

## 21 RELATIONEN

### 21.1 ÄQUIVALENZRELATIONEN

#### 21.1.1 Definition

In Abschnitt 15.2.1 hatten wir schon einmal erwähnt, dass eine Relation  $R \subseteq M \times M$  auf einer Menge  $M$ , die

- reflexiv,
- symmetrisch und
- transitiv

ist, *Äquivalenzrelation* heißt. Das bedeutet also, dass

*Äquivalenzrelation*

- für alle  $x \in M$  gilt:  $(x, x) \in R$
- für alle  $x, y \in M$  gilt: wenn  $(x, y) \in R$ , dann auch  $(y, x) \in R$
- für alle  $x, y, z \in M$  gilt: wenn  $(x, y) \in R$  und  $(y, z) \in R$ , dann auch  $(x, z) \in R$ .

Für Äquivalenzrelationen benutzt man oft Symbole wie  $\equiv$ ,  $\sim$  oder  $\approx$ , die mehr oder weniger deutlich an das Gleichheitszeichen erinnern, sowie Infixschreibweise. Dann liest sich die Definition so: Eine Relation  $\equiv$  ist Äquivalenzrelation auf einer Menge  $M$ , wenn gilt:

- $\forall x \in M : x \equiv x$ ,
- $\forall x \in M : \forall y \in M : x \equiv y \implies y \equiv x$
- $\forall x \in M : \forall y \in M : \forall z \in M : x \equiv y \wedge y \equiv z \implies x \equiv z$

Der Anlass für Symbole, die an das „ $=$ “ erinnern, ist natürlich der, dass Gleichheit, also die Relation  $I = \{(x, x) \mid x \in M\}$ , auf jeder Menge eine Äquivalenzrelation ist, denn offensichtlich gilt:

- $\forall x \in M : x = x$ ,
- $\forall x \in M : \forall y \in M : x = y \implies y = x$
- $\forall x \in M : \forall y \in M : \forall z \in M : x = y \wedge y = z \implies x = z$

Ein klassisches Beispiel sind die „Kongruenzen modulo  $n$ “ auf den ganzen Zahlen. Es sei  $n \in \mathbb{N}_+$ . Zwei Zahlen  $x, y \in \mathbb{Z}$  heißen *kongruent modulo  $n$* , wenn die Differenz  $x - y$  durch  $n$  teilbar, also ein ganzzahliges Vielfaches von  $n$ , ist. Man schreibt typischerweise  $x \equiv y \pmod{n}$ . Dass dies tatsächlich Äquivalenzrelationen sind, haben Sie in Mathematikvorlesungen mehr oder weniger explizit gesehen.

*kongruent modulo  $n$*

- Die Reflexivität ergibt sich aus der Tatsache, dass  $x - x = 0$  Vielfaches von  $n$  ist.
- Die Symmetrie gilt, weil mit  $x - y$  auch  $y - x = -(x - y)$  Vielfaches von  $n$  ist.
- Transitivität: Wenn  $x - y = k_1 n$  und  $y - z = k_2 n$  (mit  $k_1, k_2 \in \mathbb{Z}$ ), dann ist auch  $x - z = (x - y) + (y - z) = (k_1 + k_2)n$  ein ganzzahliges Vielfaches von  $n$ .

### 21.1.2 Äquivalenzklassen und Faktormengen

*Äquivalenzklasse* Für  $x \in M$  heißt  $\{y \in M \mid x \equiv y\}$  die *Äquivalenzklasse von  $x$* . Man schreibt für die Äquivalenzklasse von  $x$  mitunter  $[x]_{\equiv}$  oder einfach  $[x]$ , falls klar ist, welche Äquivalenzrelation gemeint ist.

*Faktormenge* Für die Menge aller Äquivalenzklassen schreibt man  $M/_{\equiv}$  und nennt das manchmal auch die *Faktormenge* oder *Faserung von  $M$  nach  $\equiv$* , also  $M/_{\equiv} = \{[x]_{\equiv} \mid x \in M\}$ .

*Faserung* Ist konkret  $\equiv$  die Äquivalenzrelation „modulo  $n$ “ auf den ganzen Zahlen, dann schreibt man für die Faktormenge auch  $\mathbb{Z}_n$ .

## 21.2 KONGRUENZRELATIONEN

Mitunter hat eine Menge  $M$ , auf der eine Äquivalenzrelation definiert ist, zusätzliche „Struktur“, bzw. auf  $M$  sind eine oder mehrere Operationen definiert. Als Beispiel denke man etwa an die ganzen Zahlen  $\mathbb{Z}$  mit der Addition. Man kann sich dann z. B. fragen, wie sich Funktionswerte ändern, wenn man Argumente durch andere, aber äquivalente ersetzt.

### 21.2.1 Verträglichkeit von Relationen mit Operationen

Um den Formalismus nicht zu sehr aufzublähen, beschränken wir uns in diesem Unterabschnitt auf die zwei am häufigsten vorkommenden einfachen Fälle.

*verträglich* Es sei  $\equiv$  eine Äquivalenzrelation auf einer Menge  $M$  und  $f : M \rightarrow M$  eine Abbildung. Man sagt, dass  $\equiv$  mit  $f$  *verträglich* ist, wenn für alle  $x_1, x_2 \in M$  gilt:

$$x_1 \equiv x_2 \implies f(x_1) \equiv f(x_2) .$$

*verträglich* Ist  $\square$  eine binäre Operation auf einer Menge  $M$ , dann heißen  $\equiv$  und  $\square$  *verträglich*, wenn für alle  $x_1, x_2 \in M$  und alle  $y_1, y_2 \in M$  gilt:

$$x_1 \equiv x_2 \wedge y_1 \equiv y_2 \implies x_1 \square y_1 \equiv x_2 \square y_2 .$$

Ein typisches Beispiel sind wieder die Äquivalenzrelationen „modulo  $n$ “. Diese Relationen sind mit Addition, Subtraktion und Multiplikation verträglich. Ist etwa

$$\begin{aligned} x_1 \equiv x_2 \pmod{n} & \quad \text{also} \quad x_1 - x_2 = kn \\ \text{und} \quad y_1 \equiv y_2 \pmod{n} & \quad \text{also} \quad y_1 - y_2 = mn \end{aligned}$$

dann ist zum Beispiel

$$(x_1 + y_1) - (x_2 + y_2) = (x_1 - x_2) + (y_1 - y_2) = (k + m)n .$$

Mit anderen Worten ist dann auch

$$x_1 + y_1 \equiv x_2 + y_2 \pmod{n}.$$

Eine Äquivalenzrelation, die mit allen gerade interessierenden Funktionen oder/und Operationen verträglich ist, nennt man auch eine *Kongruenzrelation*.

*Kongruenzrelation*

### 21.2.2 Wohldefiniertheit von Operationen mit Äquivalenzklassen

Wann immer man eine Kongruenzrelation vorliegen hat, also z. B. eine Äquivalenzrelation  $\equiv$  auf  $M$  die mit einer binären Operation  $\square$  auf  $M$  verträglich ist, induziert diese Operation auf  $M$  eine Operation auf  $M/\equiv$ . Analoges gilt für Abbildungen  $f : M \rightarrow M$ .

*induzierte Operation*

## 21.3 HALBORDNUNGEN

Eine Ihnen wohlvertraute Halbordnung ist die Mengeninklusion  $\subseteq$ . Entsprechende Beispiele tauchen daher im folgenden immer wieder auf.

### 21.3.1 Grundlegende Definitionen

Eine Relation  $R \subseteq M \times M$  heißt *antisymmetrisch*, wenn für alle  $x, y \in M$  gilt:

*Antisymmetrie*

$$xRy \wedge yRx \implies x = y$$

Eine Relation  $R \subseteq M \times M$  heißt *Halbordnung*, wenn sie

*Halbordnung*

- reflexiv,
- antisymmetrisch und
- transitiv

ist. Wenn  $R$  eine Halbordnung auf einer Menge  $M$  ist, sagt man auch, die Menge sei *halbgeordnet*.

*halbgeordnete Menge*

Auf der Menge aller Wörter über einem Alphabet  $A$  ist die Relation  $\sqsubseteq_p$  ein einfaches Beispiel, die definiert sei vermöge der Festlegung  $w_1 \sqsubseteq_p w_2 \iff \exists u \in A^* : w_1u = w_2$ . Machen Sie sich zur Übung klar, dass sie tatsächlich die drei definierenden Eigenschaften einer Halbordnung hat.

Es sei  $M'$  eine Menge und  $M = 2^{M'}$  die Potenzmenge von  $M'$ . Dann ist die Mengeninklusion  $\subseteq$  eine Halbordnung auf  $M$ . Auch hier sollten Sie noch einmal aufschreiben, was die Aussagen der drei definierenden Eigenschaften einer Halbordnung sind. Sie werden merken, dass die Antisymmetrie eine Möglichkeit an die Hand gibt, die Gleichheit zweier Mengen zu beweisen (wir haben das auch schon ausgenutzt).

Wenn  $R$  Halbordnung auf einer *endlichen* Menge  $M$  ist, dann stellt man sie manchmal graphisch dar. Wir hatten schon in Unterabschnitt 15.1.4 darauf hingewiesen, dass Relationen und gerichtete Graphen sich formal nicht unterscheiden. Betrachten wir als Beispiel die halbgeordnete Menge  $(2^{\{a,b,c\}}, \subseteq)$ . Im zugehörigen Graphen führt eine Kante von  $M_1$  zu  $M_2$ , wenn  $M_1 \subseteq M_2$  ist. Es ergibt sich also die Darstellung aus Abbildung 21.1. Wie man sieht wird das ganze recht schnell

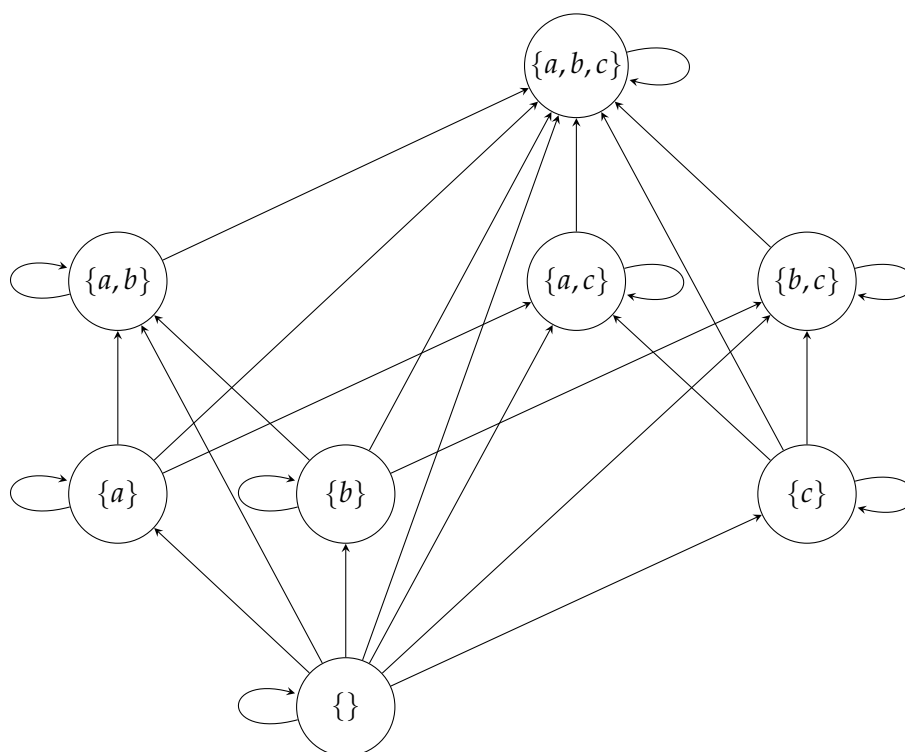


Abbildung 21.1: Die Halbordnung  $(2^{\{a,b,c\}}, \subseteq)$  als Graph

relativ unübersichtlich. Dabei ist ein Teil der Kanten nicht ganz so wichtig, weil deren Existenz ohnehin klar ist (wegen der Reflexivität) oder aus anderen Kanten gefolgert werden kann (wegen der Transitivität). Deswegen wählt man meist die Darstellung als sogenanntes *Hasse-Diagramm* dar. Das ist eine Art „Skelett“ der Halbordnung, bei dem die eben angesprochenen Kanten fehlen. Genauer gesagt ist es der Graph der Relation  $H_R = (R \setminus I) \setminus (R \setminus I)^2$ . In unserem Beispiel ergibt sich aus Abbildung 21.1 durch Weglassen der Kanten Abbildung 21.2.

Vom Hassediagramm kommt man „ganz leicht“ wieder zur ursprünglichen Halbordnung: Man muss nur die reflexiv-transitive Hülle bilden.

*Hasse-Diagramm*

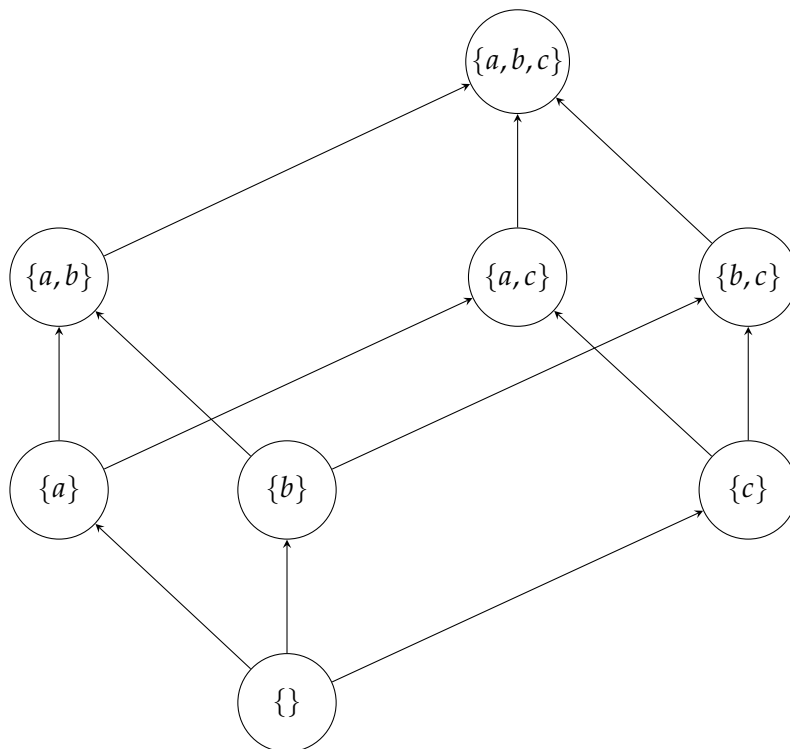


Abbildung 21.2: Hassediagramm der Halbordnung  $(2^{\{a,b,c\}}, \subseteq)$

**21.1 Lemma.** Wenn  $R$  eine Halbordnung auf einer endlichen Menge  $M$  ist und  $H_R$  das zugehörige Hassediagramm, dann ist  $H_R^* = R$ .

**21.2 Beweis.**  $R$  und  $H_R$  sind beides Relationen über der gleichen Grundmenge  $M$  (also ist  $R^0 = H_R^0$ ) und offensichtlich ist  $H_R \subseteq R$ . Eine ganz leichte Induktion (machen Sie sie) zeigt, dass für alle  $i \in \mathbb{N}_0$  gilt:  $H_R^i \subseteq R^i$  und folglich  $H_R^* \subseteq R^*$ . Da  $R$  reflexiv und transitiv ist, ist  $R^* = R$ , also  $H_R^* \subseteq R$ .

Nun wollen wir zeigen, dass umgekehrt auch gilt:  $R \subseteq H_R^*$ . Sei dazu  $(x, y) \in R$ . Falls  $x = y$  ist, ist auch  $(x, y) \in I \subseteq H_R^*$ .

Sei daher im folgenden  $x \neq y$ , also  $(x, y) \in R \setminus I$ , und sein  $(x_0, x_1, \dots, x_m)$  eine Folge von Elementen mit folgenden Eigenschaften:

- $x_0 = x$  und  $x_m = y$
- für alle  $0 \leq i < m$  ist  $x_i \subseteq x_{i+1}$
- für alle  $0 \leq i < m$  ist  $x_i \neq x_{i+1}$

In einer solchen Folge kann kein Element  $z \in M$  zweimal auftauchen. Wäre nämlich  $x_i = z$  und  $x_k = z$  mit  $k > i$ , dann wäre jedenfalls  $k \geq i + 2$  und  $x_{i+1} \neq z$ . Folglich

wäre einerseits  $z = x_i \sqsubseteq x_{i+1}$  und andererseits  $x_{i+1} \sqsubseteq \dots \sqsubseteq x_k$ , also wegen Transitivität  $x_{i+1} \sqsubseteq x_k = z$ . Aus der Antisymmetrie von  $\sqsubseteq$  würde  $x_{i+1} = z$  folgen im Widerspruch zum eben Festgehaltenen.

Da in einer Folge  $(x_0, x_1, \dots, x_m)$  der beschriebenen Art kein Element zweimal vorkommen kann und  $M$  endlich ist, gibt es auch maximal lange solche Folgen. Sei im folgenden  $(x_0, x_1, \dots, x_m)$  maximal lang. Dann gilt also für alle  $0 \leq i < m$ , dass man zwischen zwei Elemente  $x_i$  und  $x_{i+1}$  kein weiteres Element einfügen kann, dass also gilt:  $\neg \exists z \in M : x_i \sqsubseteq z \sqsubseteq x_{i+1} \wedge x_i \neq z \wedge x_{i+1} \neq z$ .

Dafür kann man auch schreiben:  $\neg \exists z \in M : (x_i, z) \in R \setminus I \wedge (z, x_{i+1}) \in R \setminus I$ , d. h.  $(x_i, x_{i+1}) \notin (R \setminus I)^2$ .

Also gilt für alle  $0 \leq i < m$ :  $(x_i, x_{i+1}) \in (R \setminus I) \setminus (R \setminus I)^2 = H_R$ . Daher ist  $(x, y) = (x_0, x_m) \in H_R^m \subseteq H_R^*$ . ■

*gerichtete azyklische Graphen*  
*Dag*

Graphen, die das Hassediagramm einer endlichen Halbordnung sind, heißen auch *gerichtete azyklische Graphen* (im Englischen *directed acyclic graph* oder kurz *Dag*), weil sie keine Zyklen mit mindestens einer Kante enthalten. Denn andernfalls hätte man eine Schlinge oder (fast die gleiche Argumentation wie eben im Beweis 21.2) des Lemmas verschiedene Elemente  $x$  und  $y$  mit  $x \sqsubseteq y$  und  $y \sqsubseteq x$ .

Gerichtete azyklische Graphen tauchen an vielen Stellen in der Informatik auf, nicht nur natürlich bei Problemstellungen im Zusammenhang mit Graphen, sondern z. B. auch bei der Darstellung von Datenflüssen, im Compilerbau, bei sogenannten *binary decision diagrams* zur Darstellung logischer Funktionen usw.

### 21.3.2 „Extreme“ Elemente

Es sei  $(M, \sqsubseteq)$  eine halbgeordnete Menge und  $T$  eine beliebige Teilmenge von  $M$ .

*minimales Element*  
*maximales Element*

Ein Element  $x \in T$  heißt *minimales Element* von  $T$ , wenn es kein  $y \in T$  gibt mit  $y \sqsubseteq x$  und  $y \neq x$ . Ein Element  $x \in T$  heißt *maximales Element* von  $T$ , wenn es kein  $y \in T$  gibt mit  $x \sqsubseteq y$  und  $x \neq y$ .

*größtes Element*  
*kleinstes Element*


Ein Element  $x \in T$  heißt *größtes Element* von  $T$ , wenn für alle  $y \in T$  gilt:  $y \sqsubseteq x$ . Ein Element  $x \in T$  heißt *kleinstes Element* von  $T$ , wenn für alle  $y \in T$  gilt:  $x \sqsubseteq y$ .

Eine Teilmenge  $T$  kann mehrere minimale (bzw. maximale) Elemente besitzen, aber nur ein kleinstes (bzw. größtes). Als Beispiel betrachte man die Teilmenge  $T \subseteq 2^{\{a,b,c\}}$  aller Teilmengen von  $\{a, b, c\}$ , die nichtleer sind. Diese Teilmenge besitzt die drei minimalen Elemente  $\{a\}$ ,  $\{b\}$  und  $\{c\}$ . Und sie besitzt ein größtes Element, nämlich  $\{a, b, c\}$ .

*obere Schranke*  
*untere Schranke*

Ein Element  $x \in M$  heißt *obere Schranke* von  $T$ , wenn für alle  $y \in T$  gilt:  $y \sqsubseteq x$ . Ein  $x \in M$  heißt *untere Schranke* von  $T$ , wenn für alle  $y \in T$  gilt:  $x \sqsubseteq y$ . In der

Halbordnung  $(2^{\{a,b,c\}}, \subseteq)$  besitzt zum Beispiel  $T = \{\{\}, \{a\}, \{b\}\}$  zwei obere Schranken:  $\{a, b\}$  und  $\{a, b, c\}$ . Die Teilmenge  $T = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$  besitzt die gleichen oberen Schranken.

In einer Halbordnung muss nicht jede Teilmenge eine obere Schranke besitzen. Zum Beispiel besitzt die Teilmenge aller Elemente der Halbordnung mit dem Hassediagramm  keine obere Schranke.

Besitzt die Menge der oberen Schranken einer Teilmenge  $T$  ein kleinstes Element, so heißt dies das *Supremum* von  $T$  und wir schreiben dafür  $\sqcup T$  (oder  $\sup(T)$ ).

*Supremum*  
 $\sqcup T, \sup(T)$

Besitzt die Menge der unteren Schranken einer Teilmenge  $T$  ein größtes Element, so heißt dies das *Infimum* von  $T$ . Das werden wir in dieser Vorlesung aber nicht benötigen.

*Infimum*

Wenn eine Teilmenge kein Supremum besitzt, dann kann das daran liegen, dass sie gar keine oberen Schranken besitzt, oder daran, dass die Menge der oberen Schranken kein kleinstes Element hat. Liegt eine Halbordnung der Form  $(2^M, \subseteq)$ , dann besitzt aber jede Teilmenge  $T \subseteq 2^M$  ein Supremum.  $\sqcup T$  ist dann nämlich die Vereinigung aller Elemente von  $T$  (die Teilmengen von  $M$  sind).

### 21.3.3 Vollständige Halbordnungen

Eine *aufsteigende Kette* ist eine abzählbar unendliche Folge  $(x_0, x_1, x_2, \dots)$  von Elementen einer Halbordnung mit der Eigenschaft:  $\forall i \in \mathbb{N}_0 : x_i \subseteq x_{i+1}$ .

*aufsteigende Kette*

Eine Halbordnung heißt *vollständig*, wenn sie ein kleinstes Element besitzt und jede aufsteigende Kette ein Supremum besitzt. Für das kleinste Element schreiben wir im folgenden  $\perp$ . Für das Supremum einer aufsteigenden Kette  $x_0 \subseteq x_1 \subseteq x_2 \subseteq \dots$  schreiben wir  $\sqcup_i x_i$ .

*vollständige Halbordnung*

Ein ganz wichtiges Beispiel für eine vollständige Halbordnung ist die schon mehrfach erwähnte Potenzmenge  $2^{M'}$  einer Menge  $M'$  mit Mengeninklusion  $\subseteq$  als Relation. Das kleinste Element ist die leere Menge  $\emptyset$ . Und das Supremum einer aufsteigenden Kette  $T_0 \subseteq T_1 \subseteq T_2 \subseteq \dots$  ist  $\sqcup_i T_i = \cup T_i$ .

Andererseits ist  $(\mathbb{N}_0, \leq)$  *keine* vollständige Halbordnung, denn unbeschränkt wachsende aufsteigende Ketten wie z. B.  $0 \leq 1 \leq 2 \leq \dots$  besitzen kein Supremum in  $\mathbb{N}_0$ . Wenn man aber noch ein weiteres Element  $u$  „über“ allen Zahlen hinzufügt, dann ist die Ordnung vollständig. Man setzt also  $N = \mathbb{N}_0 \cup \{u\}$  und definiert

$$x \subseteq y \iff (x, y \in \mathbb{N}_0 \wedge x \leq y) \vee (y = u)$$

Weil wir es später noch brauchen können, definieren wir auch noch  $N' = \mathbb{N}_0 \cup \{u_1, u_2\}$  mit der totalen Ordnung

$$x \subseteq y \iff (x, y \in \mathbb{N}_0 \wedge x \leq y) \vee (x \in \mathbb{N}_0 \cup \{u_1\} \wedge y = u_1) \vee y = u_2$$

also sozusagen

$$0 \sqsubseteq 1 \sqsubseteq 2 \sqsubseteq 3 \sqsubseteq \dots \sqsubseteq u_1 \sqsubseteq u_2$$

Ein anderes Beispiel einer (sogar totalen) Ordnung, die *nicht* vollständig ist, werden wir am Ende von Abschnitt 21.4 sehen.

#### 21.3.4 Stetige Abbildungen auf vollständigen Halbordnungen

monotone Abbildung

Es sei  $\sqsubseteq$  eine Halbordnung auf einer Menge  $M$ . Eine Abbildung  $f : M \rightarrow M$  heißt *monoton*, wenn für alle  $x, y \in M$  gilt:  $x \sqsubseteq y \implies f(x) \sqsubseteq f(y)$ .

Die Abbildung  $f(x) = x + 1$  etwa ist auf der Halbordnung  $(\mathbb{N}_0, \leq)$  monoton. Die Abbildung  $f(x) = x \bmod 5$  ist auf der gleichen Halbordnung dagegen nicht monoton, denn es ist zwar  $3 \leq 10$ , aber  $f(3) = 3 \not\leq 0 = f(10)$ .

stetige Abbildung

Eine monotone Abbildung  $f : D \rightarrow D$  auf einer vollständigen Halbordnung  $(D, \sqsubseteq)$  heißt *stetig*, wenn für jede aufsteigende Kette  $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$  gilt:  $f(\bigsqcup_i x_i) = \bigsqcup_i f(x_i)$ .

Betrachten wir als erstes Beispiel noch einmal die vollständige Halbordnung  $N' = \mathbb{N}_0 \cup \{u_1, u_2\}$  von oben. Die Abbildung  $f : N' \rightarrow N'$  mit

$$f(x) = \begin{cases} x + 1 & \text{falls } x \in \mathbb{N}_0 \\ u_1 & \text{falls } x = u_1 \\ u_2 & \text{falls } x = u_2 \end{cases}$$

ist stetig. Denn für jede aufsteigende Kette  $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$  gibt es nur zwei Möglichkeiten:

- Die Kette wird konstant. Es gibt also ein  $n' \in N'$  und ein  $i \in \mathbb{N}_0$  so dass gilt:  $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots \sqsubseteq x_i = x_{i+1} = x_{i+2} = \dots = n'$ . Dann ist jedenfalls  $\bigsqcup_i x_i = n'$ . Es gibt nun drei Unterfälle zu betrachten:
  - Wenn  $n' = u_2$  ist, dann ist wegen  $f(u_2) = u_2$  ist auch  $\bigsqcup_i f(x_i) = u_2$ , also ist  $f(\bigsqcup_i x_i) = \bigsqcup_i f(x_i)$ .
  - Wenn  $n' = u_1$ , gilt eine analoge Überlegung.
  - Wenn  $n' \in \mathbb{N}_0$  ist, dann ist  $f(\bigsqcup_i x_i) = f(n') = n' + 1$ . Andererseits ist die Kette der Funktionswerte  $f(x_0) \sqsubseteq f(x_1) \sqsubseteq f(x_2) \sqsubseteq \dots \sqsubseteq f(x_i) = f(x_{i+1}) = f(x_{i+2}) = \dots = f(n') = n' + 1$ . Also ist  $f(\bigsqcup_i x_i) = \bigsqcup_i f(x_i)$ .
- Der einzige andere Fall ist: die Kette wird nicht konstant. Dann müssen alle  $x_i \in \mathbb{N}_0$  sein, und die Kette wächst unbeschränkt. Das gleiche gilt dann auch für die Kette der Funktionswerte. Also haben beide als Supremum  $u_1$  und wegen  $f(u_1) = u_1$  ist  $f(\bigsqcup_i x_i) = \bigsqcup_i f(x_i)$ .



Der letzte Fall zeigt einem auch gleich schon, dass dagegen die folgende Funktion  $g : N' \rightarrow N'$  nicht stetig ist:

$$g(x) = \begin{cases} x+1 & \text{falls } x \in \mathbb{N}_0 \\ u_2 & \text{falls } x = u_1 \\ u_2 & \text{falls } x = u_2 \end{cases}$$

Der einzige Unterschied zu  $f$  ist, dass nun  $g(u_1) = u_2$ . Eine unbeschränkt wachsende Kette  $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$  nichtnegativer ganzer Zahlen hat Supremum  $u_1$ , so dass  $g(\bigsqcup_i x_i) = u_2$  ist. Aber die Kette der Funktionswerte  $g(x_0) \sqsubseteq g(x_1) \sqsubseteq g(x_2) \sqsubseteq \dots$  hat Supremum  $\bigsqcup_i g(x_i) = u_1 \neq g(\bigsqcup_i x_i)$ .

Der folgende Satz ist eine abgeschwächte Version des sogenannten Fixpunktsatzes von Knaster und Tarski.

**21.3 Satz.** *Es sei  $f : D \rightarrow D$  eine monotone und stetige Abbildung auf einer vollständigen Halbordnung  $(D, \sqsubseteq)$  mit kleinstem Element  $\perp$ . Elemente  $x_i \in D$  seien wie folgt definiert:*

$$\begin{aligned} x_0 &= \perp \\ \forall i \in \mathbb{N}_0 : x_{i+1} &= f(x_i) \end{aligned}$$

Dann gilt:

1. Die  $x_i$  bilden eine Kette:  $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$ .
2. Das Supremum  $x_f = \bigsqcup_i x_i$  dieser Kette ist Fixpunkt von  $f$ , also  $f(x_f) = x_f$ .
3.  $x_f$  ist der kleinste Fixpunkt von  $f$ : Wenn  $f(y_f) = y_f$  ist, dann ist  $x_f \sqsubseteq y_f$ .

**21.4 Beweis.** Mit den Bezeichnungen wie im Satz gilt:

1. Dass für alle  $i \in \mathbb{N}_0$  gilt  $x_i \sqsubseteq x_{i+1}$ , sieht man durch vollständige Induktion:  $x_0 \sqsubseteq x_1$  gilt, weil  $x_0 = \perp$  das kleinste Element der Halbordnung ist. Und wenn man schon weiß, dass  $x_i \sqsubseteq x_{i+1}$  ist, dann folgt wegen der Monotonie von  $f$  sofort  $f(x_i) \sqsubseteq f(x_{i+1})$ , also  $x_{i+1} \sqsubseteq x_{i+2}$ .
2. Wegen der Stetigkeit von  $f$  ist  $f(x_f) = f(\bigsqcup_i x_i) = \bigsqcup_i f(x_i) = \bigsqcup_i x_{i+1}$ . Die Folge der  $x_{i+1}$  unterscheidet sich von der Folge der  $x_i$  nur durch das fehlende erste Element  $\perp$ . Also haben „natürlich“ beide Folgen das gleiche Supremum  $x_f$ ; also ist  $f(x_f) = x_f$ .

Falls Sie das nicht ganz „natürlich“ fanden, hier eine ganz genaue Begründung:

- Einerseits ist für alle  $i \geq 1$ :  $x_i \sqsubseteq \bigsqcup_i x_{i+1}$ . Außerdem ist  $\perp = x_0 \sqsubseteq \bigsqcup_i x_{i+1}$ . Also ist  $\bigsqcup_i x_{i+1}$  eine obere Schranke für alle  $x_i$ ,  $i \in \mathbb{N}_0$ , also ist  $\bigsqcup_i x_i \sqsubseteq \bigsqcup_i x_{i+1}$ .
- Andererseits ist für alle  $i \geq 1$ :  $x_i \sqsubseteq \bigsqcup_i x_i$ . Also ist  $\bigsqcup_i x_i$  eine obere Schranke für alle  $x_{i+1}$ ,  $i \in \mathbb{N}_0$ , also ist  $\bigsqcup_i x_{i+1} \sqsubseteq \bigsqcup_i x_i$ .

- Aus  $\sqcup_i x_i \sqsubseteq \sqcup_i x_{i+1}$  und  $\sqcup_i x_{i+1} \sqsubseteq \sqcup_i x_i$  folgt mit der Antisymmetrie von  $\sqsubseteq$  sofort die Gleichheit der beiden Ausdrücke.
3. Durch Induktion sieht man zunächst einmal:  $\forall i \in \mathbb{N}_0 : x_i \sqsubseteq y_f$ . Denn  $x_0 \sqsubseteq y_f$  gilt, weil  $x_0 = \perp$  das kleinste Element der Halbordnung ist. Und wenn man schon weiß, dass  $x_i \sqsubseteq y_f$  ist, dann folgt wegen der Monotonie von  $f$  sofort  $f(x_i) \sqsubseteq f(y_f)$ , also  $x_{i+1} \sqsubseteq y_f$ . Also ist  $y_f$  eine obere Schranke der Kette, also ist gilt für die kleinste obere Schranke:  $x_f = \sqcup_i x_i \sqsubseteq y_f$ . ■

Dieser Fixpunktsatz (und ähnliche) finden in der Informatik an mehreren Stellen Anwendung. Zum Beispiel kann er Ihnen in Vorlesungen über Semantik von Programmiersprachen wieder begegnen.

Hier können wir Ihnen schon andeuten, wie er im Zusammenhang mit kontextfreien Grammatiken nützlich sein kann. Betrachten wir als Terminalzeichenalphabet  $T = \{\mathbf{a}, \mathbf{b}\}$  und die kontextfreie Grammatik  $G = (\{X\}, T, X, P)$  mit Produktionsmenge  $P = \{X \rightarrow \mathbf{a}X\mathbf{b} \mid \varepsilon\}$ . Als halbgeordnete Menge  $D$  verwenden wir die Potenzmenge  $D = 2^{T^*}$  der Menge aller Wörter mit Inklusion als Halbordnungsrelation. Die Elemente der Halbordnung sind also Mengen von Wörtern, d. h. formale Sprachen. Kleinstes Element der Halbordnung ist die leere Menge  $\emptyset$ . Wie erwähnt, ist diese Halbordnung vollständig.

Es sei nun  $f : D \rightarrow D$  die Abbildung mit  $f(L) = \{\mathbf{a}\}L\{\mathbf{b}\} \cup \{\varepsilon\}$ . Der Bequemlichkeit halber wollen wir nun einfach glauben, dass  $f$  stetig ist. (Wenn Ihnen das nicht passt, prüfen Sie es nach. Es ist nicht schwer.) Der Fixpunktsatz besagt, dass man den kleinsten Fixpunkt dieser Abbildung erhält als Supremum, hier also Vereinigung, aller der folgenden Mengen:

$$\begin{aligned} L_0 &= \emptyset \\ L_1 &= f(L_0) = \{\mathbf{a}\}L_0\{\mathbf{b}\} \cup \{\varepsilon\} \\ &= \{\varepsilon\} \\ L_2 &= f(L_1) = \{\mathbf{a}\}L_1\{\mathbf{b}\} \cup \{\varepsilon\} \\ &= \{\mathbf{ab}, \varepsilon\} \\ L_3 &= f(L_2) = \{\mathbf{a}\}L_2\{\mathbf{b}\} \cup \{\varepsilon\} \\ &= \{\mathbf{aabb}, \mathbf{ab}, \varepsilon\} \end{aligned}$$

Sie sehen, wie der Hase läuft. Der kleinste Fixpunkt ist  $L = \{\mathbf{a}^k\mathbf{b}^k \mid k \in \mathbb{N}_0\}$ . Das ist auch genau die Sprache, die die Grammatik erzeugt. Und  $L$  ist Fixpunkt von  $f$ , also

$$L = \{\mathbf{a}\}L\{\mathbf{b}\} \cup \{\varepsilon\}$$

Es ist also sozusagen die kleinste Lösung der Gleichung  $X = \{a\}X\{b\} \cup \{\varepsilon\}$ . Was das mit den Produktionen der Grammatik zu tun, sehen Sie vermutlich.

## 21.4 ORDNUNGEN

Eine Relation  $R \subseteq M \times M$  ist eine *Ordnung*, oder auch genauer *totale Ordnung*, wenn  $R$  Halbordnung ist und außerdem gilt:

*Ordnung*  
*totale Ordnung*

$$\forall x, y \in M : xRy \vee yRx$$

Wie kann man aus der weiter vorne definierten Halbordnung  $\sqsubseteq_p$  auf  $A^*$  eine totale Ordnung machen? Dafür gibt es natürlich verschiedene Möglichkeiten. Auf jeden Fall muss aber z. B. festgelegt werden, ob  $a \sqsubseteq b$  oder  $b \sqsubseteq a$ .

Es ist also auf jeden Fall eine totale Ordnung  $\sqsubseteq_A$  auf den Symbolen des Alphabets erforderlich. Nehmen wir an, wir haben das: also z. B.  $a \sqsubseteq_A b$ .

Dann betrachtet man des öfteren zwei sogenannte *lexikographische Ordnungen*. Die eine ist die naheliegende Verallgemeinerung dessen, was man aus Wörterbüchern kennt. Die andere ist für algorithmische Zwecke besser geeignet.

*lexikographische Ordnung*

- Die lexikographische Ordnung  $\sqsubseteq_1$ , nach der Wörter in Lexika usw. sortiert sind, kann man wie folgt definieren. Seien  $w_1, w_2 \in A^*$ . Dann gibt es das eindeutig bestimmte maximal lange gemeinsame Präfix von  $w_1$  und  $w_2$ , also das maximal lange Wort  $v \in A^*$ , so dass es  $u_1, u_2 \in A^*$  gibt mit  $w_1 = v u_1$  und  $w_2 = v u_2$ . Drei Fälle sind möglich:
  1. Falls  $v = w_1$  ist, gilt  $w_1 \sqsubseteq_1 w_2$ .
  2. Falls  $v = w_2$  ist, gilt  $w_2 \sqsubseteq_1 w_1$ .
  3. Falls  $w_1 \neq v \neq w_2$ , gibt es  $x, y \in A$  und  $u'_1, u'_2 \in A^*$  mit
    - $x \neq y$  und
    - $w_1 = v x u'_1$  und  $w_2 = v y u'_2$ .

Dann gilt  $w_1 \sqsubseteq_1 w_2 \iff x \sqsubseteq_A y$ .

Muss man wie bei einem Wörterbuch nur endlich viele Wörter ordnen, dann ergibt sich zum Beispiel

$$\begin{aligned} a \sqsubseteq_1 aa \sqsubseteq_1 aaa \sqsubseteq_1 aaaa \\ \sqsubseteq_1 ab \sqsubseteq_1 aba \sqsubseteq_1 abbb \\ \sqsubseteq_1 b \sqsubseteq_1 baaaaa \sqsubseteq_1 baab \\ \sqsubseteq_1 bbbbb \end{aligned}$$

Allgemein auf der Menge aller Wörter ist diese Ordnung aber nicht ganz so „harmlos“. Wir gehen gleich noch darauf ein.

- Eine andere lexikographische Ordnung  $\sqsubseteq_2$  auf  $A^*$  kann man definieren, indem man festlegt, dass  $w_1 \sqsubseteq_2 w_2$  genau dann gilt, wenn
  - entweder  $|w_1| < |w_2|$
  - oder  $|w_1| = |w_2|$  und  $w_1 \sqsubseteq_1 w_2$  gilt.

Diese Ordnung beginnt also z. B. im Falle  $A = \{a, b\}$  mit der naheliegenden Ordnung  $\sqsubseteq_A$  so:

$$\begin{aligned} \varepsilon &\sqsubseteq_2 a \sqsubseteq_2 b \\ &\sqsubseteq_2 aa \sqsubseteq_2 ab \sqsubseteq_2 ba \sqsubseteq_2 bb \\ &\sqsubseteq_2 aaa \sqsubseteq_2 \dots \sqsubseteq_2 bbb \\ &\sqsubseteq_2 aaaa \sqsubseteq_2 \dots \sqsubseteq_2 bbbb \\ &\dots \end{aligned}$$

Wir wollen noch darauf hinweisen, dass die lexikographische Ordnung  $\sqsubseteq_1$  als Relation auf der Menge aller Wörter einige Eigenschaften hat, an die man als Anfänger vermutlich nicht gewöhnt ist. Zunächst einmal merkt man, dass die Ordnung nicht vollständig ist. Die aufsteigende Kette

$$\varepsilon \sqsubseteq_1 a \sqsubseteq_1 aa \sqsubseteq_1 aaa \sqsubseteq_1 aaaa \sqsubseteq_1 \dots$$

besitzt kein Supremum. Zwar ist jedes Wort, das mindestens ein  $b$  enthält, obere Schranke, aber es gibt keine kleinste. Das merkt man, wenn man die absteigende Kette

$$b \supseteq_1 ab \supseteq_1 aab \supseteq_1 aaab \supseteq_1 aaaab \supseteq_1 \dots$$

betrachtet. Jede obere Schranke der aufsteigenden Kette muss ein  $b$  enthalten. Aber gleich, welche obere Schranke  $w$  man betrachtet, das Wort  $a^{|w|}b$  ist eine echt kleinere obere Schranke. Also gibt es keine kleinste.

Dass es sich bei obigen Relationen überhaupt um totale Ordnungen handelt, ist auch unterschiedlich schwer zu sehen. Als erstes sollte man sich klar machen, dass  $\sqsubseteq_1$  auf der Menge  $A^n$  aller Wörter einer festen Länge  $n$  eine totale Ordnung ist. Das liegt daran, dass für verschiedene Wörter gleicher Länge niemals Punkt 1 oder Punkt 2 zutrifft. Und da  $\sqsubseteq_A$  als totale Ordnung vorausgesetzt wird, ist in Punkt 3 stets  $x \sqsubseteq_A y$  oder  $y \sqsubseteq_A x$  und folglich  $w_1 \sqsubseteq_1 w_2$  oder  $w_2 \sqsubseteq_1 w_1$ .

Daraus folgt schon einmal das auch  $\sqsubseteq_2$  auf der Menge  $A^n$  aller Wörter einer festen Länge  $n$  eine totale Ordnung ist, und damit überhaupt eine totale Ordnung.

Für  $\sqsubseteq_1$  muss man dafür noch einmal genauer Wörter unterschiedlicher Länge in Betracht ziehen. Wie bei der Formulierung der Definition schon suggeriert, decken die drei Punkte alle Möglichkeiten ab.

## 21.5 AUSBLICK

Vollständige Halbordnungen spielen zum Beispiel eine wichtige Rolle, wenn man sich mit sogenannter denotationaler Semantik von Programmiersprachen beschäftigt und die Bedeutung von while-Schleifen und Programmen mit rekursiven Funktionsaufrufen präzisieren will. Den erwähnten Fixpunktsatz (oder verwandte Ergebnisse) kann man auch zum Beispiel bei der automatischen statischen Datenflussanalyse von Programmen ausnutzen. Diese und andere Anwendungen werden ihnen in weiteren Vorlesungen begegnen.

