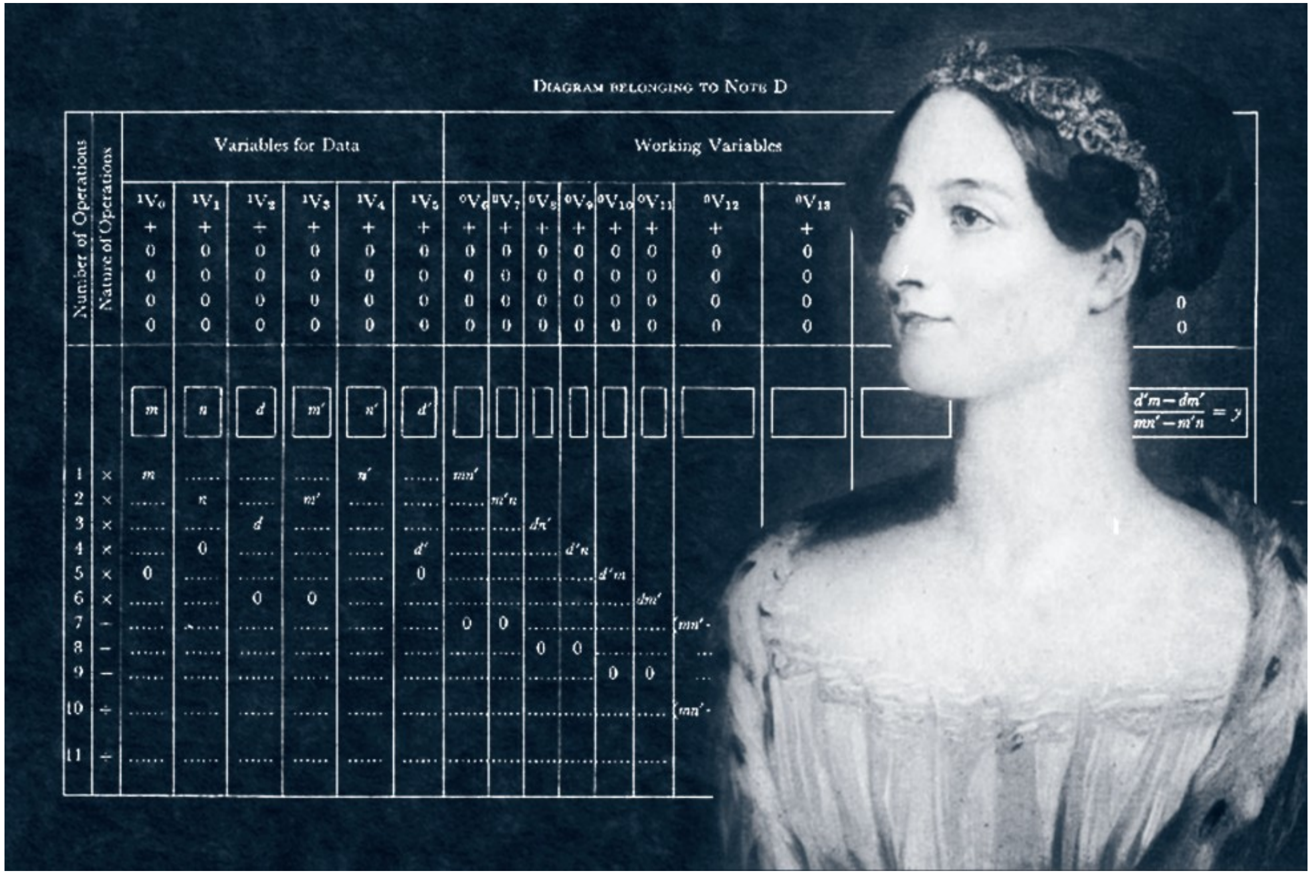
Formalismen—Eine “sprachlose Kunst”

Die *Strukturwissenschaften* Mathematik und Informatik haben über Jahrhunderte eine Formelsprache entwickelt, die wahrhaft völkerverbindend ist.
So finden sich die indisch-arabischen Zahlen in jeder (bedeutenden) lebenden Schriftsprache.

Andere Symbole sind jüngeren Datums, aber haben sich ebenfalls schnell international durchgesetzt. Man kann sie wie Vokabeln erlernen. Oft gibt es gut einprägsame Merkregeln.

Beispiele:
Das Summenzeichen ist das Sigma: $\sum$.
Das Produktzeichen ist das Pi: Π.
Das Differenzieren wird durch ein (evtl. verfremdetes) d bezeichnet.
Das Integrieren hieß früher auch Summieren und wird durch ein langgezogenes S bezeichnet: $\int$.

Ada Lovelace, die erste Programmiererin



Programmtexte als Formalismen—ein Weg zur Informatik

```
procedure BS(A)
//Die Eingabe A ist eine Liste zu sortierender Gegenstände
  for each i from 1 to length(A) do:
      for each j from length(A) downto i + 1 do:
        if A[ j ] < A[ j-1 ] then
          swap( A[ j ], A[ j-1 ] )
        end if
      end for
  end for
end procedure
```

Programmtexte als Formalismen—ein Weg zur Informatik

```
procedure BS(A)
  for each i from 1 to length(A) do:
      for each j from length(A) downto i + 1 do:
        if A[ j ] < A[ j-1 ] then
          swap( A[ j ], A[ j-1 ] )
        end if
      end for
  end for
end procedure
```

Ein Beispiel:

7 5 3 8
7 5 3 8
7 3 5 8
3 7 5 8
3 7 5 8
3 5 7 8

Ist Sortieren wichtig ?

Ist die vorgeschlagene Prozedur richtig ?

Wie sieht man das ein ?

Modellieren

In der Praxis liegen die Aufgaben nicht “fertig formalisiert” vor.
Zentrale Aufgabe von “Strukturwissenschaftlern”:
Entdecke und beschreibe die wesentlichen Aspekte einer Aufgabe.
Nur nach erfolgreicher *Modellierung* kann darauf aufbauend eine geeignete, passende Formalisierung erfolgen.
Von einer geeigneten Formalisierung zu einem guten Programm sollte es nur noch ein kleiner Schritt sein...

Im Folgenden einige ausführlichere Beispiele. . .

Missverständnisse...

Obwohl doch alles klar ist ?!

Wieviele Seiten hat ein quaderförmiger Gegenstand?

Selbstverständlich SECHS!

Meinen wir das immer im Alltag??

Das hat Perscheid mal karikiert ...

WICHTIG: Genaues Verständnis des Zusammenhangs!

Manchmal hat ein Quader nur zwei Seiten . . .

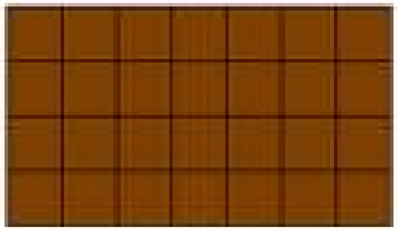
Algorithmen im Alltag?

aus der Badischen Zeitung vom 26. August 1963:

Wenn die Hausfrau wissen möchte, an welchem Tage die Butter ausgeformt wurde, so muss sie das Stanniolpapier zurückpellen, bis sie drei eingestanzte Buchstaben entdeckt. Schreibt man dazu das Schlüsselwort “Milchprobe” auf ein Blatt Papier und setzt darunter die Zahlen 1 bis 10, so ergibt sich aus den Buchstaben MBP die Zahl 196. Damit ist der Tag errechnet, an dem die Butter ausgeformt wurde, nämlich der 196. Tag des Jahres. Das wäre also an Hand des Kalenders der 15. Juli.

Wo steckt hierin der Algorithmus? Ist alles klar und eindeutig beschrieben?
(Leider nein, die Erklärung ist in gewissem Sinne sogar falsch.)
Jedenfalls ist die Milchprobe ein Politikum gewesen.

Chomp: Das Feld links unten auf der $n \times m$-Tafel ist vergiftet !



"Zugteile":  

Die zwei Spieler wählen abwechselnd ein Schokoladenstück und essen nicht nur das gewählte Stück, sondern alles, was sich rechts und oberhalb davon befindet.
Der letzte Biss legt somit die gesamte verbleibende Schokolade fest.
Der Spieler, der das vergiftete Stückchen essen muss(!), hat verloren.

Chomp: Das Feld links unten auf der $n \times m$-Tafel ist vergiftet !
Aber manche schert das nicht !

Chomp: Das Feld links unten auf der $n \times m$-Tafel ist vergiftet !

Chomp: Das Feld links unten auf der $n \times m$-Tafel ist vergiftet !

Satz: Spieler A hat eine Gewinnstrategie, falls $n = m$.
Das bedeutet: A kann unabhängig von B's Zügen stets gewinnen.

Beweis: konstruktiv (!) (D.h.: Wir erklären jetzt A, wie er gewinnen kann.)
Im ersten Zug isst A alles bis auf die erste Zeile und die erste Spalte.
Isst nun B k Stückchen von der Zeile, isst A k Stückchen von der Spalte und umgekehrt.
Zum Schluss bleibt das Eck links unten für B übrig. $\rightsquigarrow$ A gewinnt.

Ebenso konstruktiv: Satz: Spieler A hat eine Gewinnstrategie, falls $n = 2$.
Wie geht's ?

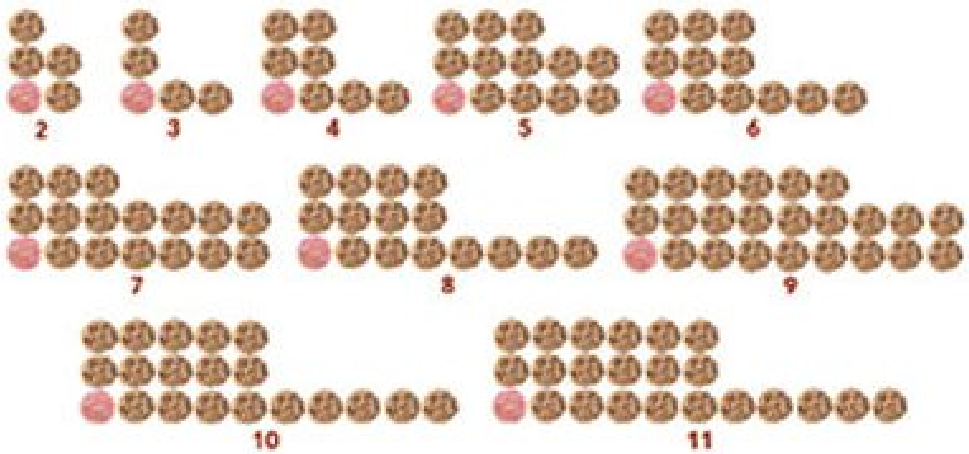
Chomp: Das Feld links unten auf der $n \times m$-Tafel ist vergiftet !

Satz: Spieler A hat eine Gewinnstrategie.

Beweis: Eingangs gibt es $n \times m$ Schokoladenstücke, und bei jedem Biss verringert sich die Zahl der Schokoladenstückchen um wenigstens Eins.
$\rightsquigarrow$ Das Spiel endet stets mit Spieler A oder B als Gewinner.
Annahme: A hat keine Gewinnstrategie.
$\rightsquigarrow$ B hat eine Gewinnstrategie. (Einer-Wird-Gewinnen)
Auf einen beliebigen Zug α hat also B eine richtige Antwort β, um stets zu gewinnen.
Dies gilt insbesondere am Anfang für den ersten Zug α von A, bei dem er nur ein einziges Stück abbeißen könnte. Aufgrund der Struktur des Spiels hätte aber auch A den Antwort-Biss β bereits ausführen können und hätte so eine Gewinnstrategie. Widerspruch !

Masterarbeit (oder mehr. . .): Finde einen konstruktiven Beweis !

Chomp: Das Feld links unten auf der $3 \times m$-Tafel ist vergiftet !



Exkurs — Spieltheorie

Hier: Zweipersonenspiele mit vollständiger Information

Wichtig: Spielbeschreibung mit Regelwerk (kann man logisch spezifizieren); wann gilt das Spiel für wen als beendet / gewonnen ?

Fragestellungen:
Wie sollten Spieler A (macht ersten Zug) oder Spieler B ziehen ?
Gesucht: Vorschrift (*Strategie*), die jeder *Spielkonfiguration* einen Zug zuordnet.
Besitzt Spieler A (oder Spieler B) eine Gewinnstrategie ?

Exkurs — Spieltheorie

Strategie, lokale Sicht: B sucht “Antwort” auf Zug von A und umgekehrt.

Wie gelangt man zu einer guten Strategie ? *Minimax-Ansatz*

A denkt darüber nach, einen Zug α zu machen.
Dazu überlegt sich A, was B auf α antworten könnte.
Um zu gewinnen, muss A für jede Antwort β von B einen “Gewinnzug” α' usw.
Etwas formaler (c Anfangskonfiguration, C_A Menge der Endkonfigurationen, in denen A gewonnen hat): A besitzt *Gewinnstrategie* gdw.

$$\exists\alpha\forall\beta\exists\alpha'\forall\beta'\ldots(c\alpha\beta\alpha'\beta'\ldots)\in C_A$$

Exkurs — Spieltheorie

Strategie, lokale Sicht: B sucht “Antwort” auf Zug von A und umgekehrt.

Wie gelangt man zu einer guten Strategie ?

Entsprechend besitzt B eine Gewinnstrategie gdw.

$$\forall\alpha\exists\beta\forall\alpha'\exists\beta'\ldots(c\alpha\beta\alpha'\beta'\ldots)\in C_B$$

Satz: Gibt es in Zweipersonenspielen mit vollständiger Information kein Unentschieden, so hat entweder Spieler A oder Spieler B eine Gewinnstrategie. (Einer-Wird-Gewinnen)

Beweis: . . . de Morgan. . .

Exkurs — Spieltheorie: ein Beispiel

Satz: Spieler A hat eine Gewinnstrategie für “Vier gewinnt”.

<u>Beweis:</u> ist konstruktiv, siehe folgende Diplomarbeit von 1988

Exkurs — Spieltheorie ein weiteres Beispiel

 Rekonstruktion eines antiken Mühle-Spiels

Satz: Es gibt weder für A noch für B eine Gewinnstrategie bei “Mühle”.

Beweis: besteht aus einer 17 GB großen Datei . . .

Ebenfalls Ergebnis einer Diplomarbeit (von 1993)

Exkurs — Spieltheorie (ernsthaftere) Motivation

Schwerpunkt Spieleprogrammierung bei Prof. Sturm

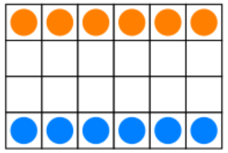
Für Wirtschaftsinformatiker insbesondere: Historischer Ausgangspunkt der Spieltheorie ist die Analyse von Gesellschaftsspielen durch Joh(an)n von Neumann (ungarisch: Neumann János) im Jahre 1928.
Anwendbarkeit des von ihm entwickelten Ansatzes zur Analyse wirtschaftlicher Fragestellungen $\rightsquigarrow$
"Spieltheorie und wirtschaftliches Verhalten" (Theory of Games and Economic Behavior; JvN und Oskar Morgenstern, 1944):
Startpunkt der modernen Spieltheorie

Latrunculi: Unser Spielprojekt bei Campus Illuminale

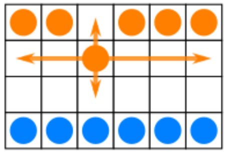
Latrunculi

Regeln

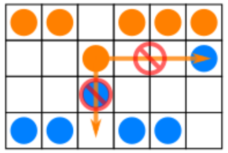
Henning Fernau & Moritz Gobbert
fernau@uni-trier.de gobbert@uni-trier.de



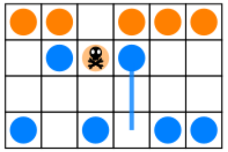
Zu Beginn des Spiels platziert jeder Spieler auf der ihm nächsten Reihe seine Spielsteine. (Alternativ: Die Spieler setzen abwechselnd ihre Steine auf beliebige Felder)



Die Spieler ziehen abwechselnd je einen ihrer Spielsteine horizontal oder vertikal, so weit sie wollen. (Vgl. Schach-Turm)



Hierbei dürfen sie weder andere Spielsteine überspringen, noch auf ein Feld ziehen, auf dem bereits ein Spielstein liegt.



Ein Spieler kann einen Spielstein des Gegners *erobern*, wenn er so zieht, dass der gegnerische Spielstein auf zwei gegenüberliegenden Feldern von eigenen Steinen umstellt ist.

„Spiele vorsichtig auch und schlau den Krieg der Steine, wo ein einzelner Stein durch zweifachen Feind verloren geht."
— *Ovid*

Geritztes Latrunculi-Spielfeld auf einer römischen Tegula
Quelle: GDKE - Landesmuseum Mainz (Ursula Rudischer)

Induktiv definierte Folgen

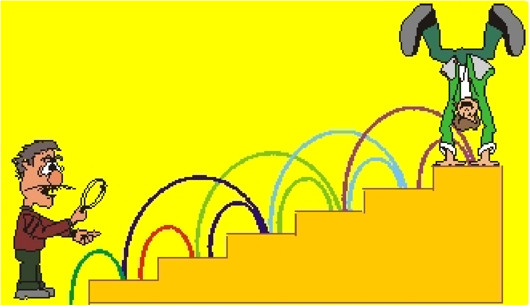
Beispiel: Die Folge $(a_0, a_1, a_2, \ldots) = (a_n)_{n \in \mathbb{N}}$ ist gegeben durch:

$$\begin{aligned} a_0 &= 1, \\ a_n &= a_{n-1} \cdot n, \text{ für } n > 0. \end{aligned}$$

Schreibweise: $\prod_{j \in [n]} b_j$ bezeichnet das Produkt aller Zahlen der endlichen Folge $(b_0, \ldots, b_{n-1})$. Das leere Produkt wird als Eins interpretiert.

Satz: Für die oben definierte Folge a_n gilt: $a_n = \prod_{j \in [n]} (j+1)$.
$f(n) = a_n$ heißt auch *Fakultätsfunktion*; Schreibweise: $n!$

Induktiv definierte Folgen: Treppensteigen



Bei jeder Stufe kann man sich die Frage stellen:
Nehme ich eine Stufe oder überspringe ich eine Stufe?
Die erste Stufe muss auf jeden Fall betreten werden.
Frage: Auf wieviel verschiedene Arten f_n kann man nun eine n-stufige Treppe heraufgehen?

Versuchen wir (an der Tafel), eine Tabelle dafür aufzustellen.

Induktiv definierte Folgen: Treppensteigen

Finden wir ein *Bildungsgesetz* ?

Für $n \geq 2$ gibt es zwei Möglichkeiten, eine n-stufige Treppe zu erklimmen:
– entweder hatten wir einen Schritt von einer $(n-1)$-stufigen Treppe aus gemacht
– oder zwei Stufen auf einmal von einer $(n-2)$-stufigen Treppe aus genommen.

$\rightsquigarrow f_n = f_{n-1} + f_{n-2}$; Sonderfälle: $f(0) = 0$ und $f(1) = 1$.

Diese Folge kommt sehr häufig in der Natur und Kultur vor und wird gemeinhin die Folge der *Fibonacci-Zahlen* genannt !
Mehr über Leonardo Fibonacci bei einem virtuellen Museumsbesuch.

Der Goldene Schnitt

ist die Teilung einer Strecke so, dass die gesamte Strecke X sich zu dem größerem Teilstück der Länge 1 verhält wie das größere Teilstück zum kleineren.

Das Teilverhältnis lässt sich nun einfach ausrechnen. Es gilt:

$$X : 1 = 1 : (X-1), \quad \text{also:} \quad X^2 - X = 1$$

mit den beiden Lösungen ϕ und $\hat{\phi} = 1 - \phi$, wobei

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.6181\ldots$$

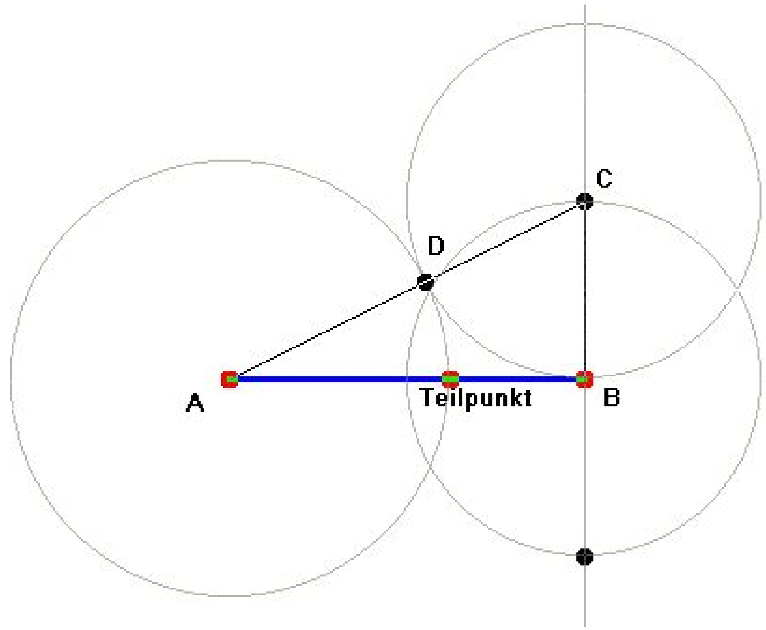
die *goldene Schnittzahl* ist.

Satz: (Formel von Binet) $f_n = \frac{\phi^n - (1-\phi)^n}{\sqrt{5}}$. Daher: $\lim_{n\to\infty} \frac{f_{n+1}}{f_n} = \phi$.

Ein architektonischer Exkurs

Beim Parthenontempel in Athen bildet der Säuleneingang hierbei ein goldenes Rechteck, also ein Rechteck, dessen Seiten sich genau wie der goldene Schnitt verhalten. Auch verhält sich die Höhe bis zum Dach zur Höhe der Säulen wie der goldene Schnitt.

Konstruktion des Goldenen Schnitts



Im Endpunkt der Strecke AB wird die Senkrechte errichtet. Auf ihr trägt man die Hälfte von AB ab. Es ergibt sich Punkt C. Der Kreis um C mit dem Radius CB schneidet AC bei D. Überträgt man den Abstand AD auf die Strecke AB, so ergibt sich der Teilpunkt T. T teilt AB im Verhältnis des Goldenen Schnittes.

Fibonacci-Zahlen: Eine Philatelistische Annäherung

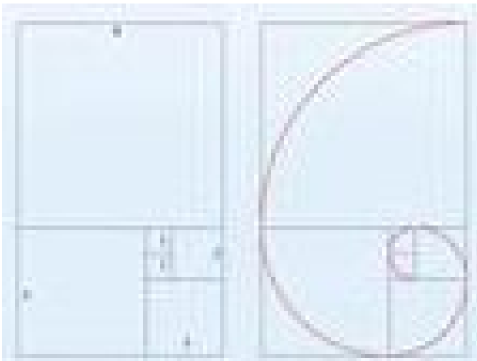
Diese Schweizer Briefmarke wurde zum 150-jährigen Bestehen des Schweizerischen Ingenieur- und Architektenvereins SIA herausgegeben und enthält eine interessante mathematische Konstruktion.

Die Briefmarke zeigt den Zusammenhang des Goldenen Schnitts mit der Logarithmischen Spirale, auch Fibonacci-Spirale genannnt.

Nähere Erläuterungen finden Sie hier.

Ein **geometrisches Problem**, das zu den Fibonacci-Zahlen führt:
Konstruktion aneinanderliegender Quadrate.

Wird in jedem Quadrat ein Viertel eines Kreises gezogen wie in der Abbildung, erhält man die sogenannte *Fibonacci-Spirale*, eine Form, die bei gewissen Muscheln beobachtet werden kann.

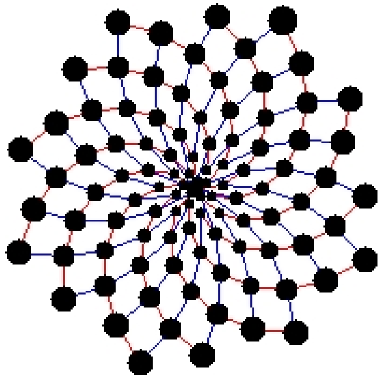


Die Muscheln sind lediglich ein Beispiel für ein verbreitetes Phänomen: das **Vorkommen der Fibonacci-Zahlen in der Natur**.

Die Fibonacci-Zahlen finden sich in der Position der Blätter und der Blumenblätter von Blumen, in den Verzweigungen einiger Pflanzen, in der Anordnung der Samen der Sonnenblumen oder der Schuppen der Tannzapfen.
Letztere sind so angeordnet, dass sie zwei Serien von entgegengesetzten Spiralen bilden, die im Zentrum zusammenfließen.
Im selben Tannzapfen oder derselben Sonnenblume sind die Zahlen der Spiralen, die sich in beide Richtungen winden, aufeinanderfolgende Fibonacci-Zahlen.

Warum sind Fibonacci-Zahlen "natürlich" ? Gute Erklärungen mit vielen geschichtlichen Erläuterungen (auch für Lateiner) finden Sie hier.

Konkretes Naturbeispiel: Die Sonnenblume

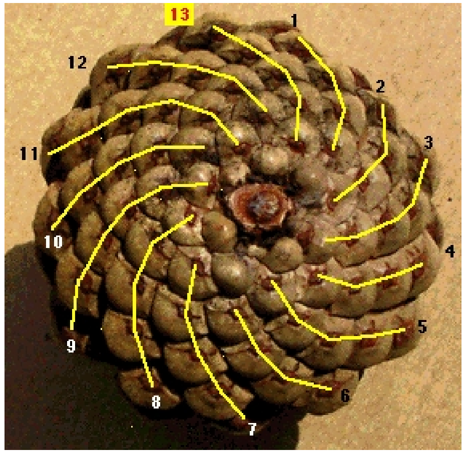


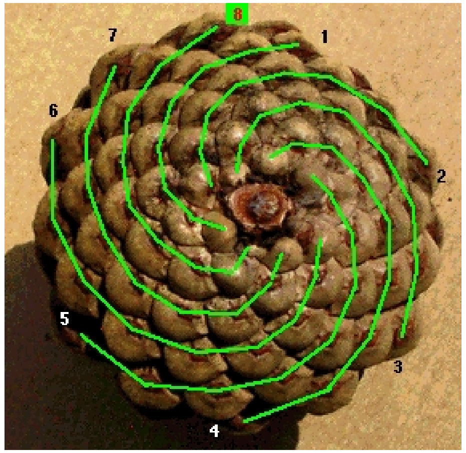
Bei der Sonnenblume sind die Samen bogenförmig angeordnet. Das heißt, wenn man die Anzahl der Bögen gegen den Uhrzeigersinn und die der Bögen im Uhrzeigersinn betrachtet, erhält man zwei aufeinanderfolgende Fibonacci-Zahlen.

Begründung: Die Sonnenblumenkerne wachsen kreisförmig um den Mittelpunkt der Sonnenblume. Zwei in ihrer Entwicklung aufeinander folgende Kerne teilen den Umfang dabei im Verhältnis des Goldenen Schnitts. Der Winkel zwischen ihnen beträgt also $360^o - 360^o/\phi \approx 137,518\ldots^o$.

Mehr finden Sie bei matheprisma.

Konkretes Naturbeispiel: Ein Tannenzapfen





Mehr Infos zu Fibonacci hier.

Gehen Sie mit wachen Sinnen durch die Natur !

Konkretes Modellieren

ist Ihnen aus der Schulmathematik vielleicht leidvoll bekannt:

Textaufgaben

Diese sind aber für die angewandte Mathematik und Informatik unverzichtbar.

Gute Informatiker brauchen neben gut ausgebauten mathematischen auch **sehr gute sprachliche Fähigkeiten**.
Insbesondere müssen **Kundenwünsche** richtig gedeutet, verstanden und umgesetzt werden können.

Textaufgaben

Versuchen Sie stets:
evtl. (sprachliche) Unklarheiten zu erkennen und zu benennen,
sich klar darüber zu werden, was wirklich bekannt ist,
möglichst eine Skizze von der Lage anzufertigen,
bei einer “zu abstrakten” Frage zunächst nach konkreten Beispielen zu suchen,
sich klar darüber zu werden, was wirklich gesucht ist,
die Aufgabenstellung und den Lösungsweg möglichst klar zu gliedern,
sich bewusst machen, welche Lösungsstrategien man anwenden kann,
auf bekannte oder ähnliche Aufgaben zurückzugehen (Erfahrungen !) und
zu klären, wie und wo man sich evtl. noch fehlenden Informationen besorgen kann.

Textaufgaben

1. Eine Ziege ist mit Hilfe eines sechs Meter langen Seils an der Ecke eines Stalls mit fünf Meter Länge und vier Meter Breite angebunden; sie befindet sich im Freien. Der Stall ist von einer Grasfläche umgeben. Was für eine Fläche kann die Ziege abweiden ?

2. Unter einem *Palindrom* versteht man eine Zahl (oder allgemeiner eine Buchstabenreihe), die vorwärts wie rückwärts gelesen denselben Wert hat (dasselbe Wort liefert). Ein Beispiel ist 12321 oder das Wort “Reliefpfeiler”. Ein Freund behauptet, alle Palindrome mit vier Ziffern seien durch 11 teilbar. Stimmt das ?

3. Die Durchquerung einer Wüste nimmt neun Tage in Anspruch. Ein Mann muss eine Botschaft auf die andere Seite bringen, auf der seine Vorräte nicht aufgefrischt werden können, und dann wieder zurückgehen. Ein einzelner Mann kann Nahrungsmittel für 12 Tage tragen. Allerdings können unterwegs Depots angelegt werden. Wie schnell kann die Botschaft an die andere Seite gelangen ? Wie schnell geht es, wenn zwei Boten zur Verfügung stehen ?

4. Auf einem Tisch liegen zwei Streichholzstapel. Zwei Spieler entfernen abwechselnd Streichhölzer nach der folgenden Regel: Ein Spieler kann entweder von beiden Stapeln je ein Streichholz nehmen oder ein einzelnes Streichholz entfernen. Der Spieler, der das letzte Streichholz wegnehmen muss, hat verloren. Gibt es eine Gewinnstrategie für den Spieler, der mit dem Spiel beginnt ? Oder gibt es eine Gewinnstrategie für den Nachziehenden ? Oder gibt es für keinen Spieler eine Gewinnstrategie ? Und wie sähe so eine Gewinnstrategie aus ?

   Betrachten Sie (zunächst) “kleine Streichholzstapel”.
   Die Aufgabe ist schwieriger, als sie zuerst scheinen mag.

   Daher vereinfachte Annahme: Beide Stapel enthalten zu Spielbeginn gleich viele Hölzer.