

Formale Grundlagen der Informatik

SoSe 2008 in Trier

Henning Fernau

Universität Trier

fernau@uni-trier.de

Formale Grundlagen der Informatik

Gesamtübersicht

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

Organisatorisches

Vorlesung Montag 8.15 bis 11.30 im HS 11 (mit Pause)

an den folgenden Tagen: 8.45 bis 12.00 (mit Pause)

12-14 Uhr: Gelegenheit zur Gruppenarbeit zu ausgewählten Übungen

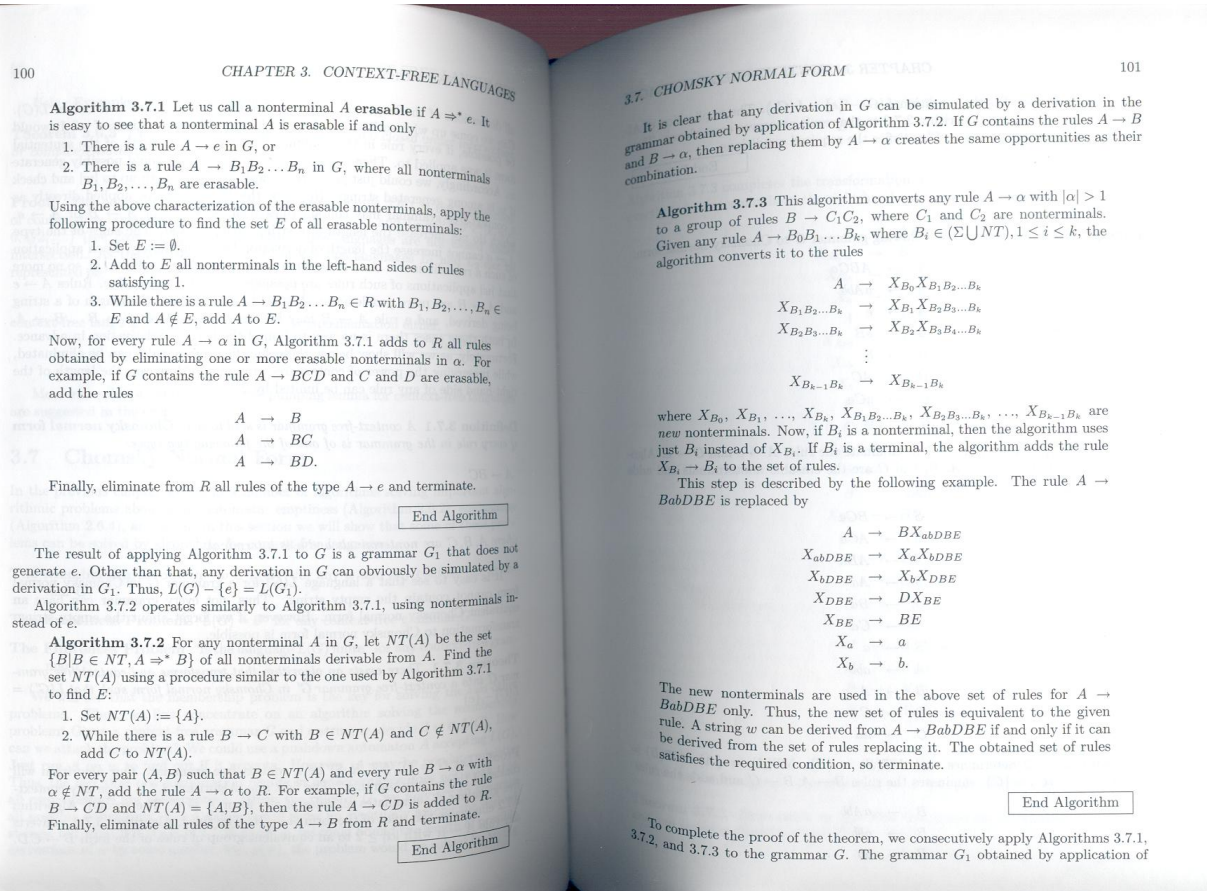
parallel dazu: 13.30-14.15 **Sprechstunde**: Wir fünf sind im 4.OG für Sie da.

(Große) Übung 14-16 Uhr, d.h.: 14.15-15.45 (Gulan, Raible, Bakman, Schlecht)
im H 406

Am Abend überdenken Sie bitte den Stoff; sie erhalten auch Gelegenheit zu weiteren Übungen.

An den folgenden Tagen stehen Ihnen Studierende (Bakman, Schlecht) von 8.00 bis 8.45 im H 406 für Fragen hier zur Verfügung. Dann werden auch die Übungen vom letzten Abend angesprochen. **(Kleine) Übung; Sprechstunde**
(Weitere Unterrichtsformen: praktische Übungen, Seminare)

Formalismen — Eine “sprachlose Kunst”



Formalismen — Eine “sprachlose Kunst”

no determina N y los estados son los propios elementos. Hay una transición de $a \rightarrow \alpha\beta$ a $a \rightarrow \alpha\chi\beta$ etiquetada con X , y hay una transición de $A \rightarrow \alpha\beta$ a $B \rightarrow \gamma$ 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, $\langle \text{cl}_\epsilon(I), X \rangle$ 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.

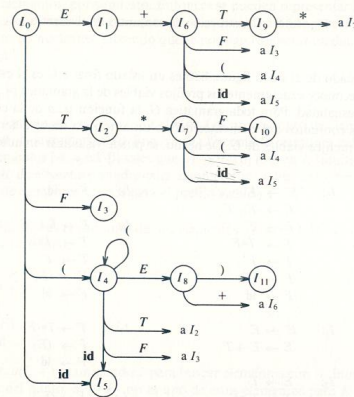


Fig. 4.36. Diagrama de transiciones del AFD D para los prefijos viables.

Elementos válidos. Se dice que el elemento $A \rightarrow \beta_1, \beta_2$ es *válido* para un prefijo viable α_1 si existe una derivación $S' \xRightarrow{*} \alpha_1 \omega \xRightarrow{*} \alpha_1 \beta_1, \beta_2$. En general, un elemento será válido para muchos prefijos viables. El hecho de que $A \rightarrow \beta_1, \beta_2$ sea válido para α_1 informa sobre si desplazar o reducir cuando se encuentre α_1 en la pila de análisis sintáctico. En concreto, si $\beta_1 \neq \epsilon$, 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 $A \rightarrow \beta_1$ 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 q_0 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 lr_{-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

$$\begin{array}{l} T \rightarrow T * \cdot F \\ F \rightarrow \cdot (E) \\ F \rightarrow \cdot \text{id} \end{array}$$

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

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

La primera derivación muestra la validez de $T \rightarrow T * \cdot F$; la segunda, la validez de $F \rightarrow \cdot (E)$, y la tercera, la validez de $F \rightarrow \text{id}$ 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. $\square$

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' ,

Formalismi — Eine “sprachlose Kunst”

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

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

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)

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

sono linearmente dipendenti o meno.

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

$$\begin{array}{c} 5 \\ 1 \\ 4 \end{array} \begin{array}{c} 0 \\ -\frac{1}{5} \\ \frac{31}{5} \end{array} \begin{array}{c} 3 \\ 1 \\ 2 \end{array} \quad \begin{array}{c} 5 \\ 1 \\ 4 \end{array} \begin{array}{c} 0 \\ -\frac{1}{5} \\ \frac{31}{5} \end{array} \begin{array}{c} 0 \\ \frac{2}{5} \\ -\frac{2}{5} \end{array} \quad \begin{array}{c} 5 \\ 1 \\ 4 \end{array} \begin{array}{c} 0 \\ -1 \\ 31 \end{array} \begin{array}{c} 0 \\ 1 \\ -1 \end{array} \quad \begin{array}{c} 5 \\ 1 \\ 4 \end{array} \begin{array}{c} 0 \\ -1 \\ 31 \end{array} \begin{array}{c} 0 \\ 0 \\ 30 \end{array}$$

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

$$f(u + v) = f(u) + f(v) \quad (1)$$

$$f(cu) = cf(u) \quad (2)$$

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

Se la f è lineare, si ha anche

$$f\left(\sum_1^r c_i u_i\right) = \sum_1^r c_i f(u_i)$$

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 $c = 0$)

$$f(0) = 0.$$

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**).

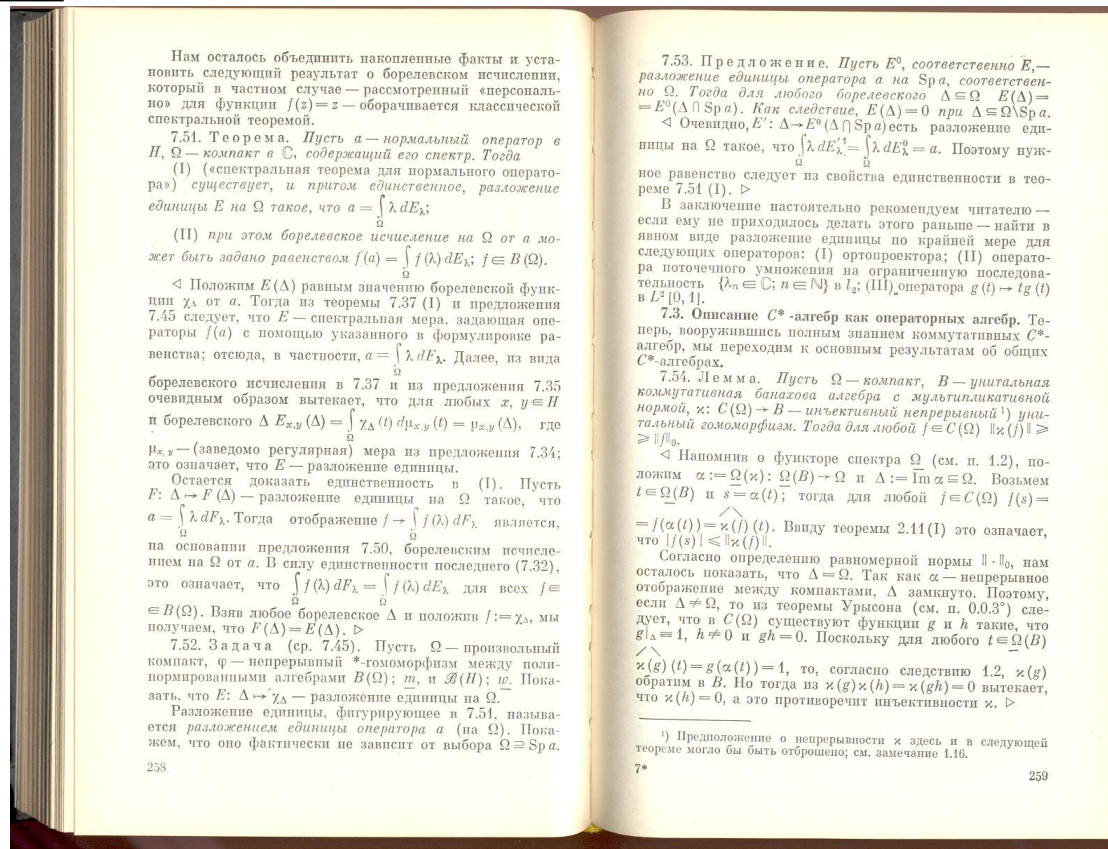
Le applicazioni lineari soddisfacenti alle due prime condizioni, ossia biietive (biunivoche fra E ed E') sono gli **isomorfismi**.

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 $f(u) = \lambda u$ 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 $f: E \rightarrow E'$, l'insieme $f(E)$ è un sottospazio lineare di E' (spazio immagine di E).

Formalismen — Eine “sprachlose Kunst”



Formalismen — Eine “sprachlose Kunst”

42

3. Transitivität

nach Bearbeitung der ersten Spalten von $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ die Matrix

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

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 $RR^T \subset R$ erfüllt.

3.2.3 Man zeige, daß für jede homogene Relation R gilt
 $\inf\{H \mid R \cup RH \subset H\} = \inf\{H \mid R \cup RH = H\};$
 $\inf\{H \mid S \cup RH \subset H\} = R^*S.$

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

3.2.5 Für jedes R ist $\overline{R^T R} = R \cdot R$ 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) $R^n \subset (I \cup R)^{n-1}$.
- ii) $R^* = \sup_{0 \leq i < n} R^i$.
- iii) $R^+ = \sup_{0 < i \leq n} R^i$.
- iv) $(I \cup R)(I \cup R^2)(I \cup R^4)(I \cup R^8) \dots (I \cup R^{2^{i-1}}) = R^*$.

3.3 Extrema, Schranken und Grenzen

Wenn eine Ordnung vorgelegt ist, verlangen typische Aufgabenstellungen, diese 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 ungangssprachlichen Formulierung die Ordnung $R \subset S$ von Relationen und an-

3.3 Extrema, Schranken und Grenzen

43

dererseits auf der Objekt-Ebene eine Ordnung E als Relation. Mit der nun folgenden relationalalgebraischen 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 $t : \Leftrightarrow x \in t \subset \overline{C^T x}$
 $\Leftrightarrow x \in t$ und $xt^T \subset \overline{C}$
 $\Leftrightarrow$ Der Punkt x gehört zur Menge t und ist nicht kleiner als irgendein Punkt aus t .

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

- ii) x minimales Element von $t : \Leftrightarrow x \in t \subset \overline{Cx}$
 $\Leftrightarrow x \in t$ und $tx^T \subset \overline{C}$
 $\Leftrightarrow$ Der Punkt x gehört zur Menge t und ist nicht größer als irgendein Punkt aus t .

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

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

Die ersten beiden Varianten der Definition sind nach (2.4.4.i) äquivalent. Ferner ist $x \in \text{Max}(t) = t \cap \overline{Ct}$ äquivalent zu $x \in t$ und $x \subset \overline{Ct}$. Letzteres ist weiter gleichbedeutend mit $Ct \subset \overline{x}$ und $xt^T \subset \overline{C}$, so daß $\text{Max}(t)$ 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 maximale Elemente enthalten. Andererseits gibt es für eine Menge u 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 $\text{Max}(X) := X \cap \overline{CX}$ „spaltenweise“ Maxima bilden. Viele der folgenden Aussagen bleiben erhalten, und es ergeben sich interessante Identitäten wie unter anderem $\text{Max}(E) = I$.

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 jüngerer Datums, aber haben sich ebenfalls schnell international durchgesetzt. Man kann sie wie Vokabeln erlernen. Oft gibt es gut einprägsame Merkgeregeln.

Beispiele:

Das Summenzeichen ist das Sigma: $\sum$.

Das Produktzeichen ist das Pi: $\prod$.

Das Differenzieren wird durch ein (evtl. verfremdetes) d bezeichnet.

Das Integrieren hieß früher auch Summieren und wird durch ein langgezogenes S bezeichnet: $\int$.

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”:

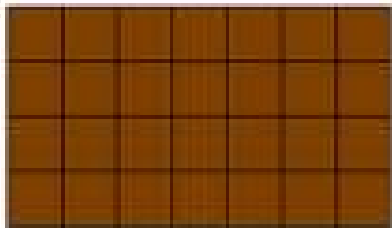
Entdecke und beschreibe die wesentlichen Aspekte einer Aufgabe.

Nur nach erfolgreicher *Modellierung* kann eine geeignete Formalisierung erfolgen.

Von einer geeigneten Formalisierung zu einem guten Programm sollte es nur noch ein kleiner Schritt sein...

Im Folgenden einige ausführlichere Beispiele...

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 $n = m$.

Beweis: konstruktiv (!)

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. $\leadsto$ 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.

$\leadsto$ Das Spiel endet stets mit Spieler A oder B als Gewinner.

Annahme: A hat keine Gewinnstrategie.

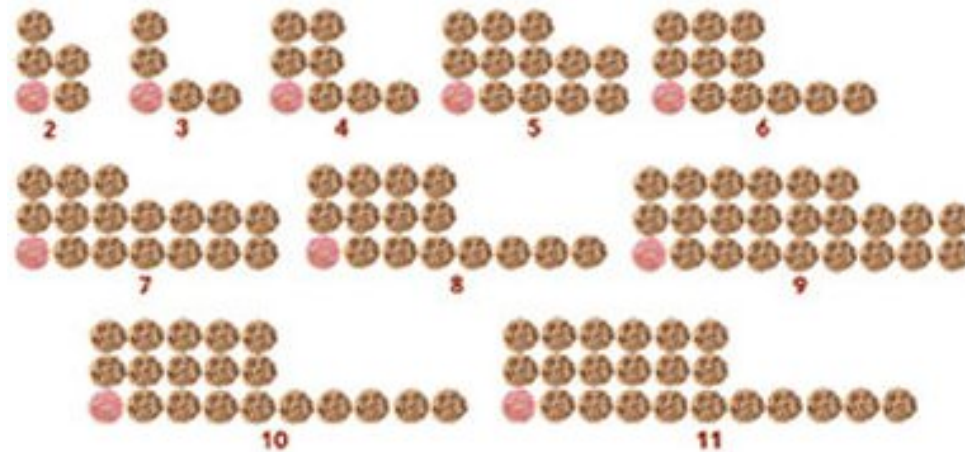
$\leadsto$ B hat eine Gewinnstrategie. (Einer-Wird-Gewinnen)

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 !**

Diplomarbeit (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.

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

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.

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

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”.

Beweis: 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 (ernsthafte) 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 $\leadsto$

“Spieltheorie und wirtschaftliches Verhalten” (Theory of Games and Economic Behavior; JvN und Oskar Morgenstern, 1944):

Startpunkt der modernen Spieltheorie

Rekursiv definierte Folgen

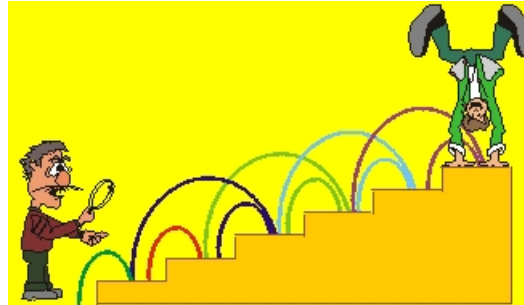
Beispiel: Die Folge $(a_0, a_1, a_2, \dots) = (a_n)_{n \in \mathbb{N}}$ ist gegeben durch:

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

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

Satz: Für die oben definierte Folge a_n gilt: $a_n = \prod_{j \in [n] \setminus \{0\}} j$. $f(n) = a_n$ heißt auch *Fakultätsfunktion*; Schreibweise: $n!$

Rekursiv 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.

Rekursiv 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.

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

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 größtem Teilstück der Länge 1 verhält wie das größere Teilstück zum kleineren.

Das Teilverhältnis lässt sich nun einfach ausrechnen. Es gilt:

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

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

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

die *goldene Schnittzahl* ist.

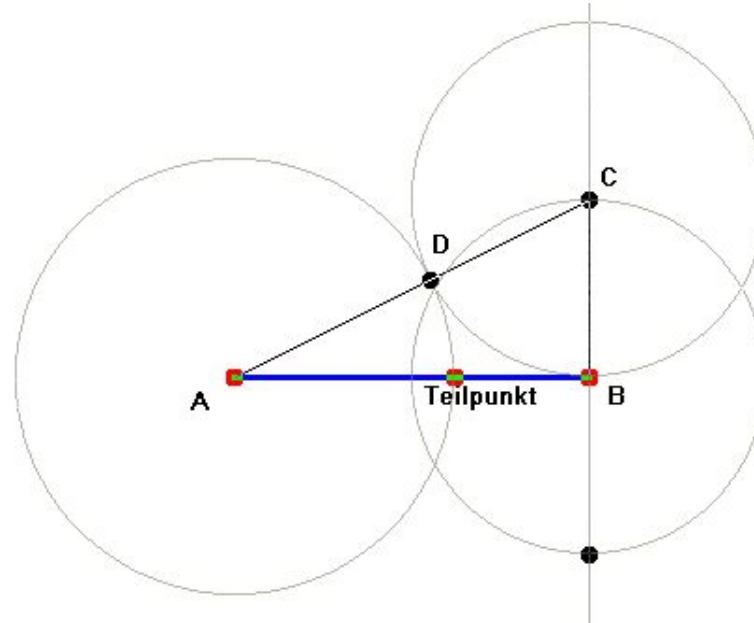
Satz: (Formel von Binet) $f_n = \frac{\phi^n - (1-\phi)^n}{\sqrt{5}}$. Daher: $\lim_{n \rightarrow \infty} \frac{f_{n+1}}{f_n} = \phi$.

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



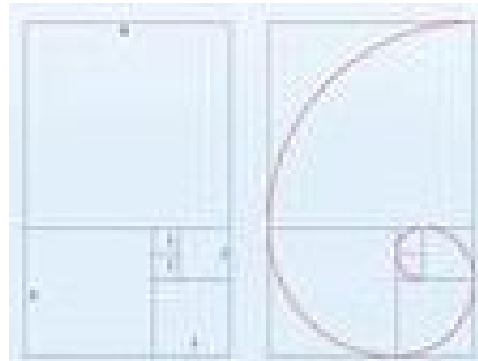
Diese Schweizer Briefmarke wurde zum 150-jährigen Bestehen des Schweizerischen Ingenieur- und Architektenvereins SIA herausgegeben und enthält eine interessante mathematische Konstruktion.

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

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.



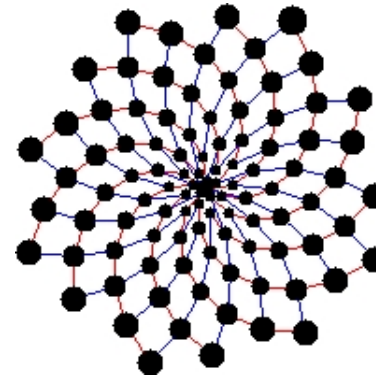
Die Muscheln sind lediglich ein Beispiel für ein verbreitetes Phänomen: das **Vorkommen der Fibonacci-Zahlen in der Natur.**

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.



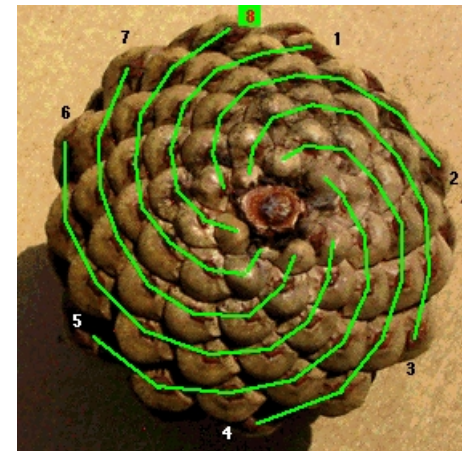
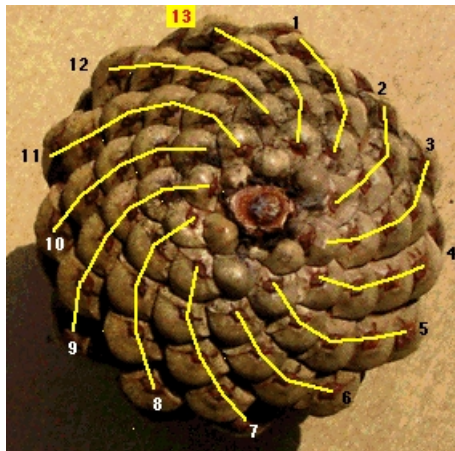
Konkretes Naturbeispiel: Die Sonnenblume

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 $360^\circ - 360^\circ/\phi \approx 137,518\dots^\circ$.

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 ausgebauten mathematischen auch sehr gute sprachliche Fähigkeiten.

Insbesondere müssen Kundenwünsche richtig gedeutet und verstanden 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 (das selbe 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. Wieviel Karton benötigt man für einen “Milchbeutel” von 1 Liter Fassungsvermögen ?
4. 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 ?

5. 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.