Kapitel 2

Elemente der Prädikatenlogik

2.1 Begriffe aus der universellen Algebra

Definition 2.1 *Eine* funktionale Struktur *M besteht aus einer Menge $\underline{M}$ (der* Grundmenge *von M) zusammen mit ausgezeichneten, benannten Operationen. Die Operationen sind Funktionen der Form $\underline{M}^k \to \underline{M}$ für $k \in \mathbb{N}$. Die Zahl k heißt hierbei die Stelligkeit der Operation. Nullstellige Operationen entsprechen ausgezeichneten Elementen von $\underline{M}$ und werden daher als Konstanten bezeichnet.*

Hier einige Beispiele:

- Die Struktur $(\mathbb{N}, 0, 1, +, \times)$ mit der Grundmenge $\mathbb{N}$, den nullstelligen Operationen $0 \in \mathbb{N}$ und $1 \in \mathbb{N}$ sowie den zweistelligen Operationen $+\colon \mathbb{N}^2 \to \mathbb{N}$ und $\times\colon \mathbb{N}^2 \to \mathbb{N}$.
- Die Struktur $(\mathbb{N}, 0, S)$ mit der Grundmenge $\mathbb{N}$, der nullstelligen Operation 0 und der einstelligen Operation $S\colon \mathbb{N} \to \mathbb{N}$, $x \mapsto x + 1$.
- Die Struktur $(\mathbb{Z}, 0, 1, -, +, \times)$ mit der Grundmenge $\mathbb{Z}$, den nullstelligen Operationen $0 \in \mathbb{N}$ und $1 \in \mathbb{N}$, der einstelligen Operation $-\colon \mathbb{N} \to \mathbb{N}$ (also $x \mapsto -x$), sowie den zweistelligen Operationen $+\colon \mathbb{N}^2 \to \mathbb{N}$ und $\times\colon \mathbb{N}^2 \to \mathbb{N}$.
- Die Struktur $(\mathbb{Q}, 0, 1, -, +, \times)$ mit der Grundmenge $\mathbb{Q}$, den nullstelligen Operationen $0 \in \mathbb{N}$ und $1 \in \mathbb{N}$, der einstelligen Operation $-\colon \mathbb{N} \to \mathbb{N}$ (also $x \mapsto -x$), sowie den zweistelligen Operationen $+\colon \mathbb{N}^2 \to \mathbb{N}$ und $\times\colon \mathbb{N}^2 \to \mathbb{N}$.
- Für jede Gruppe G die Struktur $(G, 1, \times, {}^{-1})$, wobei 1 das neutrale Element von G ist, $\times$ die Gruppenoperation und ${}^{-1}$ die Abbildung, die jedes Element auf sein Inverses abbildet.
- Für jede abelsche Gruppe G die Struktur $(G, 0, +, -)$,wobei 0 das neutrale Element von G ist, $+$ die Gruppenoperation und $-$ die (einstellige) Abbildung, die jedes Element auf sein Inverses abbildet.

Bei den letzten beiden Beispielen ist zu beachten, dass im Falle einer abelschen Gruppe G die beiden Strukturen $(G, 1, \times, {}^{-1})$ und $(G, 0, +, -)$ formal verschieden sind, weil die Operationen unterschiedlich benannt sind.

Angaben wie $(\mathbb{N}, 0, 1, +, \times, <)$ sind allgemein üblich, aber etwas schlampig. Dabei haben die Symbole wie 0, $+$ und $<$ zwei Funktionen. Einerseits geben sie die Namen der Operationen und Relationen an, also die für sie benutzten Symbole. Andererseits stehen sie aber auch für die jeweiligen Operationen und Relationen selbst. Eigentlich müsste man dazwischen unterscheiden, weil man sonst z.B. Schwierigkeiten bekommt, über eine Struktur zu sprechen, deren Grundmenge $\mathbb{N}$ ist und die eine mit $+$ benannte Operation hat, die durch $+(x, y) = 2^{x^2+y!} - x$ definiert ist.

Definition 2.2 *Die* Signatur *einer Struktur besteht aus den Namen der ausgezeichneten Operationen zusammen mit der Information, wieviele Stellen jede hat. Die Elemente der Signatur heißen* Operationssymbole.

Eine Struktur, die man durch Weglassen von einigen der benannten Operationssymbole erhält, heißt ein Redukt *der ursprünglichen Struktur.*

Eine Struktur der Signatur σ bezeichnen wir auch als σ-Struktur.

Formal können wir eine Signatur σ beispielsweise auffassen als ein Paar $\sigma = (\sigma^{\mathrm{Op}}, \mathrm{ar})$, wobei σ^{Op} eine beliebige Menge ist (deren Elemente dann Operationssymbole heißen) und $\mathrm{ar}\colon \sigma^{\mathrm{Op}} \to \mathbb{N}$ die Stelligkeiten der Symbole angibt. Manchmal nehmen wir es genau und unterscheiden zwischen dem (k-stelligen) Operationssymbol f und der zugehörigen Operation $f^M\colon \underline{M}^k \to \underline{M}$. Das ist vor allem dann sinnvoll, wenn wir zwei verschiedene Strukturen derselben Signatur betrachten, oder allgemeiner zwei Strukturen, deren Signaturen sich überschneiden.

Definition 2.3 *Ein Homomorphismus $H\colon M \cong N$ von einer Struktur M zu einer Struktur N derselben Signatur ist eine Abbildung $H\colon \underline{M} \to \underline{N}$ zwischen den Grundmengen, die mit den Operationen vertauscht. Das heißt:*

Für jedes k-stellige Operationssymbol f und jedes k-Tupel $(a_1, \ldots, a_k) \in \underline{M}^k$ gilt $H(f(a_1, \ldots, a_k)) = f(H(a_1), \ldots, H(a_k))$.

Zwei Strukturen heißen isomorph*, falls sie dieselbe Signatur haben und es einen Isomorphismus zwischen ihnen gibt, d.h. einen bijektiven Homomorphismus.*

Definition 2.4 *Eine Substruktur einer Struktur M ist eine Struktur N mit derselben Signatur, so dass $\underline{N} \subseteq \underline{M}$ gilt und die Operationen von N sich durch Einschränkung der Operationen von M ergeben.*

Wir fixieren eine abzählbare (und abgezählte) Menge $\mathbb{X} = \{\mathsf{x}^{\mathsf{0}}, \mathsf{x}^{\mathsf{1}}, \mathsf{x}^{\mathsf{2}}, \ldots, \}$ von Symbolen, die wir *Variable* nennen.

Definition 2.5 *Die Menge der σ-*Terme *ist die kleinste Sprache*[1] *über dem Alphabet $\sigma^{\mathrm{Op}} \cup \mathbb{X}$, welche die folgenden Bedingungen erfüllt.*

- *Jede Variable ist ein σ-Term.*
- *Wenn $f \in \sigma^{\mathrm{Op}}$ und $k = \mathrm{ar}(f)$ gilt und $t_1, \ldots, t_k$ σ-Terme sind, dann ist auch $ft_1 \ldots t_k$ ein σ-Term.*

Lemma 2.6 (Eindeutige Lesbarkeit von Termen) *Jeder σ-Term ist entweder eine Variable oder lässt sich in der Form $ft_1 \ldots t_k$ schreiben, wobei $f \in \sigma^{\mathrm{Op}}$ ein k-stelliges Operationssymbol ist und $t_1, \ldots, t_k$ σ-Terme sind. Im zweiten Fall sind k, f und $t_1, \ldots, t_k$ eindeutig bestimmt.*

Beweis Die Menge aller Zeichenketten, die sich überhaupt in der beschriebenen Form schreiben lassen, erfüllt die Bedingungen aus der Definition der σ-Terme und umfasst daher die Menge der σ-Terme. Es bleibt zu zeigen, dass die Darstellung eindeutig ist. Seien also $ft_1 \ldots t_k = f't'_1 \ldots t'_{k'}$ zwei solche Darstellungen. Zunächst ist klar, dass $f = f'$ und $k = k'$ sein muss. Nach dem folgenden Lemma kann es aber kein kleinstes i geben, so dass $t_i \neq t'_i$ ist. ∎

Lemma 2.7 *Kein σ-Term ist echtes Anfangsstück eines anderen σ-Terms.*

Beweis Sei s ein Anfangsstück von t. Durch Induktion über die Länge von t zeigen wir, dass $s = t$ ist. Wenn s eine Variable ist, dann muss t ebenfalls diese Variable sein, also $s = t$. Sonst ist $s = fs_1 \ldots s_k$ und $t = ft_1 \ldots t_k$, wobei f ein k-stelliges Operationssymbol ist. Wenn $s \neq t$ wäre, dann gäbe es ein kleinstes i, so dass $s_i \neq t_i$, und für dieses wäre s_i ein echtes Anfangsstück von t_i oder umgekehrt. Nach Induktionsvoraussetzung ist das unmöglich. ∎

Definition 2.8 *Eine* Belegung *der Variablen in einer Struktur M ist eine Abbildung $\beta\colon \mathbb{X} \to \underline{M}$. Wenn σ die Signatur von M ist, setzen wir eine Belegung β in M wie folgt rekursiv zu einer Abbildung $\bar{\beta}\colon \{t \mid t \text{ ist } \sigma\text{-Term}\} \to \underline{M}$ fort:*

- $\bar{\beta}(\mathsf{x}^{\mathsf{0}}) = \beta(\mathsf{x}^{\mathsf{0}})$, $\bar{\beta}(\mathsf{x}^{\mathsf{1}}) = \beta(\mathsf{x}^{\mathsf{1}})$, $\bar{\beta}(\mathsf{x}^{\mathsf{2}}) = \beta(\mathsf{x}^{\mathsf{2}})$, *usw.*
- $\bar{\beta}(ft_1 \ldots t_k) = f^M(\bar{\beta}(t_1), \ldots, \bar{\beta}(t_k))$.

[1] Eine Sprache über einem Alphabet A (d.h. einer Menge A, aufgefasst als Menge von Symbolen) ist eine beliebige Menge $L \subseteq A^* = \bigcup_{k \in \mathbb{N}} A^k$ von Wörtern über A, d.h. von endlichen Tupeln mit Werten in A.

Definition 2.9 (Substitution) *Seien σ eine funktionale Signatur, s, t σ-Terme und $x \in \mathbb{X}$ eine Variable. Dann ist $t[\frac{s}{x}]$ die Zeichenkette, die man erhält, indem man jedes Vorkommen von x durch s ersetzt.*

Seien σ eine funktionale Signatur, t ein σ-Term, $x \in \mathbb{X}$ eine Variable, M eine σ-Struktur, $a \in \underline{M}$ und $\beta\colon \mathbb{X} \to \underline{M}$ eine Belegung in M. Dann ist $\beta[\frac{a}{x}]$ die Belegung

$$\beta(u) = \begin{cases} a & \text{falls } u = x \\ \beta(u) & \text{sonst.} \end{cases}$$

Man zeigt leicht durch Induktion über t, dass auch $t[\frac{s}{x}]$ wieder ein σ-Term ist.

Lemma 2.10 (Substitutionslemma für Terme) *Seien σ eine funktionale Signatur, s, t σ-Terme, $x \in \mathbb{X}$ eine Variable, M eine σ-Struktur und $\beta\colon \mathbb{X} \to \underline{M}$ eine Belegung. Dann gilt $\bar{\beta}(t[\frac{s}{x}]) = \bar{\beta}[\frac{\bar{\beta}(s)}{x}](t)$.*

Beweis Beweis durch Induktion über t. Wenn t eine Variable ist, ist die Behauptung klar. Wenn $t = ft_1 \dots t_k$ ist, ist

$$\begin{aligned} \bar{\beta}((ft_1 \dots t_k)[\frac{s}{x}]) = \bar{\beta}(ft_1[\frac{s}{x}] \dots t_k[\frac{s}{x}]) &= f^M(\bar{\beta}(t_1[\frac{s}{x}]), \dots, \bar{\beta}(t_k[\frac{s}{x}])) \\ &= f^M(\bar{\beta}[\frac{\bar{\beta}(s)}{x}](t_1), \dots, \bar{\beta}[\frac{\bar{\beta}(s)}{x}](t_k)) = \bar{\beta}[\frac{\bar{\beta}(s)}{x}](t), \end{aligned}$$

wobei die Gleichung zwischen der ersten und zweiten Zeile aus der Induktionsvoraussetzung folgt. ∎

2.2 Tarskis Definition der Wahrheit

Wir nehmen stillschweigend an, dass die speziellen Symbole $=$, $\exists$, $\neg$, $\wedge$, die wir im Folgenden benutzen werden, ebenso wie Variablennamen nie in einer Signatur auftreten. (Sollte das doch einmal geschehen, müssten wir zwischen dem Symbol als Teil der Signatur und dem Symbol der Prädikatenlogik unterscheiden wie bei disjunkten Vereinigungen.)

Definition 2.11 *Eine* Signatur σ *besteht aus einer Menge* σ^{Op} *von* Operationssymbolen *(auch* Funktionssymbole *genannt), einer Menge* σ^{Rel} *von* Relationssymbolen *(mit* $\sigma^{\mathrm{Op}} \cap \sigma^{\mathrm{Rel}} = \emptyset$*), sowie einer Abbildung* $\mathrm{ar}\colon \sigma^{\mathrm{Op}} \cup \sigma^{\mathrm{Rel}} \to \mathbb{N}$*, die jedem Symbol seine* Stelligkeit *zuordnet.*

Eine σ*-Struktur* M *besteht aus einer Grundmenge* $\underline{M} \neq \emptyset$ *sowie einer Funktion* $f^M\colon \underline{M}^k \to \underline{M}$ *für jedes* k*-stellige Operationssymbol* $f \in \sigma^{\mathrm{Op}}$ *und einer Teilmenge* $R^M \subseteq \underline{M}^k$ *für jedes* k*-stellige Relationssymbol* $R \in \sigma^{\mathrm{Rel}}$*.*

Gegenüber den funktionalen Signaturen und Strukturen aus dem letzten Abschnitt sind hier nur die Relationssymbole dazugekommen. Dadurch können wir beispielsweise eine lineare Ordnung in natürlicher Weise als Bestandteil einer Struktur auffassen.

Funktionale Signaturen und funktionale Strukturen können wir jetzt als den Spezialfall der allgemeinen Signaturen und Strukturen mit $\sigma^{\mathrm{Rel}} = \emptyset$ auffassen. Aus einer beliebigen Signatur oder Struktur erhalten wir eine funktionale, indem wir die Relationssymbole vergessen. Insbesondere werden wir auch für nicht funktionale Signaturen σ von σ-Termen sprechen. Die Relationssymbole sind im Zusammenhang mit Termen bedeutungslos, d.h. ein σ-Term ist dasselbe wie ein $(\sigma^{\mathrm{Op}}, \mathrm{ar}|_{\sigma^{\mathrm{Op}}})$-Term, und Relationssymbole können in Termen nicht vorkommen.

Definition 2.12 *Die Menge der* σ*-Formeln (der Prädikatenlogik 1. Stufe) ist die kleinste Sprache über dem Alphabet* $\sigma^{\mathrm{Op}} \cup \sigma^{\mathrm{Rel}} \cup \{=, \exists, \neg, \wedge\} \cup \mathbb{X}$*, welche die folgenden Bedingungen erfüllt.*

- *Wenn* t_1 *und* t_2 σ*-Terme sind, dann ist* $=t_1t_2$ *eine* σ*-Formel.*
- *Wenn* $R \in \sigma^{\mathrm{Rel}}$ *und* $k = \mathrm{ar}(R)$*, und* $t_1, \ldots, t_k$ *sind* σ*-Terme, dann ist* $Rt_1 \ldots t_k$ *eine* σ*-Formel.*
- *Wenn* φ *eine* σ*-Formel ist, dann ist auch* $\neg\varphi$ *eine* σ*-Formel.*
- *Wenn* φ *und* ψ σ*-Formeln sind, dann ist auch* $\wedge\varphi\psi$ *eine* σ*-Formel.*
- *Wenn* φ *eine* σ*-Formel ist und* $x \in \mathbb{X}$ *eine Variable, dann ist auch* $\exists x\varphi$ *eine* σ*-Formel.*

Lemma 2.13 (Eindeutige Lesbarkeit von Formeln) *Jede* σ*-Formel lässt sich in genau einer der folgenden Weisen zerlegen:*

- *Als* $=t_1t_2$*, wobei* t_1, t_2 σ*-Terme sind.*
- *Als* $Rt_1 \ldots t_k$*, wobei* $R \in \sigma^{\mathrm{Rel}}$ *und* $k = \mathrm{ar}(R)$*, und* $t_1, \ldots, t_k$ *sind* σ*-Terme.*
- *Als* $\neg\varphi$ *für eine* σ*-Formel* φ*.*
- *Als* $\wedge\varphi\psi$ *für* σ*-Formeln* φ *und* ψ*.*
- *Als* $\exists x\varphi$ *für eine* σ*-Formel* φ *und eine Variable* x*.*

Die Zerlegung ist außerdem eindeutig, d.h. t_1, t_2*, bzw.* R *und* $t_1, \ldots, t_k$*, bzw.* φ *bzw.* φ, ψ *bzw.* x *und* φ *sind eindeutig bestimmt.*

Beweis Die Menge aller Zeichenketten, die sich überhaupt in der beschriebenen Form schreiben lassen, erfüllt die Bedingungen aus der Definition der σ-Formeln und umfasst daher die Menge der σ-Formeln. Es bleibt zu zeigen, dass die Darstellung eindeutig ist. Zunächst sehen wir, dass die erste Zerlegung nur möglich ist, wenn das erste Symbol der Formel $=$ ist, die zweite nur, wenn das erste Symbol ein Relationssymbol ist, usw. Daher lässt sich jede Formel nur gemäß höchstens (also genau) einem der Unterpunkte zerlegen.

Im Falle eines der ersten beiden Unterpunkte können wir Lemma 2.7 anwenden und genauso argumentieren wie im Beweis von Lemma 2.6: Wenn $=t_1t_2 = =t_1't_2'$ ist, dann ist t_1 ein Anfangsstück von t_1' oder t_1' ein Anfangsstück von t_1. In beiden Fällen folgt $t_1 = t_1'$ und dann auch $t_2 = t_2'$. Wenn $Rt_1 \ldots t_k = R't_1' \ldots t_k'$ ist, dann ist natürlich $R = R'$. Also ist t_1 ein Anfangsstück von t_1' oder t_1' ein Anfangsstück von t_1, woraus $t_1 = t_1'$ folgt. Folglich ist t_2 ein Anfangsstück von t_2' oder t_2' ein Anfangsstück von t_2, woraus $t_2 = t_2'$ folgt, usw.

Im Falle eines der letzten drei Unterpunkte wenden wir stattdessen Lemma 2.14 an und argumentieren wieder analog. ∎

Lemma 2.14 *Keine σ-Formel ist echtes Anfangsstück einer anderen σ-Formel.*

Beweis Sei φ ein Anfangsstück von ψ. Durch Induktion über die Länge von ψ (des längeren Stücks) zeigen wir, dass $\varphi = \psi$ ist.

Wenn $\varphi = {=}t_1t_2$ ist, dann beginnt auch ψ mit $=$, und folglich ist $\psi = {=}t_1't_2'$. Es ist klar, dass t_1 ein Anfangsstück von t_1' ist oder t_1' ein Anfangsstück von t_1. Nach Lemma 2.7 ist $t_1 = t_1'$. Folglich ist t_2 ein Anfangsstück von t_2' oder t_2' ein Anfangsstück von t_2. Wiederum mit Lemma 2.7 folgt $t_2 = t_2'$. Der Fall $\varphi = Rt_1 \dots t_k$ geht ganz ähnlich.

Wenn $\varphi = \wedge\varphi_1\varphi_2$ ist, dann beginnt auch ψ mit $\wedge$, und folglich ist $\psi = \wedge\psi_1\psi_2$. Es ist klar, dass φ_1 ein Anfangsstück von ψ_1 ist oder ψ_1 ein Anfangsstück von φ_1. Nach Induktionsvoraussetzung ist daher $\varphi_1 = \psi_1$. Folglich ist φ_2 ein Anfangsstück von ψ_2 oder ψ_2 ein Anfangsstück von φ_2. Wiederum nach Induktionsvoraussetzung ist daher $\varphi_2 = \psi_2$. Die Fälle $\varphi = \neg\psi$ und $\varphi = \exists x\psi$ gehen ganz ähnlich. ∎

Unter einer Belegung der Variablen in einer Struktur entspricht jedem Term ein Element (der Grundmenge) der Struktur. Diesen Zusammenhang werden wir jetzt auf Formeln erweitern: Unter einer Belegung der Variablen in einer Struktur entspricht jeder Formel ein Wahrheitswert, d.h. wahr (1) oder falsch (0). In gewissem Sinne können wir „Wahrheit“ also formal definieren:

Definition 2.15 (Tarskis Definition der Wahrheit) [2] *Wenn σ die Signatur von M ist, definieren wir für jede Belegung β in M wie folgt rekursiv eine Abbildung $\hat{\beta}\colon \{\varphi \mid \varphi \text{ ist } \sigma\text{-Formel}\} \to \{0,1\}$:*

- $\hat{\beta}({=}t_1t_2) = \begin{cases} 1 & \text{falls } \bar{\beta}(t_1) = \bar{\beta}(t_2) \\ 0 & \text{sonst.} \end{cases}$
- $\hat{\beta}(Rt_1 \dots t_k) = \begin{cases} 1 & \text{falls } (\bar{\beta}(t_1), \dots, \bar{\beta}(t_k)) \in R^M \\ 0 & \text{sonst.} \end{cases}$
- $\hat{\beta}(\neg\varphi) = 1 - \hat{\beta}(\varphi)$.
- $\hat{\beta}(\wedge\varphi\psi) = \hat{\beta}(\varphi) \cdot \hat{\beta}(\psi)$.
- $\hat{\beta}(\exists x\varphi) = \begin{cases} 1 & \text{falls es ein Element } e \in \underline{M} \text{ gibt, so dass } \widehat{\beta[\frac{e}{x}]}(\varphi) = 1 \\ 0 & \text{sonst,} \end{cases}$
  wobei $\beta[\frac{e}{x}]$ (siehe Definition 2.9) sich von β nur durch $\beta(x) = e$ unterscheidet.

Dass $\hat{\beta}$ wohldefiniert ist, folgt aus der eindeutigen Lesbarkeit.

Die Belegungen sind ein notwendiges technisches Hilfsmittel. Nachdem wir mit ihrer Hilfe die Wahrheit definiert haben, können wir zeigen, dass es in bestimmten Fällen nicht auf sie ankommt. Dazu brauchen wir den Begriff eines σ-Satzes, d.h. einer σ-Formel ohne freie Variable.

Definition 2.16 *Die Menge der Variablen, die in einer σ-Formel frei vorkommen, ist wie folgt rekursiv definiert.*

- $\mathrm{Frei}({=}t_1t_2)$ *ist die Menge aller Variablen, die in t_1 oder t_2 vorkommen.*
- $\mathrm{Frei}(Rt_1 \dots t_k)$ *ist die Menge aller Variablen, die in $t_1, t_2, \dots$ oder t_k vorkommen.*
- $\mathrm{Frei}(\neg\varphi) = \mathrm{Frei}(\varphi)$.
- $\mathrm{Frei}(\wedge\varphi\psi) = \mathrm{Frei}(\varphi) \cup \mathrm{Frei}(\psi)$.
- $\mathrm{Frei}(\exists x\varphi) = \mathrm{Frei}(\varphi) \setminus \{x\}$.

Lemma 2.17 *Sei σ eine Signatur, φ eine σ-Formel und M eine Struktur der Signatur σ. Seien $\beta_1, \beta_2\colon \mathbb{X} \to \underline{M}$ zwei Belegungen der Variablen in M. Falls β_1 und β_2 auf $\mathrm{Frei}(\varphi)$ übereinstimmen (d.h. falls $\beta_1(x) = \beta_2(x)$ für alle $x \in \mathrm{Frei}(\varphi)$), dann ist $\hat{\beta}_1(\varphi) = \hat{\beta}_2(\varphi)$.*

[2] Alfred Tarski (1901–1983) bezeichnete sich selbst mit einiger Berechtigung als “the greatest living sane logician”. (Der größte lebende Logiker war damals eindeutig Kurt Gödel.) Tarski war ein Workaholic und Frauenheld. Zum Zeitpunkt des deutschen Überfalls auf Polen 1939 war er auf einem Kongress an der Harvard-Universität. Seine in Warschau zurückgelassene Familie sah er erst 1946 wieder; ein großer Teil seiner weiteren Verwandtschaft überlebte nicht.

Beweis Durch Induktion über den Aufbau von φ. Wirklich interessant ist nur der Fall, in dem ein Quantor eingeführt wird. ∎

Definition 2.18 *Ein σ-Satz ist eine σ-Formel φ mit* $\mathrm{Frei}(\varphi) = \emptyset$. *Wir sagen, dass ein σ-Satz φ in einer σ-Struktur M gilt, oder auch, dass M ein* Modell von φ *ist, wenn für eine (und damit für jede) Belegung β der Variablen in M gilt: $\hat{\beta}(\varphi) = 1$. Die übliche Notation für diese Beziehung ist $M \models \varphi$.*

2.3 Gödelisierung

Definition 2.19 *Die (abzählbare)* Universalsignatur σ_U *ist gegeben durch* $\sigma_U^{\mathrm{Op}} = \{f_n^k \mid n,k \in \mathbb{N}\}$, $\sigma_U^{\mathrm{Rel}} = \{R_n^k \mid n,k \in \mathbb{N}\}$ *und* $\mathrm{ar}_{\sigma_U}(f_n^k) = \mathrm{ar}_{\sigma_U}(R_n^k) = k$.

Der Vorteil von σ_U: Nach etwaigem Umbenennen der Symbole können wir jede endliche (oder sogar abzählbare) Signatur σ als Teilsignatur von σ_U auffassen, d.h. $\sigma^{\mathrm{Op}} \subseteq \sigma_U^{\mathrm{Op}}$, $\sigma^{\mathrm{Rel}} \subseteq \sigma_U^{\mathrm{Rel}}$ und $\mathrm{ar} = \mathrm{ar}_U|_{\sigma^{\mathrm{Op}} \cup \sigma^{\mathrm{Rel}}}$. Da in jeder Formel nur endlich viele Symbole vorkommen, ist also jede Formel bis auf Umbenennen von Symbolen eine σ_U-Formel.[3]

Definition 2.20 *Wie legen die folgende Aufzählung* s_U *des Alphabets* $A_U = \sigma_U^{\mathrm{Op}} \cup \sigma_U^{\mathrm{Rel}} \cup \{=, \exists, \neg, \wedge\} \cup \mathbb{X}$ *als* Standardaufzählung *fest:*

$$
\begin{aligned}
s_U(0) &= = & s_U(3m+4) &= \mathsf{x}^{\mathsf{m}} \\
s_U(1) &= \exists & s_U(3m+5) &= f_{L(m)}^{R(m)} \\
s_U(2) &= \neg & s_U(3m+6) &= R_{L(m)}^{R(m)}. \\
s_U(3) &= \wedge
\end{aligned}
$$

Dabei sind L *und* R *die Komponenten der Umkehrung der Cantorschen Paarungsfunktion (siehe Satz 1.11). Dadurch ist für jedes Symbol in* A_U *eine natürliche Zahl festgelegt, die dieses eindeutig codiert.*

Jedes Wort $w = w_0 w_1 \dots w_{k-1} \in A_U^* = \bigcup_{\ell \in \mathbb{N}} A_U^\ell$ *wird durch seine* Gödelnummer

$$\langle w \rangle := \langle s_U^{-1}(w_0), s_U^{-1}(w_1), \dots, s_U^{-1}(w_{k-1}) \rangle$$

codiert. (Für die Definition von $\langle\rangle$ *siehe Definition 1.15.)*

Satz 2.21 *Die Mengen* $\{\langle t \rangle \mid t \text{ ist } \sigma_U\text{-Term}\}$ *und* $\{\langle \varphi \rangle \mid \varphi \text{ ist } \sigma_U\text{-Formel}\}$ *sind primitiv rekursiv.*

Beweis Wir zeigen das nur für die erste Menge. Für die zweite Menge geht der Beweis dann ähnlich. Wir müssen zeigen, dass die Funktion $\chi_{\{\langle t \rangle \mid t \text{ ist } \sigma_U\text{-Term}\}}(x)$ primitiv rekursiv ist. Dazu skizzieren wir ein LOOP-Programm, das die Funktion berechnet. Das Programm setzt die Ausgabe zunächst auf 1 (d.h. Erfolg: es handelt sich um einen Term) und geht dann von links nach rechts durch das Wort durch, um zu überprüfen, dass es sich wirklich um einen Term handelt. Falls dabei irgendwann ein Problem auftritt, setzt es die Ausgabe auf 0.(Da die Ausgabe später niemals wieder auf einen von 0 verschiedenen Wert gesetzt wird, müssen wir uns in diesem Fall nicht darum kümmern, was danach geschieht.) Am Anfang müssen wir genau einen Term lesen, also hat `termsleft` den Wert 1. Wenn wir dann z.B. ein zweistelliges Relationssymbol gelesen haben, haben wir einen der noch ausstehenden Terme zu lesen angefangen (`termsleft` verringert sich um 1), müssen dafür aber zwei weitere Terme lesen (`termsleft` erhöht sich um 2). Insgesamt erhöht sich daher `termsleft` beim Lesen eines k-stelligen Relationssymbols um $k-1$.

```
w = input
output = 1
termsleft = 1
loop Lg(w) times{
  if not(termsleft) then {
    output = 0
  }
  arity = ...
  w = ...
  termsleft = termsleft - 1 + arity
}
if termsleft then {
  output = 1
}
```

[3] Wenn man allerdings überabzählbar viele Formeln hat, kann es sein, dass man die Umbenennungen nicht kohärent durchführen kann.

Wir durchlaufen die Schleife so oft, wie das Wort lang ist. In jedem Durchlauf überprüfen wir zuerst, dass wir noch nicht am Ende des Terms angekommen sind (d.h. dass `termsleft` größer als 0 ist) und melden sonst einen Fehler. Die Zeile `arity = ...` deutet an, dass wir das erste Zeichen untersuchen und, wenn es sich um ein k-stelliges Funktionssymbol oder um eine Variable handelt (die wir als gleichwertig mit einem 1-stelligen Funktionssymbol behandeln), die Variable `arity` auf k setzen. Andernfalls (und das ist im Listing nicht zu sehen) melden wir durch `output = 0` einen Fehler. Die Zeile `w = ...` entfernt von w, aufgefasst als ein Tupel von natürlichen Zahlen, das erste Element. (Dazu muss man sich eigentlich erst noch überlegen, dass das mit einer primitiv rekursiven Funktion geht.) Nach dem letzten Durchlauf, d.h., wenn das Wort vorbei ist, überprüfen wir noch, dass wir wirklich am Ende des Terms angekommen sind. Falls nicht, melden wir einen Fehler.

Die zweite Menge kann man ganz ähnlich behandeln. Allerdings muss man dazu zuerst noch das obige Programm so umschreiben, dass es überprüft, ob ein Wort mit einem Term beginnt und ggf. den Rest des Wortes zurückgibt. ∎

Jetzt sind wir also soweit, dass wir die Syntax (Formeln, Terme, Sätze usw.) mit Computern behandeln können. Unsere Motivation ist natürlich letztlich, dass wir etwas über die Semantik (mathematische Strukturen) aussagen wollen. Dazu sind Definitionen wie die folgende hilfreich:

Definition 2.22 *Eine σ-Formel φ heißt* allgemeingültig*, falls für alle σ-Strukturen M und alle Belegungen $\beta\colon \mathbb{X} \to \underline{M}$ der Variablen in M gilt: $\hat{\beta}(\varphi) = 1$.*

Ein nichttriviales Beispiel für eine allgemeingültige σ-Formel (bei beliebigem σ) ist jede Formel der Gestalt $(x = y \wedge y = z) \to x = z$, oder genauer $\neg\wedge\wedge{=}xy{=}yz\neg{=}xz$ für beliebige Variable $x, y, z \in \mathbb{X}$. Also konkret z.B. $\neg\wedge\wedge{=}\overset{01}{X}X{=}\overset{12}{X}X\neg{=}\overset{02}{X}X$.

Wir können also jetzt die Frage stellen, ob die Menge $\{\langle\varphi\rangle | \varphi$ allgemeingültige σ_U-Formel$\}$ primitiv rekursiv ist. Diese Frage werden wir in Kapitel 4 negativ beantworten. Genauer werden wir sogar zeigen, dass es grundsätzlich kein Computerprogramm geben kann, das für jede natürliche Zahl entscheidet, ob sie in der Menge ist oder nicht.

Aussagenlogik

Um das Problem etwas zu vereinfachen, schauen wir uns noch kurz die Aussagenlogik an.

Definition 2.23 *Eine* aussagenlogische Signatur *ist eine Signatur σ, so dass $\sigma^{\mathrm{Op}} = \emptyset$ und $\mathrm{ar}(R) = 0$ für alle $R \in \sigma^{\mathrm{Op}}$. Ein* aussagenlogischer Satz *ist ein σ-Satz für eine aussagenlogische Signatur σ, in dem keine Variablen oder Gleichheitszeichen vorkommen. Ein allgemeingültiger aussagenlogischer Satz heißt* Tautologie*.*

Analog zu Lemma 2.17 kann man beweisen:

Bemerkung 2.24 *Sei σ eine aussagenlogische Signatur, φ ein aussagenlogischer σ-Satz und M, M' zwei σ-Strukturen, so dass $R^M = R^{M'}$ für alle $R \in \sigma^{\mathrm{Rel}}$. Dann gilt $M \models \varphi \iff M' \models \varphi$.*

Die Elemente einer Struktur spielen für die Gültigkeit aussagenlogischer Sätze also überhaupt keine Rolle.

Da σ_U aussagenlogische Teilsignaturen hat, gibt es auch aussagenlogische σ_U-Sätze.

Satz 2.25 *Die Menge $\{\langle\varphi\rangle \mid \varphi$ ist (eine σ_U-Formel und) Tautologie$\}$ ist primitiv rekursiv.*

Beweis Beweisidee: Wir können für jeden möglichen Wahrheitswert der nullstelligen Relationssymbole, die in dem Satz wirklich vorkommen, überprüfen, ob der Satz gilt. Es sind genau 2^n Fälle zu untersuchen, wenn n die Anzahl der verschiedenen Relationssymbole im Satz ist. In jedem einzelnen Fall gehen wir von rechts nach links durch den Satz durch und merken uns eine endliche Folge von Wahrheitswerten (codiert als einzelne Zahl). Jede nullstellige Relation hängt ihren Wahrheitswert hinten an. Jedes $\neg$ negiert den aktuell letzten Wert. Bei jedem werden die beiden letzten Werte entfernt und durch deren Konjunktion ersetzt. Bei einer gültigen Formel haben wir am Ende einen einzigen Wahrheitswert in der endlichen Folge: Den Wahrheitswert der Folge. ∎

Es gibt keine universelle primitiv rekursive Menge

Gödelisierung kann man auch auf LOOP-Programme statt auf Formeln anwenden. In diesem Fall wählen wir eine Aufzählung des LOOP-Alphabets

$$A = \{\mathtt{a}, \ldots, \mathtt{z}, \mathtt{A}, \ldots, \mathtt{Z}, \mathtt{0}, \ldots, \mathtt{9}, \mathtt{(}, \mathtt{)}, \{, \}, \mathtt{,}, \mathtt{=}, \_, \text{Leerzeichen}\}.$$

In derselben Weise wie bei Formeln können wir dann jedem Wort über A einen Gödelcode zuordnen.

Als Vorgeschmack auf das nächste Kapitel schauen wir uns nun die folgende Menge an:

$$PR = \{(p, x) \in \mathbb{N}^2 \mid p \text{ ist Gödelcode eines LOOP-Programms, das bei Eingabe } (x, 0, 0, \ldots) \text{ den Wert 1 ausgibt}\}.$$

Proposition 2.26 *PR ist nicht primitiv rekursiv.*

Beweis Wenn PR primitiv rekursiv wäre, gäbe es ein LOOP-Programm, das die charakteristische Funktion von PR berechnet. Dann gäbe es auch ein LOOP-Programm, das bei der Eingabe $(x, 0, 0, \ldots)$ genau dann 0 ausgibt, wenn $(x, x) \in PR$ gilt. Sei p seine Gödelnummer. Nach Wahl des Programms würde dann gelten:

$$(p, p) \in PR \iff \text{Das durch } p \text{ codierte Programm gibt bei Eingabe } p \text{ den Wert 0 aus.}$$

Nach Definition von PR würde aber andererseits gelten:

$$(p, p) \in PR \iff \text{Das durch } p \text{ codierte Programm gibt bei Eingabe } p \text{ den Wert 1 aus.}$$

∎