Definitions-Zusammenfassung: Logik und diskrete Strukturen

David Kriesel / dkriesel.com

11. Februar 2012

Das ist eine Definitions-Zusammenfassung, die ich für mich selbst und für meine Studis schreibe, um den Stoff für meine Tutorien anschaulich aufzubereiten. Es gibt keinen Anspruch auf Korrektheit oder Vollständigkeit, am Ende zählt immer das gesprochene/geschriebene Wort in der Veranstaltung selbst. Wer sich aber schon mehrfach im Tutorium und über die Zettel mit dem Stoff auseinandersetzen musste, wird in dieser Kurz-Zusammenfassung seinen Nutzen finden. Im Vorlesungsmaterial finden sich jede Menge Beispiele, ich werde hier also nicht groß weitere suchen. Dafür werde ich aber – und das mutmaßlich das wichtige für **euch als Erstis** – die Definitionen mit ein bisschen Umgangssprache garnieren. In der Zusammenfassung geäußerte Meinungen sind von mir persönlich, rühren von Jahren an universitärer Lehrerfahrung her und müssen nicht mit der Meinung anderer übereinstimmen.

Inhaltsverzeichnis

Voraussetzungen, erster Teil

Wer die Zusammenfassung liest, sollte vorher ein paar Voraussetzungen verinnerlicht haben. Verinnerlichen bedeutet, dass ich euch nachts um drei Uhr wecken kann[1], und ihr diese Sachen parat habt.

Die Sachen, die ich hier als Voraussetzung beschreibe, kommen nicht nur in der LuDS-Vorlesung vor. Sie wurden sich auch nicht von bösen Dozenten ausgedacht. Sie sind die Grundlage für euer gesamtes Informatikstudium, jede Vorlesung und alles was ihr selbst schreibt wird diese Voraussetzungen beinhalten. **Was ich hier in den Voraussetzungen aufliste, ist die Sprache der Informatik.** Wer die Symbole nicht anwenden kann oder z.B. keine Mengen anständig aufschreiben kann, wird es ab der dritten Woche des ersten Semesters unschaffbar schwer haben **und ein Bestehen des Informatikstudiums wird ganz einfach unmöglich sein**. Das wäre wie wenn jemand, der chinesisch studieren will, die Schriftzeichen nicht lernen würde – oder ein LKW-Fahrer, der sich weigert, ein Schild mit der Aufschrift „Maximalhöhe: 3.7 Meter“ zu lesen. Der LKW-Fahrer hat in dem Fall einen Cabrio-LKW und ein paar hunderttausend Euro Schaden (und wenn es ein Gaslaster war, hat er auch nie wieder Schmerzen). Ihr fallt eben stattdessen durchs Studium.

Man könnte auch eine dieser Sheep-World-Postkarten daraus machen. *Ohne Fachsprache ist alles doof. Logik: Doof. Mathe: Doof. Programmieren: Doof. Informatik: Doof.*[2]

Ich werde die Voraussetzungen hier nicht alle definieren, sie stehen unheimlich gut im Vorlesungsmaterial. Ich liste sie hier nur auf, im Zweifel könnt ihr sie dann selbst nachschlagen.

- Mengen im Allgemeinen. Schreibweisen, verschiedene standardisiert definierte Mengen ($\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$, usw.). Anzahl der Elemente in einer Menge M: $|M|$.
- Saubere zusammenfassende Schreibweisen von Mengen. $\{x|x \in \mathbb{N} \land x < 5\}$ zum Beispiel (es gibt da ein paar Varianten, die ebenfalls im Vorlesungsmaterial vorkommen, z.B. $\{x \in \mathbb{N}|x < 5\}$, gewöhnt euch gerne einen eigenen „Geschmack“ an). **Ganz im Ernst: Gerade das ist die absoluteste aller Grundlagen.** Wenn ich Leute höheren Semesters sehe, die es nicht beherrschen, eine Mengendefinition dieser Art aufzuschreiben, frage ich mich immer, was die die ganze Zeit gemacht haben, denn die braucht man wirklich für *alles*.
- Tupel. Schreibweise wie Mengen, nur mit runden Klammern. Unterschiede zu Mengen: Es können Elemente mehrfach vorkommen, und die Ordnung ist nicht egal. Werden gerne für strukturierte Definitionen Benutzt. Man sagt, ein Gegenstand G sei definiert als ein Tupel (X, Y, Z) und danach definiert man x, y, z runter.
- Verschiedene Operatoren und Abkürzungszeichen sollten ebenfalls zu 100% und ohne jemals wieder darüber nachzudenken sitzen: $\in, \notin, \forall, \exists, \nexists, \Rightarrow, \Leftrightarrow, \subset, \subseteq, =, \subsetneq, \emptyset, \pm, \setminus, \cup, \cap, \times, \vee, \wedge$, so wie die Zusammenfassungschreibweisen, die es für viele Operatoren gibt, z.b. $\sum_{i=1}^{10} i^2$. Diese Zusammenfassungsschreibweisen gibt es für sehr viele der Operatoren und man muss sie kennen. Es gibt sie nicht nur in der Schreibweise mit Variablendefinition und Startwert unten sowie Ziel oben, sondern auch in der Indexmengenschreibweise $\sum_{i \in M} i^2, M = \{1, 5, 7, 3, 4\}$ oder sogar noch kürzer $\sum_M i^2, M = \{1, 5, 7, 3, 4\}$. Hier werden die i nicht hochgezählt, sondern aus der Indexmenge M genommen (kann auch anders als M heißen). Denjenigen Zusammenfassungsoperator, den ihr mit Abstand am häufigsten sehen werdet, ist der für die Summe, wie eben gezeigt.
- Sinnvolles anwenden der o.g. Operatoren, man sollte wissen was für Daten in einen Operator hineingegeben werden, und was man herausbekommt, dazu gleich noch mehr.
- Ihr solltet wissen, was die einzelnen Gliederungselemente eines naturwissenschaftlichen Textes sind, im Besonderen was Definitionen, Lemmata, Sätze (werden auch Theoreme genannt), Beweise und Korollare sind. In einer Definition wird etwas definiert, was man im Folgetext voraussetzen kann. Ein Lemma ist eine Teilaussage in einem größeren Kontext, die meist einen Beweis erfordert. Ein Theorem bzw. ein Satz ist eine große, wichtige Aussage, die einen Beweis erfordert. Zum Beispiel kann man mit ein paar Lemmata ein Theorem vorbereiten. Ein Korollar kann man nach einem Beweis schreiben, für irgendetwas, was man durch den Beweis noch „nebenbei geschenkt kriegt“, bzw. was direkt und offensichtlich folgt.

[1] oder aus der Wache herausrufen kann, und wer nicht weiß, was die Wache ist, der wird sich jetzt danach erkundigen...
[2] Diese Anmerkung war eine kleine Hommage an eine Person, die das schon bemerken wird, wenn sie es liest. :-)

Voraussetzungen, zweiter Teil (Kompaktkurse in meinen Tutorien)

Ihr hattet bis jetzt zwei „Kompaktkurse“ bei mir im Tutorium. Der eine ging zum Thema „Wie beweise ich richtig?“ und der andere zum Thema „Operatorenlehre“.

Im Beweis-Crashkurs hatten wir ergründet, dass Beweise keine magischen Dinge sind, von denen euer Tutor willkürlich entscheidet, ob es auch wirklich Beweise sind oder nicht, und dann die Punkte verteilt.

1. Beweise sind aufeinander aufbauende Aussagen bzw. Umformungen.
2. Jede neue Umformung muss solide logisch, z.B. anhand einer Definition *begründbar* sein und zwar interpretationsfrei!

Wir haben also tabellenförmige Beweise behandelt, in denen ihr nicht nur die aufeinander aufbauenden Aussagen in kleinstmöglichen Schritten aufschreibt, sondern auch die Begründungen für jede neue Aussage.

Im Operator-Crashkurs haben wir besprochen, wie die ganzen oben genannte Zeichen einzusetzen sind, so dass ihr auch erkennen könnt, ob eine mathematische Zeile Blödsinn ist oder nicht. In der Regel ist eine mathematische Zeile kein Blödsinn, wenn die Operatoren richtig angewandt werden, anders ausgedrückt, wenn ihr das richtige hineingebt und als Ausgabe entgegennehmt. Wir haben gelernt, dass es Operatoren in verschiedenen Welten gibt, und Operatoren, die Welten verbinden.

Mengen-Welt: Gebt Mengen ein, erhaltet Mengen. Operatoren $\cup, \cap, \times, \setminus$, Komplement etc. samt Zusammenfassungen.

Wahrheitswert-Welt: Gebt Wahrheitswerte ein, erhaltet Wahrheitswerte. Operatoren $\vee, \wedge, \Rightarrow, \Leftrightarrow$ und verschiedene Ausprägungen von „nicht“, die auch teilweise direkt als Kurzschreibweise in andere Operatoren reingezogen werden. Auch hier gibt es zusammenfassende Operatoren.

Verbindende Operatoren: Gebt Mengen oder anderes ein, erhaltet Wahrheitswerte! Operatoren $\in, \notin, \subset, \subseteq, =, \subsetneq$, etc.

Diese Operatoren bzw. ihre Definitionen könnt ihr beispielsweise nehmen, um Zeilen eines Beweises in die nächste Umzuformen. Nun, da ihr wisst, welche Operatoren was entgegennehmen und ausgeben, und wie man Mengen definiert, ist hoffentlich Schluss mit Stuss wie $\{x \in \mathbb{N}\} \vee y$, der einfach erkennbar ungültig ist, weil Dinge in Operatoren gesteckt werden, die dort einfach nicht hineinpassen.

1 Grundlegende Definitionen, aufbauend auf den Voraussetzungen: Kartesisches Produkt, Potenzmengen, Relationen, Abbildungen.

Ich werde ab jetzt alles, was in den Voraussetzungen steht, konsequent nutzen. Wir kommen jetzt zu unserer ersten richtigen Section. Sie definiert verschiedene Grundlagenkonstrukte, die nicht nur für die LuDS-Vorlesung gelten, sondern ebenfalls auf große Teile der Informatik Einfluß haben und euch über weite Teile eures Studiums begleiten werden. Damit endet der prosaische Teil dieser Zusammenfassung, und es gibt nur noch eine Definitionssammlung, die ich sowieso anlege um mich vorzubereiten. Teilweise schreibe ich noch ein zwei erklärende Sätze dazu. Für jede Definition: Seien A, B Mengen und $a \in A$ sowie $b \in B$ (so muss ich das nicht jedes mal neu hinschreiben).

Definition 1 (Potenzmenge)**.** Die Menge aller Teilmengen einer Menge B. $\mathcal{P}(B) = \{A | A \subseteq B\}$. Profitipp, um Anfängerfehler zu vermeiden: Auch $\emptyset$ sowie B selbst sind in $\mathcal{P}(B)$ enthalten. Hat $2^{|B|}$ Elemente.

Definition 2 (Kartesisches Produkt)**.** Definiert auf zwei Mengen. Operator, der zwei Mengen entgegennimmt und eine Menge von Tupeln zurückgibt, wobei das erste Element eines jeden Tupels aus der ersten Menge kommt und das zweite aus der Zweiten. Geschrieben mit dem Symbol $\times$:

$$A \times B = \{(a, b) | a \in A, b \in B\}.$$

Die Anzahl der Tupel im Kartesischen Produkt ergibt sich aus den Multiplizierten Anzahlen der Elemente in den Mengen: $|A \times B| = |A| \cdot |B|$. Nicht vergessen: Tupel sind geordnet, und Tupel sind keine

Mengen! Geht analog auch für mehr Mengen als zwei! Noch mal: Ein Kartesisches Produkt ist eine Menge von Tupeln. Wenn $I = \{1, \ldots, n\}$ eine Indexmenge ist ist die Kurzschreibweise für ein Kartesisches Produkt auch $(A_i)_{i \in I}$.

Definition 3 (Relationen)**.** Seien A und B und natürlich $A \times B$ wie gewohnt. Davon ist eine Relation R eine Teilmenge: $R \subseteq A \times B$. Man sagt, a und b stehen in Relation zueinander, wenn $(a, b) \in R$. Das ist unser mathematisches Werkzeug um Relationen zu modellieren, wie sie auch wirklich im Leben vorkommen, z.B. „ist größer als“. Mathematisch ist, wie das Kartesische Produkt ja auch, eine Relation eine Menge von Tupeln. Durch die formale Schreibweise kann man aber nun auch Relationen definieren, die man nicht mehr so einfach umgangssprachlich aufschreiben kann. Geht natürlich auch mit mehr Mengen als nur A und B. $R \subseteq A \times B$ ist eine Zweistellige Relation, analog gibt es Relationen mit mehr „Stellen“.

Definition 4 (Äquivalenzrelation)**.** Sehr oft betrachteter Spezialfall von Relationen. Also Definition wie bei den Relationen, aber mit ein paar Zusatzeigenschaften ($a, b, c \in R$):

1. $(a, a) \in R$ (Reflexivität, jedes Tupel mit 2x demselben Element ist in R)
2. $(a, b) \in R \Leftrightarrow (b, a) \in R$ (Symmetrie, auch „umgedrehte“ Tupel sind in R)
3. $(a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$ (Transitivität: zum Bespiel die Relation „hat am selben Tag Geburtstag“ ist Transitiv. Wenn a am selben Tag Geburtstag hat wie b, und b wie c, dann haben auch a und c am selben Tag Geburtstag.)

Definition 5 (Ordnung)**.** Todo

Definition 6 (Abbildungen, Definitionsbereich, Bildbereich, Rechtseindeutigkeit)**.** Schon wieder eine Menge von Tupeln! Abbildungen sind die Verallgemeinerung von Funktionen, die ihr aus der Schule kennt. Bei euch hat man Zahlen auf Zahlen abgebildet, wir bilden allgemein Mengen auf Mengen ab, egal ob die Mengen Zahlen enthalten, oder andere Mengen, Personen, Äpfel oder Birnen. Mathematisch sind Abbildungen aber ein *Spezialfall* von Relationen. Wieder sind sie definiert auf A und B. Eine Relation f auf A und B ist genau dann eine Abbildung, wenn zwei Dinge gelten (Die Anna-und-Bernd-Sprüche gehen auf das Konto meiner Tutoriengruppe 16).

1. $\forall a \in A \exists b \in B : (a, b) \in f$. Für jedes a aus A gibt es ein b aus B. Es gibt kein a, das ohne b ist. Durchaus darf es aber b geben, die kein a haben, oder auch b, die mehrere a haben!
2. $\forall a \in A \exists b, b' \in B : ((a, b) \in f \wedge (a, b') \in f \Rightarrow b = b')$. Wenn a in Relation zu b steht, UND a auch noch in Relation zu b' steht, so sind auf jeden Fall b und b' gleich. Ein a hat also nicht mehrere b, sondern immer nur eins.

Beides zusammen heisst „rechtseindeutig“. Wir sagen auch kurz $f(a) = b$ (das ist die Schreibweise, die ihr in der Informatikwelt am allermeisten sehen werdet). Tutorium 16 sagte zu den beiden Regeln „keine Anna ohne Bernd“. Manche Bernds haben aber vielleicht keine Anna, und umgekehrt gibt es vielleicht auch Bernds mit mehreren Annas. A ist die Menge, auf der f definiert ist und heisst daher auch Definitionsbereich oder auch $D(f)$. B ist die Menge, auf die f abbildet, und darum heisst B auch „Bildbereich“, $B(f)$, „Bild“ oder „Zielmenge“. Vergeßt nicht die Schreibweise der Abbildungen aus dem Script von Herrn Klein! Wie beim Programmieren heisst a auch „Argument“.

Definition 7 (Graph)**.** Wie wir schon gelernt haben, gibt es Bernds ohne Anna. Der Graph $G(f)$ einer Abbildung f sind alle *G*lücklichen Bernds, die eine Anna haben: $\{(a, f(a)) | a \in A\}$. Man beachte, dass $f(a) \in B$ ist. Der Graph besteht aus den Elementen der Zielmenge, die tatsächlich getroffen werden.

Definition 8 (Surjektivität, Injektivität, Bijektivität)**.** Sei $f : A \to B$ eine Abbildung wie oben. Ich erinnere daran, dass jedem a ein b sowieso fest zugewiesen ist, damit es eine Abbildung ist („jede Anna hat einen Bernd“). Dann ist f

surjektiv, wenn $\forall b \in B \exists a \in A : f(a) = b$ („jedes b wird getroffen“, „jeder Bernd hat eine Anna“),

injektiv, wenn $\forall a, a' \in A : (a \neq a' \Rightarrow f(a) \neq f(a'))$ („unterschiedliche Annas haben niemals den gleichen Bernd“), und

bijektiv, wenn sie surjektiv und injektiv sind. Bijektive Abbildungen sind die perfekte aller Welten: Nicht nur haben jede Anna und jeder Bernd jemanden, sondern das sind auch noch definierte Päärchen ohne irgendwelche Anomalien :-) Protipp: Wenn man zeigen will, dass zwei unendliche Mengen gleichmächtig sind, genügt es, eine bijektive Abbildung zwischen den beiden Mengen zu finden, was eine sehr praktische Art der Beweisfindung ist. Auf die Art ist man drauf gekommen, dass es mehrere Arten von „unendlich“ gibt. Man kann nämlich durchaus bijektive Abbildungen zwischen $\mathbb{N}$ und $\mathbb{Z}$ oder zwischen $\mathbb{Z}$ und $\mathbb{Q}$ finden – aber z.B. keine zwischen $\mathbb{N}$ und $\mathbb{R}$.

2 Zusammenhängende Systeme grob definieren: Signaturen. Konkrete Zusammenhänge definieren: Strukturen.

Definition 9 (Signatur)**.** Eine Signatur δ ist ein 5-Tupel (S, F, R, K, typ), für das die folgenden Kriterien gelten (mit denen wir die einzelnen Elemente nun erklären).

1. S ist eine Menge von Sorten, das sind sozusagen Datentypen. Konkrete Datentypen können theoretisch alles mögliche sein. $\mathbb{R}$, $\mathbb{N}$, Katzen, Hunde, Äpfel und Birnen. Die Menge S sagt uns aber nur symbolisch, auf wievielen verschiedenen Typen wir arbeiten, könnte also auch eine Art Durchnummerierung $\{1, 2, 3, 4\}$ sein.

2. typ ist eine Abbildung, die den Definitionsbereich $D(\text{typ}) = F \cup R \cup K$ hat (Wir erinnern uns: Für eine Abbildung f ist $D(f)$ der Definitionsbereich). Man kann also Elemente der Mengen F, R und K hineinstecken und ein Ergebnis erhalten. Was das ist, kommt gleich.

3. F ist eine Menge von Funktions*symbolen*. Achtung: Jedes $f \in F$ ist keine Funktion, sondern, ich wiederhole, nur ein Funktions*symbol*! Man kann ein Funktionssymbol in die Funktion typ stecken, dann erhält man $\text{typ}(f) \in S^{n+1}$ zurück, sprich, ein Tupel mit $n+1$ vielen Typen. Die ersten n Typen $s_1, s_2, \ldots, s_n$ sind die Eingabetypen für unser Funktionssymbol f, und der letzte Typ s_{n+1} repräsentiert den Rückgabetyp unseres Funktionssymbol. Und: Für jedes $f \in F$ gibt es so ein $\text{typ}(f) \in S^{n+1}$ – kein Funktionssymbol bleibt ohne Typzuweisung. Darum heißen die $f \in F$ auch Funktionsymbole – wir definieren nämlich gar keine „wirklichen“ Funktionen, sondern definieren nur die Eingabetypen und den Ausgabetyp, Definitionsbereich und Zielmenge, $D(f)$ und $B(f)$. Ein $f \in F$ symbolisiert also nur eine Abbildung von $S_1 \times S_2 \times \cdots \times S_n$ nach S_{n+1}.

4. R ist eine Menge von Relations*symbolen*, und der Rest läuft im Grunde genauso ab wie bei den Funktionssymbolen. Jedes $r \in R$ kriegt von der Typfunktion ein $\text{typ}(r) \in S^n$ zugewiesen. Warum hier n und nicht wie bei den Funktionssymbolen $n+1$? Ganz einfach: Relationen haben ja keinen expliziten Rückgabewert, darum gibt es einen Typ weniger für ein n-stelliges Relationssymbol im Vergleich zu einem n-stelligen Funktionssymbol. Ein $r \in R$ symbolisiert also nur eine Relation $\subseteq S_1 \times S_2 \times \cdots \times S_n$. So, das schwere haben wir jetzt hinter uns!

5. Bleibt noch die Menge K. Sie ist eine Menge von Konstantensymbolen, und auch hier gibt es für jedes $k \in K$ ein $\text{typ}(k)$ – nur dass $\text{typ}(k) \in S$ gilt. Typen von Konstanten kommen also direkt aus den Sorten, und sind nicht irgendwie aus verschiedenen Sorten zusammengesetzt.

Wir haben nun gerade die „Keksformen“ definiert für ein Zusammenspiel aus „echten“ Sorten, „echten“ Funktionen und „echten“ Relationen und „echten“ Konstanten. Wir haben sie alle nicht genau definiert, sondern nur symbolisiert, gewissermaßen eine Hülle geschaffen, Grundzusammenhänge definiert. Signaturen sind relativ sinnfrei ohne Definition der Strukturen, die jetzt kommt.

Definition 10 (Struktur zu einer Signatur)**.** Strukturen nutzen eine vorgegebene Signatur-Hülle, füllen sie mit Leben, wenden sie konkret an. Eine Struktur muss eine Signatur als Vorbild haben. Sei also wie oben definiert $\delta = (S, F, R, K, \text{typ})$ eine Signatur. Eine Struktur zu δ, kurz δ-Struktur, ist ein Tupel $\mathcal{A} = ((A_s)_{s\in S}, (f^{\mathcal{A}})_{f\in F}, (r^{\mathcal{A}})_{r\in R}, (k^{\mathcal{A}})_{k\in K})$. Sieht sehr verwirrend aus, ist aber einfach, also klamüsern wir es mal auseinander (die Stichpunkte sind gleichzeitig Kriterien für die Definition einer Struktur).

1. $(A_s)_{s\in S}$ bedeutet nichts anderes als „für jede Sorte s aus unserer Signatur-Sortenmenge S gibt es eine „Realisierung“ A_s in unserer Struktur. Hier werden also die oben definierten Durchnummerierungen realisiert zu tatsächlichen Typen, z.B. $\mathbb{R}$, $\mathbb{N}$, Katzen, Hunde, Äpfel und Birnen.

2. Die $(f^{\mathcal{A}})_{f\in F}$ sind dann Realisierungen unserer Funktionssymbole. Für jedes f aus der Signatur gibt es ein $f^{\mathcal{A}}$ aus der Struktur. Auch die Typen sind schon in der Signatur festgelegt, nämlich durch die Ausgabe von $\text{typ}(f)$. Angenommen, ein Funktionssymbol $f \in F$ bekommt die von der Typfunktion die Typen $\text{typ}(f) = (s_1, s_2, \ldots, s_{n+1})$ zugewiesen, so geht die zum Funktionssymbol gehörende Funktion $f^{\mathcal{A}}$ unserer Struktur $\mathcal{A}$ von den jeweiligen Realisierungen der Eingabetypen $A_{s_1} \times \ldots \times A_{s_n}$ zur Realisierung des Zieltyps $A_{s_{n+1}}$. Lange Rede, kurzer Sinn: Für jedes Funktionssymbol samt Typen, die bereits in der Signatur vorgesehen wurden, gibt es nun in unserer Struktur eine Funktion als Implementierung, die auf den ebenso implementierten Typen definiert ist.

3. Genauso gibt es für jedes Relationssymbol eine Relation als Implementierung, wobei die Typen wieder durch die Typfunktion definiert werden. Jedes $r \in R$ mit $\text{typ}(r) = s_1, s_2, \ldots, s_n$ wird realisiert durch eine Relation $r^{\mathcal{A}}$. Wir erinnerun uns, dass eine Relation Untermenge eines kartesischen Produktes verschiedener Mengen ist, und die Mengen sind in diesem Fall genau die Realisierungen der durch die Typ-Funktion vorgegebenen Sorten: $r^{\mathcal{A}} \subset A_{s_1} \times \ldots \times A_{s_n}$. Alles also analog zu den Funktionen.

4. Bleiben wieder die Konstanten: Auch hier gibt es zu jedem Konstantensymbol k mit $\text{typ}(k) = s$ eine Realisierung $k^{\mathcal{A}}$ im zugehörigen realisierten Typ A_s.

Wenn man mit einer Struktur eine Signatur realisiert, kann man noch weitere Forderungen (Axiome) anfügen, die die Struktur genauer definieren. Genauso machen wir es jetzt bei Gruppen und Körpern.

Definition 11 (Unterstruktur einer Struktur)**.** Eine Struktur U ist eine Unterstruktur von einer Strukur S, wenn gilt:

- ▷ Die Trägermenge vpn U muss Teilmenge der von S sein
- ▷ Konstanten müssen dieselben sein
- ▷ Verknüpfungen müssen dieselben sein
- ▷ Bezüglich der Verknüpfungen muss U abgeschlossen sein, d.h. wenn ich zwei Elemente aus U verknüpfe muss das Ergebnis wieder in U liegen

2.1 Sehr häufige konkrete Strukturen: Gruppen, Körper, etc

Tipp vorab: Oft kommen in Prüfungen Fragen wie „ist das-und-das eine Gruppe / ein Körper / etc?“ – wie geht ihr am besten an sowas ran? Was fällt einem da zuerst ein? Na klaaaar, die Axiome durchzurechnen. Und genau diese Herangehensweise ist SCHLECHT, weil sie im Zweifel lange dauert, und nichts bringt. Zuerst checkt ihr bitte, ob die Operationen der Struktur bezüglich der Trägermenge abgeschlossen sind, und erst wenn das geglückt ist, prüft ihr die Axiome. Hintergrund: Oft kann man nämlich sehr schnell und intuitiv irgendein Element finden, was durch die Operationen erzeug werden kann und nicht in der Trägermenge liegt, und dann kann man sich das abrattern der Axiome sparen.

Definition 12 (Gruppen, Gruppenordnung, endliche, abelsche)**.** Eine konkrete Signatur wäre $\gamma = (\{1\}, \{f\}, \emptyset, \text{typ})$. Es gibt also einen einzigen Datentyp (Symbol 1), ein einziges Funktionssymbol f, kein Relationssymbol und ein einziges Konstantensymbol k. Fehlt noch die Definition von typ: Sei $\text{typ}(f) = (1, 1, 1)$ und $\text{typ}(k) = 1$. Mit anderen Worten, das Funktionssymbol f repräsentiert eine Funktion, mit einer zweistelligen Eingabe unseres einzigen Datentyps, die ein Element unseres einzigen Datentyps ausgibt. Weiter gibt es eine Konstante, wieder aus unserem einzigen Datentyp. Das ist eine sehr einfache Signatur.

Eine Gruppe ist nun eine Struktur, die genau diese Signatur γ realisiert – wenn noch ein paar zusätzliche Axiome gelten. Wir realisieren nun γ. Eine γ-Struktur $(G, \circ, \emptyset, e)$ heißt genau Gruppe, wenn für die Konstante e und die Abbildung $\circ : G \times G \to G$ folgende Axiome erfüllt sind (Kurzschreibweise $a \circ b$ statt $\circ(a, b)$, machen wir ja im wirklichen Leben mit $+$ und $-$ auch so:

1. $\forall a, b, c \in G : (a \circ b) \circ c = a \circ (b \circ c)$. (Assoziativität, Austauschbarkeit der Klammern bezüglich $\circ$).
2. $\forall a \in G : a \circ e = a = e \circ a$. Das heisst, dass unsere Konstante e neutrales Element ist, also mit der Operation $\circ$ ein anderes Element nicht verändert, egal, ob es von rechts oder links mittels $\circ$ mit einem $a \in G$ verknüpft wird.

3. $\forall a \in G \exists b \in G : a \circ b = e = b \circ a$. Es existieren Inverse Elemente, die ein jedes $a \in G$ bezüglich der Operation $\circ$ auf die Konstante zurückführen, egal von welcher Seite sie durch $\circ$ verknüpft werden.

Eine Gruppe ist endlich, wenn G eine endliche Menge ist, dann ist $|G|$ ihre Ordnung. Bei einer abelschen Gruppe ist $\circ$ kommutativ, also $a \circ b = b \circ a$ für alle a, b aus G (wir sehen, das ist nicht selbstverständlich).

Über Gruppen kann man allerlei beweisen, ohne wirklich zu wissen, was sich hinter G und $\circ$ konkret verbirgt. **Allgemein sind solche Strukturen schon mal da, um schon grundlegende Sachen beweisen zu können, ohne direkt konkrete Strukturen angucken zu müssen – Wer etwas für das allgemeine Konzept „Gruppe“ beweist, hat es automatisch für alle konkreten Gruppen bewiesen. Lohnt sich, oder?** Das ist genau der Sinn von solchen Strukturen: Aussagen allgemein treffen zu können, die dann für jede konkrete Struktur bewiesen sind, was nachweisbar in dasselbe Schema passt. Beispiele: $\mathbb{Z}$ ist bezüglich der Addition $+$ eine Gruppe. Das Neutrale Element ist in dem Fall 0, und die inversen Elemente die jeweilig passenden negativen Zahlen. $\mathbb{Z}$ ist mit der Multiplikation hingegen keine Gruppe, weil es keine Inversen Elemente gibt, ausser dem von 1. Es gibt im Script viele Beispiele für Gruppen. Die solltet ihr mal verstanden haben, aber braucht ihr nicht auswendig lernen.

Übrigens: Wenn G eine Gruppe ist und H Untergruppe davon, dann ist die Ordnung von H Teiler derjenigen von G.

Definition 13 (Monoid)**.** Wie Gruppe, aber ohne inverses Element.

Definition 14 (Körper)**.** Körper sind eine umfangreichere Struktur als Gruppen, uns aber aus dem täglichen Leben und der Schulmathematik bekannt. Es gibt im Vergleich zur Gruppe nicht eine, sondern zwei Konstanten und nicht eine, sondern zwei Abbildungen, sowie einige Axiome mehr. Man nehme also eine Trägermenge K, zwei Abbildungen/Verknüpfungen/Funktionen $+_K : K^2 \to K$ und $\cdot_K : K^2 \to K$, und dann noch zwei Konstanten 1_K und 0_K. Wir machen uns es wie in der Vorlesung einfach und lassen die $_K$ ab hier für den Rest der Definition weg. Zusätzlich zum Vorhandensein der obigen Elemente müssen, damit man einen Körper hat, noch einige Axiome gelten:

1. Zunächst ein paar Axiome zur Operation $+$ und der „zugehörigen“ Konstante 0. $\forall a, b, c \in K$:

$$\begin{aligned} a + b &= b + a && \text{Kommutativität von } + \\ (a + b) + c &= a + (b + c) && \text{Assoziativität von } + \\ a + 0 &= a = 0 + a && \text{Das neutrale Element von } + \text{ ist die Konstante } 0 \end{aligned}$$

2. Jetzt analog für die Operation $\cdot$ sowie 1. $\forall a, b, c \in K$:

$$\begin{aligned} a \cdot b &= b \cdot a && \text{Kommutativität von } \cdot \\ (a \cdot b) \cdot c &= a \cdot (b \cdot c) && \text{Assoziativität von } \cdot \\ a \cdot 1 &= a = 1 \cdot a && \text{Auch } \cdot \text{ hat ein neutrales Element: Die 1.} \end{aligned}$$

3. Jedes $k \in K$ hat bezüglich der Verknüpfungen $+$ und $\cdot$ auch noch ein inverses Element (im Falle von $\cdot$ hat die 0 selbst keins):

$$\begin{aligned} \forall a \in K \exists (-a) : &\quad a + (-a) = 0 \\ \forall a \in K \setminus 0 \exists \frac{1}{a} : &\quad a \cdot \frac{1}{a} = 1 \end{aligned}$$

$$ru$$

Achtung: Bis jetzt wissen wir noch gar nicht, was sich hinter den Operationen $+_K$ und $\cdot_K$ verbirgt. Wir wissen auch nicht, was genau unsere Trägermenge K ist und daher auch nicht, was denn die inversen Elemente $\frac{1}{a}$ und $-a$ sein sollen. Sowohl $\frac{1}{a}$ als auch $-a$ sind daher bis jetzt einfach nur Symbole, bzw. Platzhalter.

4. Zuletzt gibt es noch ein paar Axiome, die mit beiden Verknüpfungen gleichzeitig zusammenwirken: Die Distributivgesetze. $\forall a, b, c \in K$:

$$\begin{aligned} a \cdot (b + c) &= a \cdot b + a \cdot c \\ (a + b) \cdot + c &= a \cdot c + b \cdot c \end{aligned}$$

Tja, und wie wir sehen, entspricht das insgesamt ganz genau den uns wohlbekannten Rechenregeln auf den reellen Zahlen, wenn wir für die $+_K, \cdot_K, 1_K, 0_K$ unsere wohlbekannten Operationen und konstanten $+, \cdot, 1, 0$ nehmen. Die Trägermenge $\mathbb{R}$ zusammen mit den uns bekannten Operationen $+$ und $\cdot$ und ihren neutralen Elementen $0 \in \mathbb{R}$ und $1 \in \mathbb{R}$ bilden also einen Körper. Es gibt noch viele andere gängige Körper, zum beispiel denjenigen über die Menge der Komplexen Zahlen $\mathbb{C}$ mit den Operationen der komplexen Addition und Multiplikation und den zugehörigen Konstanten.

Definition 15 (Körpercharakteristik)**.** Wenn man beliebig oft das 1-Element eines endlichen Körpers K addiert, muss man irgendwann beim 0-Element rauskommen. Das gilt natürlich nicht für unendliche Körper. Der Angenommen, wir addieren also n mal die 1 des Körpers (kurzschreibweise $S_K(n)$). Das kleinste n, für das wir auf diese Weise zum Null-Element gelangen (also das kleinste n mit $S_K(n) = 0$) heisst die Charakteristik des Körpers und ist eine wichtige Kenngröße. Falls es kein solches n gibt (wie zum Beispiel bei den uns bekannten Körpern über $\mathbb{R}$ oder $\mathbb{C}$) ist die Charakteristik 0.

Definition 16 (Ring)**.** Wie Körper, aber bei Multiplikationsoperator ohne inverses Element und ohne Kommutativität. Hier ein paar „Verbindungseselsbrücken“ zwischen Ring und anderen Strukturen:

- ▷ Ein Ring mit kommutativem Multiplikationsoperator und multiplikativem inversen ist ein Körper
- ▷ Wenn man aus einem Ring den Multiplikationsoperator und Axiome, die damit zu tun haben, weglässt, ist der Rest eine abelsche Gruppe
- ▷ Wenn man aus einem Ring den Additionsoperator samt zugehörigen Axiomen weglässt, ist der Rest ein Monoid.

Definition 17 (Geordneter Körper)**.** Ein Körper K heißt geordnet, wenn es zusätzlich noch eine Relation $<_K \subseteq K \times K$ gibt, die die Eigenschaften einer Ordnung erfüllt:

1. $\forall x, y \in K : x = y \vee x <_K y \vee y <_K y$. Ein Element ist entweder gleich, oder kleiner, oder größer als ein anderes, aber niemals mehrere dieser Eigenschaften zusammen. Heir ist also ein exklusives oder gemeint!
2. $\forall x, y, z \in K : x <_K y \Rightarrow x +_K z < y +_K z$. Die Relation bleibt erhalten, wenn auf beiden Seiten des $<_K$ dasselbe „addiert“ wird.
3. $\forall x, y, z \in K : x <_K u \wedge 0 <_K z \Rightarrow x \cdot_K z <_K y \cdot_K z$. Die Relation bleibt erhalten, wenn auf beiden Seiten des $<_K$ ein Element „anmultipliziert“ wird, das „größer“ als 0_k ist.

Nimmt man nun unseren oben schon genannten Körper über $\mathbb{R}$ mit $+$ und $\cdot$, und nimmt als Relation $<_K$ das uns ebenfalls schon lange bekannte $<$ hinzu, so hat man einen geordneten Körper. Würde man die Signaturen von Körpern und geordneten Körpern aufschreiben, gäbe es also im geordneten noch eine Relation zusätzlich.

2.2 Ähnlichkeit von Strukturen

Man kann etwas über die Ähnlichkeit von Strukturen aussagen (in dem Fall jetzt: Gruppen), indem man schaut, ob es möglich ist, bestimmte Abbildungen zu definieren. Schafft man es, eine Abbildung zu definieren, die untenstehenden Kriterien genügt, so kann man zwei Gruppen als „praktisch gleich“ sehen.

Definition 18 (Gruppenhomomorphismus, Gruppenisomorphismus, isomorphe Gruppen)**.** Angenommen $(G, \circ)$ und $(H, *)$ sind zwei Gruppen und $f : G \to H$ eine Abbildung. Wenn f folgende Kriterien erfüllt, ist f ein Gruppenhomomorphismus:

- ▷ $f(1_G) = 1_H$ (das neutrale Element von G wird in das von H übersetzt)
- ▷ $\forall a, b \in G : f(a \circ b) = f(a) * f(b)$. Muss man sich einmal klarmachen. a und b sind $\in G$ (auf Elemente in G ist nur die Operation $\circ$ anwendbar), dahingegen sind $f(a)$ und $f(b) \in H$ (auf Sachen aus H ist wiederum nur $*$ anwendbar). Es ist nun wichtig festzustellen, dass f – als Übersetzer zwischen den beiden Welten – bei den Gruppenverknüpfungen keinen Ärger macht. Dazu weist man nach, dass es egal ist, ob man a und b vor der Abbildung mit $\circ$ verknüpft, oder halt $f(a)$ und $f(b)$ nach der Abbildung mit $*$.

f heisst weiter ein Gruppenisomorphismus (das ist stärker als Homomorphismus), wenn f ein Gruppenhomomorpmismus und dazu noch bijektiv ist. Die beiden Gruppen sind isomorph, wenn es einen solchen Isomorphismus gibt (geschrieben $(G, \circ) \simeq (H, *)$.

Definition 19 (Ringhomomorphismus, Ringisomorphismus, Kern)**.** Völlig analog wird ein Ringhomomorphismus definiert. Ringe statt Gruppen, zwei Konstanten statt einer, zwei Verknüpfungsverträglichkeiten statt einer. Der zugehörige Ringisomorphismus ist wieder bijektiv. Da Ringe eine umfangreichere Struktur sind, gibt es dennoch einen kleinen Zusatz: Den Kern. Seien R und S Ringe (mit den zugehörigen Operationen, natürlich, sind hier aber egal) – und sei ϕ ein Ringhomomorpmismus. Dann gibt es eine Teilmenge von dem Startring R, die von ϕ auf das Nullelement in S abgebildet wird (da die Nullkonstanten aufeinander abgebildet werden, ist diese Teilmenge wenigstens 1 Element groß, kann aber auch größer sein). Diese Teilmenge nennt man den Kern des Homomorphismus. Formal: $\mathrm{Ker}(\phi) = \{r \in R | \phi(r) = 0_S\}$. Ja, es heisst „Ker“ und nicht „Kern“.

Definition 20 (Körper*Morphismen)**.** Die könnt ihr euch ja wohl jetzt selbst definieren :-) Wieder Konstanten aufeinander abbilden, und dann die Axiome durchgehen und die Abbildungsverträglichkeiten definieren. Achtung, beim Körper gibt es ein paar Axiome, die beide Abbildungen gleichzeitig benutzen :-) Mehr ist aber auch schon nicht zu beachten. Da ein Körper ja auch immer ein Ring ist, gibt es hier auch einen Kern.

Definition 21 (Komplett verallgemeinerte Isomorphie und Homomorphie)**.** Ab Seite 66 der Vorlesungsnotizen findet ihr einen Komplett verallgemeinerten Homomorphie- und Isomorphiebegriff, aus dem sich alle anderen oben beschriebenen komplett ergeben. Dieser verallgemeinerte Begriff kommt dann auch mit Relationen, mehreren Trägermengen und den sonstigen Möglichkeiten, die uns die Signaturen so in die Hand geben, klar. Wenn man die Konkreten Anwendungen auf Gruppen, etc. verstanden hat, sollte man sich das mal durchlesen, dann weiss man eigentlich sofort, was da passiert.

3 Zahlen

Wir haben verschiedene Mengen von Zahlen besprochen. In der Schule spricht man über rationale Zahlen, natürliche Zahlen, reelle Zahlen ganz selbstverständlich, und im Alltag auch. Wenn es aber mal wirklich um die Frage geht **„was sind diese Zahlen?“**, dann werdet ihr feststellen, dass ihr sehr schnell denkt ... jaaa ... mmmh, ja was denn eigentlich?! Hier also ein kleiner Definitionsguide, in Richtung immer größerer Zahlenmengen. Wir gehen quasi in aller Kürze, als Überblick, folgende Reihe durch: $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$.

Definition 22 (Natürliche Zahlen, $\mathbb{N}$)**.** Die natürlichen Zahlen $\{1, 2, 3, 4, \ldots\}$, die jeder von euch aus der Schule kennt. Hier gibt es nichts weiter zu sagen. Abzählbar. Gerne auch mit $\mathbb{N}_0$ oder ähnlich beschrieben, wenn man die 0 dazunimmt.

Definition 23 (Ganze Zahlen, $\mathbb{Z}$)**.** Alle positiven und negativen ganzen Zahlen und die Null. Definition aufbauend auf den natürlichen Zahlen: $\mathbb{Z} = \{\pm x | x \in \mathbb{N}\} \cup 0$. Auch abzählbar.

Definition 24 (Rationale Zahlen, $\mathbb{Q}$)**.** Nach dem wir $\mathbb{Z}$ auf $\mathbb{N}$ aufgebaut haben, bauen wir hier nun weiter: Mittels der ganzen Zahlen $\mathbb{Z}$ kann man abzählbar unendlich viele Brüche bilden. Die Gesamtheit der Brüche sind die Rationalen Zahlen. Formaler: $\mathbb{Q} = \{\frac{a}{b} | a, b \in \mathbb{Z}, b \neq 0\}$. Da in Mengen keine Elemente mehrmals vorkommen, könnte man sagen, dass gleichwertige Brüche „automatisch gefiltert“ werden. Man kann aber auch einfach, wie Herr Klein auf Seite 33.1 des Sciptes schöner definieren, also schaut euch das noch mal an. Erst wird eine Äquivalenzrelation auf Brüchen definiert, die man mit „Bruch ist gleichwertig“ übersetzen könnte, so dass eine konkrete rationale Zahl einer Äquivalenzklasse aus dieser Relation entspricht. Umgekehrt ausgedrückt: Alle gleichwertigen Brüche sind dieselbe Rationale Zahl, und werden in einer Äquivalenzklasse zusammengefasst. Die Menge $\mathbb{Q}$ ist dann nichts weiter, als alle dieser Äquivalenzklassen. Deutlich schönere Definition, oder? Übrigens: Rationale Zahlen sind auch abzählbar, googelt mal nach dem Cantorschen Diagonalprinzip.

Definition 25 (Reelle Zahlen, $\mathbb{R}$)**.** Sozusagen alle möglichen Zahlen, auch die „zwischen“ den Rationalen. Es gibt also im Zahlenstrang noch zusätzliche Zahlen, die nicht rational sind, aber von rationalen Zahlen beliebig gut angenähert werden können. Nicht mehr Abzählbar. Die Definition hat einen Umweg über die Analysis gemacht: Es wurden Cauchyfolgen definiert und gesagt, dass die Reellen Zahlen alle Grenzwerte von Folgen rationaler Zahlen sind (wie das genau funktioniert, werdet ihr noch in Analysis lernen).

Definition 26 (Komplexe Zahlen, $\mathbb{C}$)**.** Leider ist auch $\mathbb{R}$ noch nicht mächtig genug für alles was wir machen wollen, denn nicht jede Polynomgleichung hat eine Lösung in $\mathbb{R}$ – also wurde $\mathbb{R}$ nochmals erweitert zu den Komplexen Zahlen $\mathbb{C} = \{a+bi | a, b \in \mathbb{R}\}$. In $\mathbb{C}$ kann man dann algebraisch alles mögliche tun, ohne Angst haben zu müssen, dass plötzlich irgendetwas kein Ergebnis hat. Über die komplexen Zahlen haben wir im Tutorium ja schon ausführlichst gesprochen (Komplexe Addition, komplexe Multiplikation, komplex Konjugiertes, Polarkoordinaten versus kartesische Koordinaten). **Diese Grundregeln von $\mathbb{C}$ zu kennen ist wiederum ein absolut essentieller Teil der Sprache der Informatik, und ihr werdet erhebliche Nachteile im Studium haben, wenn ihr davon kein Verständnis habt.**

4 Beweistechniken

Definition 27 (Direkter beweis durch Umformen und einsetzen)**.** Die kanonische Straight-Forward-Variante. Wer das macht, hat ein Problem tatsächlich ordentlich verstanden, geht aber nicht immer. Benötigt irgendwelche Grundlagen, z.B. Axiome, mit denen man anfängt.

Definition 28 (Indirekter Beweis)**.** Man nimmt das Gegenteil von dem was man zeigen will an, und leitet daraus eine Aussage ab, von der man weiss, dass sie unwahr ist.

Definition 29 (Beweis durch Anzahlargumente und/oder Abbildungen)**.** Wie schon in der Definition zur sur/in/bijektivität von Abbildungen gesagt, kann man mittels der Abbildungsarten beweisen, ob eine Menge mächtiger ist als die andere.

Definition 30. Hiermit kann man abzählbare Sachen beweisen. Man beweist zwei Teile:

Induktionsanfang (IA): Die Aussage gilt für eine „Startzahl“ oder ein „Startelement“ der Menge über die wir beweisen, das ist meist der einfache Teil.

Induktionsschritt (IS): Die Aussage gilt „verschiebbar“. Falls die Aussage für ein Element unserer Menge über die wir beweisen gilt, dann gilt sie auch für „das nächste Element“ (Aussage ist hier bewusst etwas schwammiger gehalten, weil es gar nicht sein muss, dass man über Zahlen beweist). Der erste Teil des Satzes, „Falls Aussage für ein Element gilt“, sagt uns: Das dürfen wir voraussetzen. Man nimmt also, wenn man über Zahlen beweist, an: Die Aussage gilt für Zahl n. Von dort formt man dann um und zeigt, dass die Aussage auch für $n+1$ gilt.

Warum ein Induktionsbeweis oft besonders cool funktioniert, wenn man eine geschlossene Formel für rekursive Definitionen finden will, haben wir ja in einem unserer Crashkurse im Tutorium ausführlich besprochen. **Die Induktionsbeweistechnik ist auch wieder sowas, was für Informatiker beliebiger Ausprägung ein must-have ist. Nicht können → derbe viele Probleme im Studium.**

5 Teilbarkeit, Kongruenzrechnung

Definition 31 (Teiler und Vielfache)**.** Seien $a, b, n \in \mathbb{Z}$. Dann gilt $n|a$, d.h. n teilt a, wenn es ein $a' \in \mathbb{Z}$ gibt mit $a = n \cdot a'$. n heißt dann Teiler von a, und a ist umgekehrt ein Vielfaches von n.

Definition 32 (Kongruenz)**.** Seien $a, b, n \in \mathbb{Z}$. Wenn a und b bei der Division durch n den gleichen Rest haben, sind sie Kongruent, geschrieben $a \equiv b \mod n$. Dies ist gleichbedeutend zu $n|a-b$ (wenn n Teiler der Differenz $(a-b)$ ist, dann haben sowohl a und b bei Division durch n den gleichen Rest). Die Kongruenz ist eine Äquivalenzrelation.

Theorem (Chinesischer Restsatz)**.** Muss man sich wie ein Spiel vorstellen. Angenommen, jemand präsentiert uns zwei teilerfremde Divisoren $n, m \in \mathbb{N}$. Die stehen fest, an denen können wir nichts verhandeln. Danach dürfen wir zwei beliebige „zu teilende Zahlen“ $a, b \in \mathbb{Z}$ wählen. Ganz egal, wie diese a, b aussehen, egal, wie fies wir sie gewählt haben: Der andere kann uns immer einen gemeinsamen Divisionsrest $k \in \mathbb{N}$ finden, so dass gleichzeitig gilt $a \equiv k \mod n$ und $b \equiv k \mod m$. Einer präsentiert ein Teilerpaar, der Gegenspieler präsentiert zwei Zahlen, die damit geteilt werden sollen, und es gibt immer ein k, dass dann für beide Divisionen Rest ist.

Tja, und wer ist der Verlierer des Spiels? Ganz einfach: Der arme dritte, der das k überhaupt suchen muss. Der Chinesische Restsatz ist nämlich nicht konstruktiv. Er sagt nur aus „es gibt ein k“, aber nicht „so findet man es“. Wir erhalten Ermutigung, aber keine Hilfe bei der Suche.

Definition 33 (Restklasse)**.** Für $x, y \in \mathbb{N}$ ist $[x]_y = \{z | z \in \mathbb{Z}, z \equiv x \mod y\}$, also salopp die Menge aller ganzen Zahlen, die kogruent zu x sind, modulo y. Wir nennen x hier Repräsentant.

Definition 34 (Restklassenring)**.** Sei $n \in \mathbb{N}$. Dann ist $\mathbb{Z}/n\mathbb{Z} = \{[0]_n, [1]_n, \ldots, [n-1]_n\}$, also die Menge aller Restklassen modulo n. Sie ist ein Ring, wenn man folgendes hinzunimmt:

- Als Addition (wie gewohnt verknüpfen wir wieder zwei Elemente der Trägermenge (das sind keine einfachen Zahlen, sondern Restklassen) zu einem, die Formalia lasse ich hier dann weg): $[a]_n + [b]_n = [a+b]_n$, sowie
- als Multiplikation $[a]_n \cdot [b]_n = [a \cdot b]_n$.
- Neutrale Elemente für beide Operationen sind $[0]_n$ bzw. $[1]_n$.

Definition 35 (Einheiten und Nullteiler)**.** In einem Restklassenring über n ist ein Element a ein Nullteiler, wenn es ein b dazu gibt, mit dem a auf null multipliziert wird. Bei Einheiten ist die Definition analog, allerdings wird hier auf eins Multipliziert. Ein Element eines Restklassenringes ist entweder Nullteiler oder Einheit, bis auf die Null, die ist beides nicht. Ein x aus einer Restklasse Modulo n ist genau dann kein Nullteiler, wenn es teilerfremd zu n ist, genau dann ist es auch eine Einheit. Das heisst zum Beispiel, dass, es keine Nullteiler gibt, wenn n Primzahl ist, denn alle Zahlen von 0 bis $n-1$ (die ja Restklassenrepräsentanten sind) sind dann Teilerfremd zu n. Ist n Primzahl, dann gibt es keine Nullteiler. Außer der Null at der Restklassenring dann nur Einheiten. Jedes Element des Rings hat dann ein multiplikatives Inverses, und unser Ring ist ein Körper.

Definition 36 (Einheitengruppe)**.** Wenn R ein kommutativer Ring mit Eins ist (also z.B. ein Restklassenring), dann sei $R^\times$ die Menge aller Einheiten von R. Diese bildet eine Gruppe bezüglich der Multiplikation.

Definition 37 (Eulersche ϕ-Funktion)**.** Sei $n \in \mathbb{N}$. Wir definieren nun die Funktion $\phi(n) = |(\mathbb{Z}/n\mathbb{Z})^\times|$, also die Ordnung der Einheitengruppe zu $\mathbb{Z}/n\mathbb{Z}$. Anders ausgedrückt: Die Menge der zu n teilerfremden Zahlen, die größer als 0 und kleiner als n sind.

Theorem (Kleiner Fermat'scher Satz)**.** Wenn p eine Primzahl ist, dann dürfen wir ein beliebiges $a \in \mathbb{Z}$ wählen, dass nicht vielfaches von p ist, und es gilt $a^{p-1} \equiv 1 \mod p$. Es gibt auch Nicht-primzahlen n, für die $a^{n-1} \equiv 1 \mod n$, also das gleiche gilt – die nennt man Carmichael-Zahl.

Theorem (Großer Fermat'scher Satz)**.** Wenn $a, b, c, n \in \mathbb{N}$ und $n \geq 3$ sowie $a, b, c \neq 0$: Dann ist $a^n + b^n \neq c^n$.

Definition 38 (Euklidischer Algorithmus)**.** *In diesem Algorithmus ist jetzt ein Fehler verbessert, in Zeile 4 stand vorher „$b \leftarrow b$“.* Wir haben ja beim Chinesischen Restsatz gesagt, dass er uns keinen Anhaltspunkt liefert, das k zu finden. Das machen wir dann jetzt. Neu!

```
1  #Assertions: n, m ∈ ℕ, n ≥ m, beide nicht 0
2  ALGORITHM Euklid (n,m)
3      a ← n
4      b ← m
5      WHILE b > 0:
6          suche ein q ∈ ℕ, r ∈ {0, 1, ..., b − 1} mit a = qb + r
7          a ← b
8          b ← r
9      END WHILE
10     return a #a ist der ggT von n und m
11 END ALGORITHM
```

Ein gutes Beispiel findet sich in der Vorlesung. Zum Üben malt man sich einfach eine Tabelle mit den a und b und den Ergebnissen der Bestimmung von q und füllt die Zeilenweise aus.

6 Graphen

Graphen sind eines der allerwichtigsten Konzepte in der Informatik und Mathematik und eine der weiteren Sachen, die ihr unbedingt aus der Vorlesung mitnehmen solltet, weil sie einfach überall zum Einsatz kommen.

Neu! **Definition 39** (Graph, Knoten, Kanten)**.** Sei $G = (V, E)$ ein Graph. Dann ist

- V die Menge der Knoten (Vertices), die man sich wie Punkte oder Orte vorstellen kann.
- E die Menge der Kanten (Edges). Kanten sind Verbindungen zwischen Knoten und daher Teilmengen von V (theoretisch könnte eine Kante also eine „Verbindung“ zwischen ganz vielen Knoten sein, das ist bei Hypergraphen der Fall). In den allermeisten Fällen ist eine Kante aber nur eine zweielementige Teilmenge der Knoten, verbindet also nur zwei Knoten, was man sich wieder gut bildlich vorstellen kann als „Straße“ zwischen zwei Orten. E ist also formal eine Menge von k-Elementigen Teilmengen von V, geschrieben $E \subseteq \binom{V}{k}$, wobei meistens $k = 2$.

Merkt: Durch die Mengendefinition werden mehrfach vorhandene Kanten automatisch eliminiert. Würden wir Mehrfachkanten wollen (solche Fälle gibt es) müssten wir uns hier also etwas anderes einfallen lassen als eine Kantenmenge, oder die Kanten irgendwie unterscheidbar machen, z.B. mit einem Index. Da die Kanten $e \in E$ außerdem als Knotenmengen repräsentiert werden, ist der Graph ungerichtet: Es gibt ja keine ausgezeichneten „Start“- und „End“knoten, sondern eben nur zwei Knoten.

Definition 40 (Schlichter Graph)**.** Ein ungerichteter Graph (wie oben) ohne mehrfachkanten (wie oben) und ohne Schleifen. Schleifen sind Kanten, die von einem zu demselben Knoten führen.

Neu! **Definition 41** (Endlicher Graph)**.** Graph mit endlich vielen Knoten. Die meisten Graphen, denen ihr begegnen werdet, haben nur endlich viele Knoten (aber unter Umständen viele davon: Googles Datenbasis ist ein riesiger Graph – Knoten sind Webadressen, Kanten sind Links dazwischen).

Neu! **Definition 42** (Bipartite Graphen)**.** Graphen, deren Knoten zwei „Parteien“ $A \subset V$ und $B \subset V$ bilden. Kanten gibt es nur zwischen den beiden Parteien, niemals innerhalb einer Partei.

Neu! **Definition 43** (Einbettung von Graphen)**.** „Aufgemalte“ Version eines Graphen auf eine bestimmte Art Körper, z.B. eine 2D-Fläche, eine Kugel, einen Torus, ... man weist den Knoten Orte auf dem Körper zu, und malt dann die Kanten dazwischen.

Neu! **Definition 44** (Planar, Plan)**.** Ein planarer Graph ist so auf einer 2D-Fläche einbettbar, dass sich keine Kanten kreuzen. Ein planer Graph ist ein tatsächlich auf solche Weise eingebetteter Graph. Für solche Graphen (planar und plan) gilt die Eulerformel.

Neu! **Definition 45** (Eulerformel samt Beweisskizze)**.** Für einen planaren Graphen gilt die wichtige Eulerformel

$$v - e + f = c + 1$$

wobei v die Anzahl der Knoten ist, e die Anzahl der Kanten, f die Anzahl der Flächen und c die Anzahl der Zusammenhangskomponenten. Der Beweis lässt sich induktiv führen, hier gibt es den „neuen“ Beweis aus der Vorlesung:

Induktion über die Kanten. Im Induktionsanfang geht man von einem „Graphen“ aus, der schon alle gewünschten Knoten besitzt, aber noch keine einzige Kante. Dann hat man n Knoten, 0 Kanten, 1 Fläche, n Zusammenhangskomponenten und die Formel stimmt. Im Induktionsschritt fügt man eine Kante hinzu. Entweder, man verschmilzt dabei zwei Zusammenhangskomponenten, dann bleibt die Anzahl der Flächen gleich, die Kantenanzahl erhöht sich um eins, und die Anzahl der der Zusammenhangskomponenten wird um eins gesenkt, Formel stimmt. Oder man verschmilzt keine Zusammenhangskomponenten, dann schließt man eber eine neue Fläche ein. In dem Fall erhöht sich die Anzahl der Kanten um eins und die Anzahl der Flächen auch, Formel stimmt wieder.

7 Aussagenlogik

Aussagenlogik dreht sich um boolesche Variablen, die entweder „wahr“ (W) oder „falsch“ (F) sein können. Diese werden verbunden mit den Junktoren $(\vee, \wedge, \rightarrow, \leftrightarrow, \neg)$ um komplexere logische Ausdrücke zu formen. Man interessiert sich dann für die Wahrheitswerte der so erzeugten komplexeren Aussagen.

Wir definieren uns im Folgenden kurz die Sprache, mit der wir reden (AL). Auf dieser werden wir dann einfache Textersetzungsregeln definieren (AL-Kalkül, Syntax) und den an sich bedeutungslosen Worten Bedeutungen zuweisen (Semantik).

7.1 Syntax: Textersetzungen in Strings, die keine Bedeutung haben

Definition 46 (Menge der aussagenlogischen Ausdrücke, Induktion über Aufbau)**.** Zuerst definieren wir uns, mit was für Strings wir überhaupt arbeiten. Sei Π eine Menge von booleschen Variablen, das sind unsere Grundbausteine. Dann definieren wir uns jetzt die Menge der aussagenlogischen Ausdrücke über Π:

1. Jede der Variablen selbst ist ein aussagenlogischer Ausdruck: Jedes $p \in \Pi \Rightarrow p \in \mathrm{AL}(\Pi)$).
2. Wenn ein beliebiger (auch zusammengesetzter!) Ausdruck α zu $\mathrm{AL}(\Pi)$ gehört, dann auch sein negiertes $\neg\alpha$.
3. Wenn zwei ebenso beliebige α, β zu $\mathrm{AL}(\Pi)$ gehören, dann auch die Zeichenkette $(\alpha \vee \beta)$. Achtung: Ich meine hier wirklich zeichenweises Ersetzen, das ganze ohne Klammern z.B. gilt nicht!
4. Sonst gehört nichts zu $\mathrm{AL}(\Pi)$.

Salopp gesagt gibt es also eine Menge von Grundbausteinen, den Variablen, und zwei Bau-Regeln, mit denen wir aus bereits existierenden Ausdrücken neue Ausdrücke zusammenbasteln können.

Definition 47. Über die Grundbausteine und Bauregeln kann man auch Induktion betreiben:

Induktionsanfang: Nachweisen, dass die gewünschte Eigenschaft E für die Grundbausteine gilt

Induktionsschritte: Annehmen, dass E für zwei beliebige Ausdrücke α und β gilt, und nachweisen, dass Ergebnisse von Anwendungen der „Bastelregeln“ auf α und β die Eigenschaft E ebenfalls besitzen. Mit andern Worten: Die Regeln machen die Eigenschaft nicht kaputt.

Wenn man das bewiesen hat, ist klar, dass E für alle Ausdrücke $\in \mathrm{AL}(\Pi)$ gilt.

Definition 48 (Zusätzliche Abkürzungen)**.** Die Zeichen $\wedge, \rightarrow, \leftrightarrow$ gibt es in unserer eigentlichen Sprache erstmal gar nicht – sie sind nicht in der Originaldefinition erfasst, sondern einfach zusätzliche Abkürzungen, siehe Seite 133 des Scriptes.

Jetzt wissen wir, auf was für Strings wir arbeiten und wie man Eigenschaften über alle diese Strings beweisen kann. Ich möchte festhalten, dass wir bis jetzt auf Zeichenketten arbeiten, in die wir keine Bedeutung irgendeiner Art hineininterpretieren. Diese Zeichenketten kann man nach bestimmten Regeln in andere Zeichenketten umändern. Diese Textersetzungsregeln stehen im aussagenlogischen Kalkül.

Definition 49 (Aussagenlogisches Kalkül)**.** Für alle $\alpha, \beta, \gamma \in \mathrm{AL}(\Pi)$ darf man folgende vier Ausdrücke hinschreiben (das sind Axiome, die werden also einfach gefordert):

1. $\alpha \vee \alpha \rightarrow \alpha$
2. $\alpha \rightarrow \alpha \vee \beta$ (Aussagen dürfen „drangeodert werden“)
3. $\alpha \vee \beta \rightarrow \beta \vee \alpha$ (Aussagen dürfen um ein $\vee$ gedreht werden)
4. $(\alpha \rightarrow \beta) \rightarrow (\gamma \vee \alpha \rightarrow \gamma \vee \beta)$

Zusätzlich gibt es noch die Regel „Modus Ponens“, die aussagt: Weiss man α und weiss man zusätzlich $\alpha \rightarrow \beta$, so darf man auch β hinschreiben. Aus dieser grundlegenden Axiomatik können wir die Higher-Level-Regeln herleiten, die auf den Seiten 139ff. beschrieben sind. Vier Axiome, eine Regel. **Dieses Kalkül ist für die Klausur unbedingt zu beherrschen! Ihr müsst die Higher-Level-Regeln aber nicht auswendig können, wenn ihr davon eine benutzen sollt, wird die angegeben.**

Definition 50 (Syntaktisches Herleiten)**.** Sei α ein Wort aus $\mathrm{AL}(\Pi)$ und M eine Menge von Wörtern aus $\mathrm{AL}(\Pi)$. Wir sagen, $M \vdash \alpha$, gesprochen „α ist aus M herleitbar“, wenn man α durch endlich viele Textersetzungen, die durch unser Kalkül erlaubt sind, unter Benutzung der Aussagen in M hinschreiben kann. Wenn das auch ohne M geht, also einfach durch die Axiome, schreiben wir einfach $\vdash \alpha$. **Eine Herleitung ist also nichts weiter als eine Kettentextersetzung innerhalb Zeichenketten, denen keine Bedeutung zugewiesen ist!**. Hier kommt wieder genau das zeilenweise Beweisprinzip aus unserem Anfangscrashkurs zum tragen.

7.2 Semantik: Logisches Folgern in Formeln mit Bedeutung

In der Syntax haben wir uns eine Menge von Wörtern definiert und durch Textersetzungsregeln mit den Wörtern herumgespielt, die einen Wörter in die anderen überführt, und so weiter. Diesen Wörtern weisen wir jetzt Bedeutung zu: Die einzelnen Variablen aus Π erhalten eine Bewertung, und die Verknüpfungen $\vee, \wedge, \rightarrow, \leftrightarrow$ definieren wir anhand der gängigen Wahrheitstabellen.

Definition 51 (Bewertung)**.** Eine Belegung aller Variablen einer Variablenmenge Π mit konkreten Wahrheitswerten $\in \{W, F\}$. Formal gesehen: Eine Abbildung von Π nach $\{W, F\}$.

So bekommen die Wörter (Aussagen), mit denen wir herumersetzt haben, eine logische Bedeutung und je nach Belegung können sie Wahr sein oder Falsch.

Definition 52 (Gültige und erfüllbare Aussagen, Normalformen)**.** Eine Aussage α ist gültig, wenn sie für jede Bewertung wahr ist (man nennt sie dann auch Tautologie), und erfüllbar, wenn sie für mindestens eine Bewertung wahr ist.

Beachtet bitte auch die Definition und Erzeugungsanleitungen für die Konjunktiven und Disjunktive Normalform von Aussagen, die ein tolles Werkzeug sind, um einfach zu prüfen, ob Aussagen erfüllbar oder gültig sind!

Definition 53 (Logische Äquivalenz)**.** Seien $\alpha, \beta \in \mathrm{AL}(\Pi)$. Dann sind die Ausdrücke logisch äquivalent, geschrieben $\alpha \sim \beta$, wenn sie für alle Bewertungen den gleichen Wert ergeben.

Definition 54 (Logische Folgerung)**.** Ein Ausdruck $\alpha \in \mathrm{AL}(\Pi)$ folgt aus einer Ausdrucksmenge $M \subset \mathrm{AL}(\Pi)$, oder kürzer geschrieben $M \vDash \alpha$, wenn gilt:

$$\forall B : (B(M) = W \Rightarrow B(\alpha) = W)$$

Definition 55 (Kontradiktorische Mengen)**.** ... sind Mengen, für die es ein α gibt mit $M \vdash \alpha$ und gleichzeitig $M \vdash \neg\alpha$.

7.3 Syntaktische und semantische Welt sind eng verbunden!

Wir können nun logisch Aussagen mit Bedeutung folgern, Zeichen $\vDash$ und Texte syntaktisch herleiten durch reines Textersetzen, Zeichen $\vdash$. Beides passiert „zufällig“[3] auf der gleichen Wortmenge, $\mathrm{AL}(\Pi)$. In zwei sehr wichtigen Theoremen haben wir die syntaktische und semantische Welt verbunden und bewiesen, dass wir beides äquivalent verwenden dürfen!

Theorem (Korrektheitssatz des Aussagenkalküls)**.** Was durch reines Kalkültextersetzen hergeleitet werden kann, ist auch inhaltlich wahr: $M \vdash \alpha \Rightarrow M \vDash \alpha$ Wir können also stupide und ohne weiteres Nachdenken textersetzen, und erhalten nur Aussagen, die auch inhaltlich wahr sind!

Theorem (Vollständigkeitssatz der Aussagenlogik)**.** Es kommt noch besser: Alles, was inhaltlich wahr ist, kann auch syntaktisch hergeleitet werden! Es gibt also keine logisch wahre Aussage, die dem Textersetzen unzugänglich ist: $M \vDash \alpha \Rightarrow M \vdash \alpha$.

8 Prädikatenlogik

Prädikatenlogik kann man sich als Erweiterung der Aussagenlogik vorstellen. Insofern kann man beide recht einfach gegenüberstellen, so dass man eine grobe Orientierung hat. Wir springen gleich ins kalte Wasser und machen das mal, dann haben wir schon etwas Grundahnung vom System, bevor wir gleich die Einzelheiten definieren.

[3] Ganz zufällig ist das natürlich nicht, ich will nur veranschaulichen, dass beides im Prinzip voneinander unabhängig passieren kann. In Wirklichkeit haben wir uns in der Vorlesung entschieden, sowohl syntaktische als auch semantische Arbeit auf $\mathrm{AL}(\Pi)$ auszuführen.

Syntax – Buchstaben: In der Syntax der Aussagenlogik bestanden die „Buchstaben“ für aus dem Inhalt einer Menge Π sowie weiteren Zeichen $(,), \neg, \vee$ (und noch ein paar weiteren, die aber einfach Abkürzungen waren). In der Prädikatenlogik-Syntax kommen diese Buchstaben einer Variablenmenge V, aus den Mengen F, R und K einer Signatur $\delta = (S, F, R, K, \text{typ})$ und zusätzlich gibt es noch die Zeichen $\neg, \vee, \bigvee, =, (,)$, da kommen im Vergleich zur Aussagenlogik also ein paar dazu. Später gibt es dann wieder ein paar Abkürzungen. δ ist eine normale Signatur, wie wir das noch von früher gewohnt sind.

Syntax – Wörter: In der Syntax der Aussagenlogik gab es nur eine Art von „Wörtern“: Die Aussagen. In der Prädikatenlogik-Syntax gibt es „prädikatenlogische Ausdrücke“, die nach bestimmten Regeln wiederum „Primformeln“ und „Terme“ beinhalten können. Das ist etwas komplizierter, aber nicht viel.

Semantik: Ausdrücken in der Aussagenlogik haben wir semantisches Leben eingehaucht, indem wir die $p \in \Pi$ als boolsche Variablen angesehen haben sowie $\neg$ und $\vee$ definiert haben. Prädikatenlogischen Ausdrücken haucht man Leben ein, indem man zur Signatur δ eine δ-Struktur eingibt (Wir erinnern uns: Strukturen erfüllen Signaturen mit Leben). Zusätzlich belegt man jede Variable aus V mit einem Wert aus der Trägermenge der δ-Struktur.

Kalküle: Sowohl Aussagenlogik als auch Prädikatenlogik besitzen ein Kalkül aus „Textersetzungsregeln“. Beide Kalküle sind vollständig und korrekt. Beim aussagenlogischen Kalkül gibt es einen Weg um definitiv zu entscheiden, ob Aussagen gültig oder erfüllbar sind, beim prädikatenlogischen Kalkül nicht.

Gängige Umformungen: Aussagenlogische Ausdrücke kann man in konjunktive und disjunktive Normalformen überführen, um Gültigkeit und Entscheidbarkeit einfach festzustellen. Prädikatenlogische Ausdrücke kann man in die Pränexe Normalform überführen. Hierbei ist es wichtig darauf zu achten, welche Variablen durch einen Quantor gebunden sind.

8.1 Syntax: Wieder Verarbeitung von Strings, die zunächst ohne Bedeutung sind

Ich gehe in den Definitionen in dieser Section entlang der Originaldefinitionen, behandele sie aber salopper. Es empfielt sich also, meine Saloppen definitionen und die originalen im Script (S. 160ff) nebeneinanderher zu lesen.

Definition 56 (Terme)**.** Sei V eine Menge von Variablen, und $\delta = (\{s\}, F, R, K, \text{typ})$ eine Signatur, die nur eine einzige Sorte s hat. Dann sind folgende Sachen ein Term:

- ▷ Jede Variable aus V
- ▷ Jede Konstante aus K
- ▷ Falls $t_1, t_2, \ldots, t_n$ Terme sind, und f ein Funktionssymbol aus F ist, dann ist auch $ft_1, t_2, \ldots, t_n$ ein Term (ja, das ist einfach aneinandergeschrieben!)
- ▷ Sonst ist nichts ein Term.

Da es nur eine Sorte s gibt, sind alle Terme vom Typ s.

Definition 57 (Primformeln)**.** Primformeln sind grundlegende logische Ausdrücke in der Prädikatenlogik, die Wahr oder Falsch sind. Wir nennen die Menge der Primformeln daher Π. Das ist eine Analogie zur Aussagenlogik: Da bestand Π aus boolschen Variablen, auch grundlegenden logischen Ausdrücken. Da Π abhängig von V und δ ist, schreiben wir korrekterweise $\Pi(V, \delta)$. Folgende Sachen sind eine Primformel:

- ▷ Wenn t_1 und t_2 Terme sind, dann ist $t_1 = t_2$ eine Primformel.
- ▷ Wenn t_1 und t_2 Terme sind, und P sei ein Relationssymbol aus unserer Signatur δ, dann ist $t_1 P t_2$ eine Primformel (Wieder einfach hintereinandergeschrieben).
- ▷ Sonst nichts.

Primformeln setzen also auf unsere Terme die Relationen auf und sind daher – wie schon gesagt – boolschen typs. Relationen nennen wir hier auch Prädikate.

Definition 58 (Prädikatenlogische Ausdrücke)**.** Bleibt zu definieren, was jetzt alles ein Prädikatenlogisher Ausdruck ist, also zu $\mathrm{PL}(\delta, V)$ gehört. Da wären:

- Einzelne Primformeln
- Wenn α und β PL-Ausdrücke sind, dann auch $\neg\alpha$ und $(\alpha \vee \beta)$ (kennen wir ja schon aus der Aussagenlogik)
- Wenn α PL-Ausdruck und v Variable ist, dann ist auch $\bigvee_v \alpha$ ein PL-Ausdruck (man darf also Quantoren hinzufügen). Achtung: Mein $\bigvee$ hier in LaTeX wird leicht anders layoutet, ihr nehmt natürlich die Variante aus der Vorlesung, wo das v direkt darunter steht.
- Sonst nichts.

Prädikatenlogische Ausdrücke sind natürlich wieder boolschen Typs. Zusätzlich gibt es wieder Abkürzungen, ähnlich zu denen in der Aussagenlogik, siehe Seite 164 unten.

Definition 59 (Freie und gebundene Variablen)**.** Variablen, die unter einem Quantor stehen, sind im Gültigkeitsbereich des Quantors gebunden. Variablen, die nicht gebunden sind, nennen wir frei. Ein einziges Vorkommen einer Variable kann entweder frei sein oder gebunden. Falls aber eine Variable in einem String mehrfach vorkommt, kann sie über die verschiedenen Vorkommen hinweg sowohl frei als auch gebunden sein. Hier muss man also auf die Gültigkeitsbereiche der Quantoren achtgeben (die aber oft durch Klammern definiert werden). Eine schöne, technische, induktive Definition der Menge der freien Variablen finden wir auf Seite 167. Es kann übrigens auch Ausdrücke geben, die nur gebundene (also keine freien) Variablen enthalten. Diese haben dann für alle Interpretationen in derselben Struktur denselben Wahrheitswert (ist ja klar, es gibt ja keine Variablen mehr, die man selbst belegen darf).

Definition 60 (Substitution von Variablen durch andere Terme)**.** Z.B. wenn dieselbe Variable in freier und gebundener Form in einem String vorkommt, kann es aus Übersichtlichkeitsgründen Sinn machen, Variablen zu ersetzen (zu substituieren). Um zu definieren, was durch was ersetzbar ist, haben wir die Relation „subst“ erschaffen (genaue Definition auf Seite 168). Salopp gesagt: Man darf eine Variable x durch einen Term t ersetzen

- falls x gar nicht frei vorkommt, dann verändert sich der gesamte Ausdruck nicht, denn gebundene Variablen werden nicht ersetzt
- falls x frei vorkommt, und t keine Variable enthält, die am Einfügeort von t gebunden würde.

Ein gebundenes x darf also nicht substituiert werden!

Definition 61 (Prädikatenlogisches Kalkül)**.** Auch für unsere prädikatenlogischen Zeichenketten gibt es feste Textersetzungsregeln. Sie bestehen aus denjenigen, die schon aus der Aussagenlogik bekannt sind (vier Axiome, eine Regel). Zusätzlich gibt es drei weitere Axiome und zwei weitere Regeln, die ich hier ebenfalls salopp beschreibe (Originaldefinition nebendran legen!). Neue Axiome:

- $\alpha \to \bigvee_x \alpha$ (Quantoren dürfen vorangesetzt werden)
- $x = x$
- $x = t \to (\alpha \to \beta)$, wenn x in α durch t substituiert werden darf und α dadurch zu β wird, kurz: $\mathrm{subst}(\alpha, x, t, \beta)$.

Neue Regeln:

- Man darf ein $\bigvee_x$ vor $\alpha \to \beta$ setzen, falls x keine freie Variable in β ist.
- Ausdruck α darf durch Ausdruck b ersetzt werden, wenn gilt $\mathrm{subst}(\alpha, x, t, \beta)$

Definition 62 (Herleitbarkeit)**.** Sei $M \subseteq \mathrm{PL}(\delta, V)$ eine Menge von Prädikatenlogischen Ausdrücken und $\alpha \in \mathrm{PL}(\delta, V)$ ein weiterer solcher Ausdruck. Dann ist die Herleitbarkeit $M \vdash \alpha$ völlig analog zu der in der Aussagenlogik definiert: Und wieder geht es nur ums Textersetzen, diesmal natürlich im Prädikatenlogischen Kalkül.

Definition 63 (Quantorenfreiheit, Pränex)**.** Sei α ein prädikatenlogischer Ausdruck. Der Übersichtlichkeit halber wäre es schön, wenn – wie das auch üblich ist – die Quantoren alle gesammelt am Anfang stünden. Diese Form nennt man „pränex“. Zuerst kommen alle Quantoren, und anschließend der Quantorenfreie rest des Ausdrucks. **Das Umformen würde ich für die Klausur beherrschen, eine schöne Anleitung findet ihr auf den Seiten 186ff**.

8.2 Semantik: Füllen prädikatenlogischer Strings mit Bedeutung

Definition 64 (Interpretation prädikatenlogischer Ausdrücke)**.** Um prädikatenlogische Ausdrücke mit Leben zu füllen, muss man

- zur Signatur δ eine δ-Struktur $\mathcal{A}$ angeben, die genau eine Trägermenge hat, auf der alle Konstanten, Funktionen und Prädikate definiert sind
- allen Variablen aus V einen konkreten Wert aus der Trägermenge zuweisen.

Technisch genau erklärt ist die Interpretation auf den Seiten 162ff. Mit der folgenden saloppen erklärung sollte diese Definition ohne weiteres lesbar sein. Dort erschaffen wir eine Funktion, die sukzessive unsere Syntax mit Leben füllt:

- Variablen und Konstanten werden auf Werte in der Trägermenge gesetze
- Funktionssymbole aus der Signatur werden den zugehörigen Funktionen in der Struktur zugeführt
- Das gleiche für Relationssymbole und Relationen (Prädikate), und das $=$ wird auch einer Bedeutung zugeführt
- ... und noch für die logischen Operatoren und den Oderquantor.

8.3 Korrektheit, Vollständigkeit, Entscheidbarkeit

Ohne großartige Beweise oder Anmerkungen hier: Auch das Prädikatenlogische Kalkül ist vollständig und korrekt, auch hier gilt also die Äquivalenz zwischen $\vdash$ und $\models$! Allerdings kann man im Unterschied zur Aussagenlogik nicht einfach so feststellen, ob ein Ausdruck gültig ist, erfüllbar ist, oder überhaupt hergeleitet werden kann! Diese Eigenschaft würde man Entscheidbarkeit nennen, das PL-Kalkül ist also nicht entscheidbar.