5 Logik erster Stufe (Prädikatenlogik)

In Kapitel 3 haben wir bereits die **Aussagenlogik** kennengelernt, die einen Formalismus darstellt, mit dessen Hilfe man "Wissen" modellieren und Schlüsse aus dem Wissen ziehen kann. In diesem Kapitel werden wir die **Logik erster Stufe** (bzw. **Prädikatenlogik**) als einen weiteren solchen Formalismus kennenlernen. Im Vergleich zur Aussagenlogik hat die Prädikatenlogik den Vorteil, dass

- eine klare Trennung zwischen "Daten" einerseits und "Logik" andererseits besteht, und dass in der Prädikatenlogik
- wesentlich umfangreichere Ausdrucksmöglichkeiten zur Verfügung stehen.

Der Preis für diese Vorteile ist allerdings, dass die Prädikatenlogik **algorithmisch** deutlich schwerer zu handhaben ist als die Aussagenlogik.

5.1 Motivation zur Logik erster Stufe

5.1.1 Grenzen der Aussagenlogik

Beispiel 5.1 (Verwandtschaftsbeziehungen)**.** Die Aussagenlogik kann helfen, um Aussagen der Art

> "Anne und Bernd sind Geschwister. Wenn Christine Annes Tochter ist, dann ist Bernd Christines Onkel."

zu modellieren und Schlüsse daraus zu ziehen. Für die Modellierung der folgenden Aussage ist die Aussagenlogik aber eher ungeeignet:

> "Es gibt in Frankfurt mindestens 2 Leute, die mehr als 3 Kinder, aber selbst keine Geschwister haben."

Beispiel 5.2 (Arithmetische Aussagen)**.** Die Aussagenlogik kann helfen, um Sätze der Art

> "Wenn eine Zahl gerade ist, dann ist sie nicht ungerade."

zu formalisieren. Für viele andere Aussagen ist die Aussagenlogik aber eher ungeeignet, zum Beispiel:

> "Es gibt eine Zahl, die nicht Summe zweier Primzahlen ist."

5.1.2 Ein Überblick über die Logik erster Stufe

Die Logik erster Stufe ist ein Formalismus, mit dem man die in den beiden obigen Beispielen genannten Aussagen bequem beschreiben kann. Genau wie die Aussagenlogik besitzt die Logik erster Stufe:

- eine **Syntax**, die festgelegt, welche Zeichenketten Formeln der Logik erster Stufe sind und
- eine **Semantik**, die festlegt, welche "Bedeutung" einzelne Formeln haben.

Die Logik erster Stufe beschäftigt sich mit **Objekten** (z.B. den Einwohnern Frankfurts und deren Verwandtschaftsbeziehungen (Beispiel 5.1) oder den natürlichen Zahlen und deren Addition und Multiplikation (Beispiel 5.2)) und **Aussagen über deren Eigenschaften**. (Im Gegensatz dazu beschäftigt sich die Aussagenlogik nicht mit Objekten sondern lediglich mit "wahren" und "falschen" Aussagen und deren Kombination.)

Vor der Einführung in die Syntax und die Semantik der Logik erster Stufe wenden wir uns zunächst den Objekten zu, über die Formeln der Logik erster Stufe "reden" können.

5.2 Strukturen

Strukturen

Die Objekte, über die Formeln der Logik erster Stufe Aussagen treffen können, heißen **Strukturen**. Viele Objekte lassen sich auf natürliche Weise durch solche Strukturen repräsentieren, beispielsweise

- Graphen $G = (V, E)$
- Bäume $B = (V, E)$
- die natürlichen Zahlen mit Addition und Multiplikation, $(\mathbb{N}, +, \times)$
- die reellen Zahlen mit Addition, Multiplikation und den Konstanten 0 und 1, $(\mathbb{R}, +, \times, 0, 1)$
- Datenbanken

usw. Die im Folgenden definierten **Signaturen** legen den "Typ" (bzw. das "Format") der entsprechenden Strukturen fest.

Signatur
Vokabular
Symbolmenge
Stelligkeit

Definition 5.3. Eine **Signatur** (bzw. ein **Vokabular** bzw. eine **Symbolmenge**; englisch: signature, vocabulary) ist eine Menge σ (in Worten: sigma) von Relationssymbolen, Funktionssymbolen und/oder Konstantensymbolen. Jedes Relationssymbol $\dot{R} \in \sigma$ und jedes Funktionssymbol $\dot{f} \in \sigma$ hat eine **Stelligkeit** (bzw. Arität, engl. arity)

$$\mathrm{ar}(\dot{R}) \in \mathbb{N}_{>0} \quad \text{bzw.} \quad \mathrm{ar}(\dot{f}) \in \mathbb{N}_{>0}.$$

Notation 5.4.

- In diesem Kapitel bezeichnet der griechische Buchstabe σ immer eine Signatur.
- Wir kennzeichnen Symbole aus σ immer mit einem Punkt, wie in $\dot{R}$ bzw. $\dot{f}$.
- Für Relationssymbole verwenden wir meistens Großbuchstaben wie $\dot{R}, \dot{P}, \dot{E}, \dot{R}_1, \dot{R}_2, \ldots$; für Funktionssymbole verwenden wir meistens Kleinbuchstaben wie $\dot{f}, \dot{g}, \dot{h}, \ldots$; für Konstantensymbole verwenden wir meistens Kleinbuchstaben wie $\dot{c}, \dot{d}, \ldots$.
- Gelegentlich verwenden wir als Relations- und Funktionssymbole auch Zeichen wie

$$\begin{aligned} \dot{\leq} &\quad \text{(2-stelliges Relationssymbol)}, \\ \dot{+}, \dot{\times} &\quad \text{(2-stellige Funktionssymbole)}, \\ \dot{0}, \dot{1} &\quad \text{(Konstantensymbole)}. \end{aligned}$$

Definition 5.5. Eine **Struktur über der Signatur** σ (kurz: **σ-Struktur**) ist ein Paar

Struktur
σ-Struktur

$$\mathfrak{A} \;=\; (A, \alpha),$$

bestehend aus:

- einer nicht-leeren Menge A, dem so genannten **Universum** (bzw. **Träger**, engl.: universe, domain) von $\mathfrak{A}$, und

  Universum
  Träger
- einer auf σ definierten Abbildung α, die
  - jedem Relationssymbol $\dot{R} \in \sigma$ eine Relation $\alpha(\dot{R}) \subseteq A^{\mathrm{ar}(\dot{R})}$ der Stelligkeit $\mathrm{ar}(\dot{R})$ zuordnet,
  - jedem Funktionssymbol $\dot{f} \in \sigma$ eine Funktion $\alpha(\dot{f})\colon A^{\mathrm{ar}(\dot{f})} \to A$ zuordnet,
  - jedem Konstantensymbol $\dot{c} \in \sigma$ ein Element $\alpha(\dot{c}) \in A$ zuordnet.

Notation 5.6.

- Strukturen bezeichnen wir meistens mit Fraktur-Buchstaben $\mathfrak{A}, \mathfrak{B}, \mathfrak{G}, \ldots$; das Universum der Strukturen durch die entsprechenden lateinischen Großbuchstaben, also $A, B, G, \ldots$.
- Ist $\mathfrak{A} = (A, \alpha)$ eine σ-Struktur, so schreiben wir für jedes Symbol $\dot{S} \in \sigma$ oft

  $$\dot{S}^{\mathfrak{A}} \quad \text{an Stelle von} \quad \alpha(\dot{S}).$$

  An Stelle von $\mathfrak{A} = (A, \alpha)$ schreiben wir oft auch $\mathfrak{A} = (A, (\dot{S}^{\mathfrak{A}})_{\dot{S} \in \sigma})$, oder falls $\sigma = \{\dot{R}_1, \ldots, \dot{R}_k, \dot{f}_1, \ldots, \dot{f}_l, \dot{c}_1, \ldots, \dot{c}_m\}$ ist,

  $$\mathfrak{A} \;=\; (A, \dot{R}_1^{\mathfrak{A}}, \ldots, \dot{R}_k^{\mathfrak{A}}, \dot{f}_1^{\mathfrak{A}}, \ldots, \dot{f}_l^{\mathfrak{A}}, \dot{c}_1^{\mathfrak{A}}, \ldots, \dot{c}_m^{\mathfrak{A}}).$$

Beispiel 5.7 (Arithmetische Strukturen)**.** Sei $\sigma_{\mathrm{Ar}} := \{\dot{+}, \dot{\times}, \dot{0}, \dot{1}\}$, wobei $\dot{+}$ und $\dot{\times}$ 2-stellige Funktionssymbole und $\dot{0}$ und $\dot{1}$ Konstantensymbole sind. Wir betrachten die σ_{Ar}-Struktur

$$\mathcal{N} \;:=\; (\mathbb{N}, \dot{+}^{\mathcal{N}}, \dot{\times}^{\mathcal{N}}, \dot{0}^{\mathcal{N}}, \dot{1}^{\mathcal{N}}),$$

wobei $\dot{+}^{\mathcal{N}}$ und $\dot{\times}^{\mathcal{N}}$ die natürliche Addition bzw. Multiplikation auf $\mathbb{N}$ sind und $\dot{0}^{\mathcal{N}} := 0$, $\dot{1}^{\mathcal{N}} := 1$. Entsprechend können wir σ_{Ar}-Strukturen $\mathcal{Z}, \mathcal{Q}, \mathcal{R}$ mit Universum $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ definieren.

Beispiel 5.8 (Graphen und Bäume)**.** Sei $\sigma_{\mathrm{Graph}} := \{\dot{E}\}$, wobei $\dot{E}$ ein 2-stelliges Relationssymbol ist. Jeder gerichtete Graph bzw. gerichtete Baum (V, E) lässt sich als σ_{Graph}-Struktur $\mathfrak{A} = (A, E^{\mathfrak{A}})$ mit Universum $A := V$ und Relation $\dot{E}^{\mathfrak{A}} := E$ auffassen.

Beispiel 5.9 (Ordnungen)**.** Sei $\sigma_{\mathrm{Ord}} := \{\dot{\leq}\}$, wobei $\dot{\leq}$ ein 2-stelliges Relationssymbol ist. Jeder Präordnung, partiellen Ordnung oder linearen Ordnung $\leq$ auf einer Menge A entspricht eine σ_{Ord}-Struktur

$$\mathfrak{A} \;=\; (A, \dot{\leq}^{\mathfrak{A}})$$

mit $\dot{\leq}^{\mathfrak{A}} := \leq$.

Frage: Wann sind zwei σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ "prinzipiell gleich" (Fachbegriff: isomorph)?

Antwort: Falls $\mathfrak{B}$ aus $\mathfrak{A}$ entsteht, indem man die Elemente des Universums von $\mathfrak{A}$ umbenennt.

Analog zum Begriff der Isomorphie von Graphen (Definition 4.25) wird dies durch folgende Definition präzisiert:

isomorph

Definition 5.10. Sei σ eine Signatur und seien $\mathfrak{A}$ und $\mathfrak{B}$ zwei σ-Strukturen. $\mathfrak{A}$ und $\mathfrak{B}$ heißen **isomorph** (kurz: $\mathfrak{A} \cong \mathfrak{B}$, in Worten: $\mathfrak{A}$ ist isomorph zu $\mathfrak{B}$), falls es eine **bijektive** Abbildung $\pi\colon A \to B$ gibt, für die gilt:

- für jedes Relationssymbol $\dot{R} \in \sigma$, für $r := \mathrm{ar}(\dot{R})$ und für alle r-Tupel $(a_1, \ldots, a_r) \in A^r$ gilt:

$$(a_1, \ldots, a_r) \in \dot{R}^{\mathfrak{A}} \iff \big(\pi(a_1), \ldots, \pi(a_r)\big) \in \dot{R}^{\mathfrak{B}}.$$

- für jedes Konstantensymbol $\dot{c} \in \sigma$ gilt:

$$\pi(\dot{c}^{\mathfrak{A}}) \;=\; \dot{c}^{\mathfrak{B}}.$$

- für jedes Funktionssymbol $\dot{f} \in \sigma$, für $r := \mathrm{ar}(\dot{f})$ und für alle r-Tupel $(a_1, \ldots, a_r) \in A^r$ gilt:

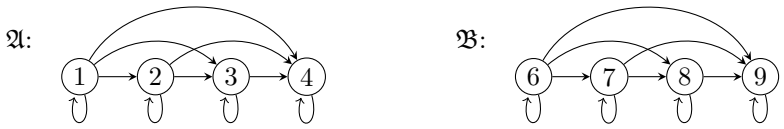
$$\pi\big(\dot{f}^{\mathfrak{A}}(a_1, \ldots, a_r)\big) \;=\; \dot{f}^{\mathfrak{B}}\big(\pi(a_1), \ldots, \pi(a_r)\big).$$

Isomorphismus

Eine solche Abbildung π wird **Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$** genannt.

Beispiel 5.11.

(a) Ist $A = \{1, 2, 3, 4\}, B = \{6, 7, 8, 9\}$, und sind $\dot{\leq}^{\mathfrak{A}}$ und $\dot{\leq}^{\mathfrak{B}}$ die natürlichen linearen Ordnungen auf A und B, so sind die beiden σ_{Ord}-Strukturen $\mathfrak{A} = (A, \dot{\leq}^{\mathfrak{A}})$ und $\mathfrak{B} = (B, \dot{\leq}^{\mathfrak{B}})$ isomorph.
Skizze:



Allgemein gilt: Sind A und B endliche Mengen mit $|A| = |B|$ und sind $\dot{\leq}^{\mathfrak{A}}$ und $\dot{\leq}^{\mathfrak{B}}$ lineare Ordnungen auf $\mathfrak{A} = (A, \dot{\leq}^{\mathfrak{A}})$ und $\mathfrak{B} = (B, \dot{\leq}^{\mathfrak{B}})$, so ist $\mathfrak{A} \cong \mathfrak{B}$, und die Abbildung π, die das (bzgl. $\dot{\leq}^{\mathfrak{A}}$) kleinste Element in A auf das (bzgl. $\dot{\leq}^{\mathfrak{B}}$) kleinste Element von $\mathfrak{B}$ abbildet und, allgemein, für jedes $i \in \{1, \ldots, |A|\}$ das i-kleinste Element in A (bzgl. $\dot{\leq}^{\mathfrak{A}}$) auf das i-kleinste Element in B (bzgl. $\dot{\leq}^{\mathfrak{B}}$) abbildet, ein Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

(b) Sind $\dot{\leq}^{\mathcal{N}}$ und $\dot{\leq}^{\mathcal{Z}}$ die natürlichen linearen Ordnungen auf $\mathbb{N}$ und $\mathbb{Z}$, so sind die σ_{Ord}-Strukturen $\mathcal{N} := (\mathbb{N}, \dot{\leq}^{\mathcal{N}})$ und $\mathcal{Z} := (\mathbb{Z}, \dot{\leq}^{\mathcal{Z}})$ **nicht isomorph** (kurz: $\mathcal{N} \ncong \mathcal{Z}$).
Skizze:

(c) Sei $\sigma := \{\dot{f}, \dot{c}\}$, wobei $\dot{f}$ ein 2-stelliges Funktionssymbol und $\dot{c}$ ein Konstantensymbol ist. Sei $\mathfrak{A} := (A, \dot{f}^{\mathfrak{A}}, \dot{c}^{\mathfrak{A}})$, wobei

- $A := \mathbb{N}$
- $\dot{f}^{\mathfrak{A}} := \dot{+}^{\mathcal{N}}$ die Addition auf $\mathbb{N}$
- $\dot{c}^{\mathfrak{A}} := \dot{0}^{\mathcal{N}}$ die natürliche Zahl 0 ist

und sei $\mathfrak{B} := (B, \dot{f}^{\mathfrak{B}}, \dot{c}^{\mathfrak{B}})$, wobei

- $B := \{2^n : n \in \mathbb{N}\}$ die Menge aller Zweierpotenzen
- $\dot{f}^{\mathfrak{B}} \colon B \times B \to B$ die Funktion mit

$$\dot{f}^{\mathfrak{B}}(b_1, b_2) := b_1 \cdot b_2, \quad \text{f.a. } b_1, b_2 \in B$$

- $\dot{c}^{\mathfrak{B}} := 1 = 2^0 \in B$.

Dann gilt: $\mathfrak{A} \cong \mathfrak{B}$, und die Abbildung $\pi \colon A \to B$ mit $\pi(n) := 2^n$, f.a. $n \in \mathbb{N}$, ist ein Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$, denn:

- π ist eine bijektive Abbildung von A nach B.
- Für das Konstantensymbol $\dot{c} \in \sigma$ gilt:

$$\pi(\dot{c}^{\mathfrak{A}}) \overset{\text{Def. } \dot{c}^{\mathfrak{A}}}{=} \pi(0) \overset{\text{Def. } \pi}{=} 2^0 \overset{\text{Def. } \dot{c}^{\mathfrak{B}}}{=} \dot{c}^{\mathfrak{B}}.$$

- Für das Funktionssymbol $\dot{f} \in \sigma$ und für alle $(a_1, a_2) \in A^2$ gilt:

$$\pi\big(\dot{f}^{\mathfrak{A}}(a_1, a_2)\big) \overset{\text{Def. } \dot{f}^{\mathfrak{A}}}{=} \pi(a_1 + a_2) \overset{\text{Def. } \pi}{=} 2^{a_1 + a_2}$$

und

$$\dot{f}^{\mathfrak{B}}\big(\pi(a_1), \pi(a_2)\big) \overset{\text{Def. } \pi}{=} \dot{f}^{\mathfrak{B}}(2^{a_1}, 2^{a_2}) \overset{\text{Def. } \dot{f}^{\mathfrak{B}}}{=} 2^{a_1} \cdot 2^{a_2} = 2^{a_1 + a_2}.$$

Also: $\pi\big(\dot{f}^{\mathfrak{A}}(a_1, a_2)\big) = \dot{f}^{\mathfrak{B}}\big(\pi(a_1), \pi(a_2)\big)$. Somit ist π ein Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

Wir wissen nun, über welche Objekte Formeln der Logik erster Stufe "reden" können: über σ-Strukturen, wobei σ eine Signatur ist. Als nächstes legen wir die Syntax der Logik erster Stufe fest.

5.3 Syntax der Logik erster Stufe

Die Logik erster Stufe übernimmt, verändert und erweitert die Syntax der Aussagenlogik.

- Was gleich bleibt:
  - Alle Junktoren $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ werden übernommen.
- Was sich verändert:
  - Variablen stehen nicht mehr für "wahre" oder "falsche" Aussagen, sondern für Elemente im Universum einer σ-Struktur.
  - Variablen sind keine atomaren Formeln mehr.
- Was neu hinzukommt:
  - Es gibt **Quantoren** $\exists$ (für "es existiert") und $\forall$ (für "für alle").
  - Es gibt Symbole für Elemente aus der Signatur σ.

Definition 5.12 (Variablen und Alphabet der Logik erster Stufe)**.**

Individuenvariable
Variable

(a) Eine **Individuenvariable** (kurz: **Variable**) hat die Form v_i, für $i \in \mathbb{N}$.
Die Menge aller Variablen bezeichnen wir mit VAR, d.h. $\text{VAR} = \{v_i : i \in \mathbb{N}\}$.

Alphabet A_σ

(b) Sei σ eine Signatur. Das **Alphabet** A_σ der **Logik erster Stufe über** σ besteht aus:

- den Variablen in VAR
- den Symbolen in σ
- den Quantoren $\exists$ (Existenzquantor) und $\forall$ (Allquantor)
- dem Gleichheitssymbol $\doteq$
- den Junktoren $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- den Klammern $(,)$ und dem Komma $,$

D.h.

$$A_\sigma \;=\; \text{VAR} \;\cup\; \sigma \;\cup\; \{\exists, \forall\} \;\cup\; \{\doteq\} \;\cup\; \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\} \;\cup\; \{,\}.$$

σ-Terme

Definition 5.13 (Terme der Logik erster Stufe)**.** Sei σ eine Signatur. Die Menge T_σ der **σ-Terme** ist die folgendermaßen rekursiv definierte Teilmenge von A_σ^*:

Basisregeln:

- Für jedes Konstantensymbol $\dot{c} \in \sigma$ ist $\dot{c} \in T_\sigma$.
- Für jede Variable $x \in \text{VAR}$ ist $x \in T_\sigma$.

Rekursive Regeln:

- Für jedes Funktionssymbol $\dot{f} \in \sigma$ und für $r := \text{ar}(\dot{f})$ gilt: Sind $t_1 \in T_\sigma, \ldots, t_r \in T_\sigma$, so ist auch $\dot{f}(t_1, \ldots, t_r) \in T_\sigma$.

Beispiel 5.14. Sei $\sigma = \{\dot{f}, \dot{c}\}$ die Signatur aus Beispiel 5.11(c), die aus einem 2-stelligen Funktionssymbol $\dot{f}$ und einem Konstantensymbol $\dot{c}$ besteht.

Folgende Worte aus A_σ^* sind σ-Terme:

$$\dot{c}, \quad v_4, \quad \dot{f}(\dot{c}, \dot{c}), \quad \dot{f}(\dot{c}, v_0), \quad \dot{f}\big(\dot{c}, \dot{f}(\dot{c}, v_0)\big).$$

Folgende Worte sind keine σ-Terme:

$$\mathbf{0}, \quad \dot{f}(\mathbf{0}, \dot{c}), \quad \dot{f}(v_0, \dot{c}, v_1), \quad f^{\mathfrak{A}}(2, 3).$$

FO[σ]-Formeln

Definition 5.15 (Formeln der Logik erster Stufe)**.** Sei σ eine Signatur. Die Menge FO[σ] aller **Formeln der Logik erster Stufe über der Signatur** σ (kurz: **FO[σ]-Formeln**; FO steht für die englische Bezeichnung der Logik erster Stufe: first-order logic) ist die folgendermaßen rekursiv definierte Teilmenge von A_σ^*:

Basisregeln:

- Für alle σ-Terme t_1 und t_2 in T_σ gilt:

$$t_1 \doteq t_2 \;\in\; \text{FO}[\sigma].$$

- Für jedes Relationssymbol $\dot{R} \in \sigma$, für $r := \text{ar}(\dot{R})$ und für alle σ-Terme $t_1, \ldots, t_r$ in T_σ gilt:

$$\dot{R}(t_1, \ldots, t_r) \;\in\; \text{FO}[\sigma].$$

Bemerkung. FO[σ]-Formeln der Form $t_1 \doteq t_2$ oder $\dot{R}(t_1, \ldots, t_r)$ heißen **atomare σ-Formeln**. atomare σ-Formeln

Rekursive Regeln:

- Ist $\varphi \in \text{FO}[\sigma]$, so auch $\neg\varphi \in \text{FO}[\sigma]$.
- Ist $\varphi \in \text{FO}[\sigma]$ und $\psi \in \text{FO}[\sigma]$, so ist auch
  - $(\varphi \wedge \psi) \in \text{FO}[\sigma]$
  - $(\varphi \vee \psi) \in \text{FO}[\sigma]$
  - $(\varphi \to \psi) \in \text{FO}[\sigma]$
  - $(\varphi \leftrightarrow \psi) \in \text{FO}[\sigma]$.
- Ist $\varphi \in \text{FO}[\sigma]$ und ist $x \in \text{VAR}$, so ist auch
  - $\exists x\, \varphi \in \text{FO}[\sigma]$
  - $\forall x\, \varphi \in \text{FO}[\sigma]$.

Beispiel 5.16.

(a) Sei $\sigma = \{\dot{f}, \dot{c}\}$ die Signatur aus Beispiel 5.11(c), die aus einem 2-stelligen Funktionssymbol $\dot{f}$ und einem Konstantensymbol $\dot{c}$ besteht.

Folgende Worte aus A_σ^* sind FO[σ]-Formeln:

- $\dot{f}(v_0, v_1) \doteq \dot{c}$ (atomare σ-Formel)
- $\forall v_2\, \dot{f}(v_2, \dot{c}) \doteq v_2$
- $\neg\exists v_3 \big(\dot{f}(v_3, v_3) \doteq v_3 \wedge \neg v_3 \doteq \dot{c}\big)$

Folgende Worte sind **keine** FO[σ]-Formeln:

- $\big(\dot{f}(v_0, v_1) \doteq \dot{c}\big)$
- $\Big(\forall v_2 \big(\dot{f}(v_2, \dot{c}) \doteq v_2\big)\Big)$
- $\exists \dot{c}\, \dot{f}(v_0, \dot{c}) \doteq v_0$

(b) Sei $\sigma_{\text{Graph}} = \{\dot{E}\}$ die Signatur, die aus einem 2-stelligen Relationssymbol besteht. Folgendes ist eine FO[σ_{Graph}]-Formel:

$$\forall v_0 \forall v_1 \Big(\big(\dot{E}(v_0, v_1) \wedge \dot{E}(v_1, v_0)\big) \to v_0 \doteq v_1 \Big).$$

Intuition zur Semantik: In einem Graphen $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ sagt diese Formel folgendes aus:

> "für alle Knoten $a_0 \in A$ und
> für alle Knoten $a_1 \in A$ gilt:
> falls $(a_0, a_1) \in \dot{E}^{\mathfrak{A}}$ und $(a_1, a_0) \in \dot{E}^{\mathfrak{A}}$, so ist $a_0 = a_1$."

Die Formel sagt in einem Graph $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ also gerade aus, dass die Kantenrelation $\dot{E}^{\mathfrak{A}}$ antisymmetrisch ist (vgl. Definition 4.64). D.h.: Ein Graph $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ **erfüllt** die Formel genau dann, wenn die Kantenrelation $\dot{E}^{\mathfrak{A}}$ antisymmetrisch ist.

Notation 5.17.

- Statt mit $v_0, v_1, v_2, \ldots$ bezeichnen wir Variablen oft auch mit $x, y, z, \ldots$ oder mit Varianten wie $x', y_1, y_2, \ldots$.

- Für gewisse 2-stellige Funktionssymbole wie $\dot{+}, \dot{\times} \in \sigma_{\mathrm{Ar}}$ oder $\dot{\leq} \in \sigma_{\mathrm{Ord}}$ verwenden wir **Infix- statt Präfixschreibweise** und setzen Klammern dabei auf natürliche Weise, um die eindeutige Lesbarkeit zu gewährleisten.

  Beispiel: An Stelle des (formal korrekten) Terms $\dot{\times}(\dot{+}(x, y), z)$ schreiben wir $(x \dot{+} y) \dot{\times} z$. An Stelle der (formal korrekten) atomaren Formel $\dot{\leq}(x, y)$ schreiben wir $x \dot{\leq} y$.

Wir wissen nun, welche Zeichenketten (über dem Alphabet A_σ) **FO[σ]-Formeln** genannt werden (Syntax). Bevor wir die formale Definition der **Semantik** angeben, betrachten wir zunächst einige Beispiele, um ein "intuitives Verständnis" der Semantik zu bekommen.

5.4 Beispiele zur Semantik der Logik erster Stufe

Beispiel 5.18 (gerichtete Graphen)**.** Sei $\sigma_{\mathrm{Graph}} = \{\dot{E}\}$, wobei $\dot{E}$ ein 2-stelliges Relationssymbol ist.

(a) Die FO[σ_{Graph}]-Formel

$$\varphi \;:=\; \forall x \forall y \big(\dot{E}(x, y) \to \dot{E}(y, x)\big)$$

besagt:

> "Für alle Knoten x und für alle Knoten y gilt: Falls es eine Kante von x nach y gibt, so gibt es auch eine Kante von y nach x."

Für jeden Graphen $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ gilt daher:

$$\mathfrak{A} \text{ erfüllt } \varphi \iff \dot{E}^{\mathfrak{A}} \text{ ist symmetrisch.}$$

Umgangssprachlich sagen wir auch: "Die Formel φ sagt **in einem Graphen $\mathfrak{A}$ aus, dass** dessen Kantenrelation symmetrisch ist."

(b) Die folgende FO[σ_{Graph}]-Formel drückt aus, dass es von Knoten x zu Knoten y einen Weg der Länge 3 gibt:

$$\varphi(x, y) \;:=\; \exists z_1 \exists z_2 \Big(\big(\dot{E}(x, z_1) \wedge \dot{E}(z_1, z_2)\big) \wedge \dot{E}(z_2, y) \Big).$$

(c) Die FO[σ_{Graph}]-Formel

$$\forall x \forall y \exists z_1 \exists z_2 \Big(\big(\dot{E}(x, z_1) \wedge \dot{E}(z_1, z_2)\big) \wedge \dot{E}(z_2, y) \Big)$$

sagt in einem Graphen $\mathfrak{A}$ aus, dass es zwischen je 2 Knoten einen Weg der Länge 3 gibt.

Beispiel 5.19 (Verwandtschaftsbeziehungen)**.** Um Verwandtschaftsbeziehungen zu modellieren, können wir die Symbolmenge σ benutzen, die aus den folgenden Symbolen besteht:

- 1-stellige Funktionen $\dot{Vater}$, $\dot{Mutter}$

  (Bedeutung: $x \doteq \dot{Vater}(y)$ besagt "x ist der Vater von y".)

- 2-stellige Relationen $\dot{Geschwister}$, $\dot{Vorfahr}$

  (Bedeutung: $\dot{Geschwister}(x, y)$ besagt, dass x und y Geschwister sind; $\dot{Vorfahr}(x, y)$ besagt, dass x ein Vorfahr von y ist.)

Generelles Wissen über Verwandtschaftsbeziehungen lässt sich durch Formeln der Logik erster Stufe repräsentieren, beispielsweise:

- "Personen mit gleichem Vater und gleicher Mutter sind Geschwister":

$$\forall x \forall y \Big(\big(\dot{Vater}(x) \doteq \dot{Vater}(y) \wedge \dot{Mutter}(x) \doteq \dot{Mutter}(y) \big) \rightarrow \dot{Geschwister}(x, y) \Big).$$

- "Eltern sind gerade die unmittelbaren Vorfahren":

$$\forall x \forall y \Bigg(\big(x \doteq \dot{Vater}(y) \vee x \doteq \dot{Mutter}(y) \big) \leftrightarrow \Big(\dot{Vorfahr}(x, y) \wedge \neg \exists z \big(\dot{Vorfahr}(x, z) \wedge \dot{Vorfahr}(z, y) \big) \Big) \Bigg).$$

- "Die Relation $\dot{Vorfahr}$ ist transitiv":

$$\forall x \forall y \forall z \Big(\big(\dot{Vorfahr}(x, y) \wedge \dot{Vorfahr}(y, z) \big) \rightarrow \dot{Vorfahr}(x, z) \Big).$$

- Die folgende Formel $\varphi(x, y)$ besagt, dass x Tante oder Onkel von y ist:

$$\varphi(x, y) \;:=\; \exists z \Big(\dot{Geschwister}(x, z) \wedge \big(z \doteq \dot{Vater}(y) \vee z \doteq \dot{Mutter}(y) \big) \Big).$$

- Die folgende Formel $\psi(x)$ besagt, dass x Vater von genau 2 Kindern ist:

$$\psi(x) \;:=\; \exists y_1 \exists y_2 \Bigg(\Big(\big(x \doteq \dot{Vater}(y_1) \wedge x \doteq \dot{Vater}(y_2) \big) \wedge \neg y_1 \doteq y_2 \Big) \wedge \forall z \big(x \doteq \dot{Vater}(z) \rightarrow (z \doteq y_1 \vee z \doteq y_2) \big) \Bigg).$$

5.5 Semantik der Logik erster Stufe

Um die formale Definition der Semantik der Logik erster Stufe angeben zu können, benötigen wir noch folgende Begriffe:

Notation 5.20.

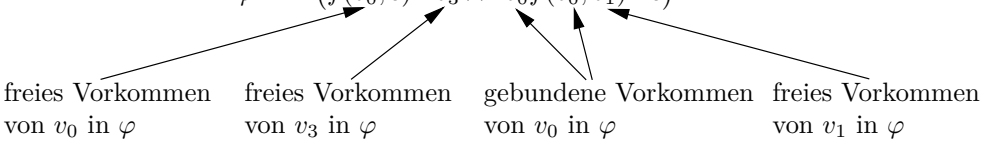
1. Eine Formel ψ ist **Teilformel** einer Formel φ, wenn ψ als Teil-Wort in φ vorkommt. Teilformel

   Beispiel: $\psi := \dot{f}(v_0, v_1) \doteq \dot{c}$ ist Teilformel der Formel $\exists v_0 \, \dot{f}(v_0, v_1) \doteq \dot{c}$.

2. Ist φ eine Formel und x eine Variable, so heißt jedes Vorkommen von x in einer Teilformel der Form $\exists x \, \psi$ oder $\forall x \, \psi$ **gebunden**. Jedes andere Vorkommen von x in φ heißt **frei**. gebunden frei

Beispiel:

$$\varphi \;:=\; \big(\dot f(v_0,\dot c)\doteq v_3 \wedge \exists v_0 \dot f(v_0,v_1)\doteq \dot c\big)$$



frei(φ)
freien Variablen

3. Die Menge frei(φ) aller **freien Variablen** einer FO[σ]-Formel φ besteht aus allen Variablen, die mindestens einmal frei in φ vorkommen.

   Beispiel:

   - $\text{frei}\big(\dot f(v_0,\dot c)\doteq v_3\big) = \{v_0, v_3\}$
   - $\text{frei}\big(\exists v_0\, \dot f(v_0,v_1)\doteq \dot c\big) = \{v_1\}$
   - $\text{frei}\big(\dot f(v_0,\dot c)\doteq v_3 \wedge \exists v_0\, \dot f(v_0,v_1)\doteq \dot c\big) = \{v_0, v_3, v_1\}$

Satz

4. Eine FO[σ]-Formel φ heißt **Satz** (genauer: FO[σ]-Satz), falls sie keine freien Variablen besitzt, d.h. falls $\text{frei}(\varphi) = \varnothing$.

Definition 5.21 (Belegung und Interpretation)**.**

Belegung

(a) Eine **Belegung** in einer σ-Struktur $\mathfrak{A} = (A, \alpha)$ ist eine partielle Funktion β von VAR nach A (d.h. β ordnet jeder Variablen $x \in \text{Def}(\beta)$ ein Element $\beta(x)$ aus dem Universum von $\mathfrak{A}$ zu).

Belegung für eine FO[σ]-Formel φ
passend zu φ

(b) Eine Belegung β ist eine **Belegung für eine FO[σ]-Formel** φ (bzw. **passend zu** φ), wenn $\text{frei}(\varphi) \subseteq \text{Def}(\beta)$.

σ-Interpretation

(c) Eine **σ-Interpretation** ist ein Paar

$$\mathcal{I} \;=\; (\mathfrak{A}, \beta)$$

bestehend aus einer σ-Struktur $\mathfrak{A}$ und einer Belegung β in $\mathfrak{A}$. $\mathcal{I} = (\mathfrak{A}, \beta)$ ist eine **Interpretation für eine FO[σ]-Formel** φ (bzw. passend zu φ), wenn β passend zu φ ist.

Interpretation für eine FO[σ]-Formel

Definition 5.22 (Semantik von σ-Termen)**.** Sei σ eine Signatur. Rekursiv über den Aufbau von T_σ definieren wir eine Funktion $[\![\cdot]\!]^{\cdot}$, die jedem σ-Term $t \in T_\sigma$ und jeder σ-Interpretation $\mathcal{I} = (\mathfrak{A}, \beta)$, so dass $\text{Def}(\beta)$ jede in t vorkommende Variable enthält, einen Wert $[\![t]\!]^{\mathcal{I}} \in A$ zuordnet:

- Für alle $x \in \text{VAR}$ ist $[\![x]\!]^{\mathcal{I}} \;:=\; \beta(x)$.
- Für alle Konstantensymbole $\dot c \in \sigma$ ist $[\![\dot c]\!]^{\mathcal{I}} \;:=\; \dot c^{\mathfrak{A}}$.
- Für alle Funktionssymbole $\dot f \in \sigma$, für $r := \text{ar}(\dot f)$ und für alle σ-Terme $t_1, \ldots, t_r \in T_\sigma$ gilt:

$$[\![\dot f(t_1, \ldots, t_r)]\!]^{\mathcal{I}} \;:=\; \dot f([\![t_1]\!]^{\mathcal{I}}, \ldots, [\![t_r]\!]^{\mathcal{I}}).$$

Beispiel 5.23. Sei $\sigma = \{\dot f, \dot c\}$ und sei $\mathfrak{A} = (A, \dot f^{\mathfrak{A}}, \dot c^{\mathfrak{A}})$ mit

- $A \;:=\; \mathbb{N}$
- $\dot f^{\mathfrak{A}} \;:=\; \dot +^{\mathcal{N}}$ die Addition auf $\mathbb{N}$
- $\dot c^{\mathfrak{A}} \;:=\; \dot 0^{\mathcal{N}}$ die natürliche Zahl 0

wie im Beispiel 5.11(c). Sei β eine Belegung mit $\beta(v_1) = 1$ und $\beta(v_2) = 7$. Und sei $\mathcal{I} := (\mathfrak{A}, \beta)$. Sei t der Term $\dot{f}\big(v_2, \dot{f}(v_1, \dot{c})\big)$. Dann gilt:

$$\begin{aligned}
[\![t]\!]^{\mathcal{I}} &= [\![\dot{f}\big(v_2, \dot{f}(v_1, \dot{c})\big)]\!]^{\mathcal{I}} \\
&= \dot{f}^{\mathfrak{A}}\big([\![v_2]\!]^{\mathcal{I}}, [\![\dot{f}(v_1, \dot{c})]\!]^{\mathcal{I}}\big) \\
&\overset{\dot{f}^{\mathfrak{A}} = \text{Addition auf } \mathbb{N}}{=} [\![v_2]\!]^{\mathcal{I}} + [\![\dot{f}(v_1, \dot{c})]\!]^{\mathcal{I}} \\
&= \beta(v_2) + \dot{f}^{\mathfrak{A}}([\![v_1]\!]^{\mathcal{I}}, [\![\dot{c}]\!]^{\mathcal{I}}) \\
&= \beta(v_2) + ([\![v_1]\!]^{\mathcal{I}} + [\![\dot{c}]\!]^{\mathcal{I}}) \\
&= \beta(v_2) + (\beta(v_1) + \dot{c}^{\mathfrak{A}}) \\
&\overset{\text{Def. } \beta \text{ und } \dot{c}^{\mathfrak{A}}}{=} 7 + (1 + 0) \\
&= 8.
\end{aligned}$$

Notation 5.24.

- Ist β eine Belegung in einer σ-Struktur $\mathfrak{A}$, ist $x \in \text{VAR}$ und ist $a \in A$, so sei

  $$\beta\frac{a}{x}$$

  die Belegung mit $\text{Def}(\beta\frac{a}{x}) := \text{Def}(\beta) \cup \{x\}$, die für alle $y \in \text{Def}(\beta\frac{a}{x})$ definiert ist durch

  $$\beta\frac{a}{x}(y) \;:=\; \begin{cases} a, & \text{falls } y = x \\ \beta(y) & \text{sonst.} \end{cases}$$

- Ist $\mathcal{I} = (\mathfrak{A}, \beta)$ eine σ-Interpretation, ist $x \in \text{VAR}$ und ist $a \in A$, so sei

  $$\mathcal{I}\frac{a}{x} \;:=\; (\mathfrak{A}, \beta\frac{a}{x}).$$

Wir können nun (endlich) die formale Semantik der Logik erster Stufe festlegen.

Definition 5.25 (Semantik der Logik erster Stufe)**.** Sei σ eine Signatur. Rekursiv über den Aufbau von FO[σ] definieren wir eine Funktion $[\![\cdot]\!]^{\cdot}$, die jeder FO[σ]-Formel φ und jeder zu φ passenden Interpretationen $\mathcal{I} = (\mathfrak{A}, \beta)$ einen **Wahrheitswert** (kurz: **Wert**) $[\![\varphi]\!]^{\mathcal{I}} \in \{0, 1\}$ zuordnet: Wahrheitswert

Rekursionsanfang:

- Für alle σ-Terme t_1 und t_2 in T_σ gilt:

  $$[\![t_1 \doteq t_2]\!]^{\mathcal{I}} \;:=\; \begin{cases} 1, & \text{falls } [\![t_1]\!]^{\mathcal{I}} = [\![t_2]\!]^{\mathcal{I}} \\ 0, & \text{sonst.} \end{cases}$$

- Für jedes Relationssymbol $\dot{R} \in \sigma$, für $r := \text{ar}(\dot{R})$ und für alle σ-Terme $t_1, \ldots, t_r$ in T_σ gilt:

  $$[\![\dot{R}(t_1, \ldots, t_r)]\!]^{\mathcal{I}} \;:=\; \begin{cases} 1, & \text{falls } ([\![t_1]\!]^{\mathcal{I}}, \ldots, [\![t_r]\!]^{\mathcal{I}}) \in \dot{R}^{\mathfrak{A}} \\ 0, & \text{sonst.} \end{cases}$$

Rekursionsschritt:

- Die Semantik der Junktoren $\neg$, $\wedge$, $\vee$, $\rightarrow, \leftrightarrow$ ist wie in der Aussagenlogik definiert – beispielsweise ist für $\varphi \in \mathrm{FO}[\sigma]$ und $\psi \in \mathrm{FO}[\sigma]$:

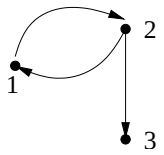
$$[\![(\varphi \wedge \psi)]\!]^{\mathcal{I}} := \begin{cases} 1, & \text{falls } [\![\varphi]\!]^{\mathcal{I}} = 1 \text{ und } [\![\psi]\!]^{\mathcal{I}} = 1 \\ 0, & \text{sonst.} \end{cases}$$

- Ist $\varphi \in \mathrm{FO}[\sigma]$ und ist $x \in \textsc{Var}$, so ist
  - $[\![\exists x\, \varphi]\!]^{\mathcal{I}} := \begin{cases} 1, & \text{falls es (mindestens) ein } a \in A \text{ gibt, so dass } [\![\varphi]\!]^{\mathcal{I}\frac{a}{x}} = 1 \\ 0, & \text{sonst.} \end{cases}$
  - $[\![\forall x\, \varphi]\!]^{\mathcal{I}} := \begin{cases} 1, & \text{falls für alle } a \in A \text{ gilt: } [\![\varphi]\!]^{\mathcal{I}\frac{a}{x}} = 1 \\ 0, & \text{sonst.} \end{cases}$

Beispiel 5.26. Sei $\sigma_{\mathrm{Graph}} = \dot{E}$ die Signatur, die aus einem 2-stelligen Relationssymbol $\dot{E}$ besteht. Betrachte die $\mathrm{FO}[\sigma]$-Formel

$$\varphi := \forall x \forall y \big(\dot{E}(x,y) \rightarrow \dot{E}(y,x)\big).$$

Sei $\mathfrak{A}$ die σ_{Graph}-Struktur, die den gerichteten Graphen



repräsentiert, d.h. $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ mit $A = \{1,2,3\}$ und $\dot{E}^{\mathfrak{A}} = \{(1,2),(2,1),(2,3)\}$. Sei β die Belegung mit leerem Definitionsbereich und sei $\mathcal{I} := (\mathfrak{A}, \beta)$. Dann gilt:

$$\begin{aligned}
[\![\varphi]\!]^{\mathcal{I}} = 1 &\iff \text{für alle } a \in A \text{ gilt: } [\![\forall y\big(\dot{E}(x,y) \rightarrow \dot{E}(y,x)\big)]\!]^{\mathcal{I}\frac{a}{x}} = 1 \\
&\iff \text{für alle } a \in A \text{ gilt:} \\
&\qquad \text{für alle } b \in A \text{ gilt: } [\![\big(\dot{E}(x,y) \rightarrow \dot{E}(y,x)\big)]\!]^{\mathcal{I}\frac{a}{x}\frac{b}{y}} = 1 \\
&\iff \text{für alle } a \in A \text{ und alle } b \in A \text{ gilt:} \\
&\qquad \text{falls } [\![\dot{E}(x,y)]\!]^{\mathcal{I}\frac{a}{x}\frac{b}{y}} = 1, \text{ so auch } [\![\dot{E}(y,x)]\!]^{\mathcal{I}\frac{a}{x}\frac{b}{y}} = 1 \\
&\iff \text{für alle } a \in A \text{ und alle } b \in A \text{ gilt:} \\
&\qquad \text{falls } (a,b) \in \dot{E}^{\mathfrak{A}}, \text{ so auch } (b,a) \in \dot{E}^{\mathfrak{A}} \\
&\iff \dot{E}^{\mathfrak{A}} \text{ ist symmetrisch, vgl. Def. 4.64.}
\end{aligned}$$

Da in unserem konkreten Graphen $\mathfrak{A}$ für $a = 2$ und $b = 3$ gilt: $(a,b) \in \dot{E}^{\mathfrak{A}}$, aber $(b,a) \notin \dot{E}^{\mathfrak{A}}$, ist hier $[\![\varphi]\!]^{\mathcal{I}} = 0$.

Notation 5.27. Sei σ eine Signatur und sei φ eine $\mathrm{FO}[\sigma]$-Formel.

- Ist $\mathcal{I} = (\mathfrak{A}, \beta)$ eine zu φ passende σ-Interpretation, so sagen wir "$\boldsymbol{\mathcal{I}}$ **erfüllt** $\boldsymbol{\varphi}$" (bzw. "$\boldsymbol{\mathcal{I}}$ **ist ein Modell von** $\boldsymbol{\varphi}$", kurz: $\boldsymbol{\mathcal{I} \models \varphi}$), falls $[\![\varphi]\!]^{\mathcal{I}} = 1$.
- Ist φ ein **Satz** (d.h. φ hat keine freien Variablen), so hängt die Tatsache, ob φ von einer Interpretation $\mathcal{I} = (\mathfrak{A}, \beta)$ erfüllt wird, nur von der Struktur $\mathfrak{A}$ und nicht von der Belegung β ab. An Stelle von "$\mathcal{I} \models \varphi$" schreiben wir dann kurz "$\mathfrak{A} \models \varphi$" und sagen "die σ-Struktur $\mathfrak{A}$ erfüllt den Satz φ."

5.5.1 Erfüllbarkeit, Allgemeingültigkeit, Folgerung und Äquivalenz

Definition 5.28. Sei σ eine Signatur und sei φ eine FO[σ]-Formel.

(a) φ heißt **erfüllbar**, wenn es (mindestens) eine zu φ passende σ-Interpretation $\mathcal{I}$ gibt, die φ erfüllt. — erfüllbar

(b) φ heißt **unerfüllbar**, wenn φ nicht erfüllbar ist. — unerfüllbar

(c) φ heißt **allgemeingültig**, wenn jede zu φ passende σ-Interpretation φ erfüllt. — allgemeingültig

Beispiel 5.29. Sei $\sigma_{\text{Graph}} := \{\dot{E}\}$ die Signatur, die aus einem 2-stelligen Relationssymbol $\dot{E}$ besteht. Die FO[σ_{Graph}]-Formel $\varphi := \forall y\, \dot{E}(x, y)$ ist erfüllbar, aber nicht allgemeingültig, denn: Sei $\mathfrak{A} := (A, \dot{E}^{\mathfrak{A}})$ der gerichtete Graph

und sei β die Belegung mit $\beta(x) = 1$. Dann erfüllt die Interpretation $(\mathfrak{A}, \beta)$ die Formel φ. Somit ist φ erfüllbar.

Andererseits gilt für den Graphen $\mathfrak{B} := (B, \dot{E}^{\mathfrak{B}})$

und die Belegung β mit $\beta(x) = 1$, dass die zu φ passende σ-Interpretation $\mathcal{I} := (\mathfrak{B}, \beta)$ die Formel φ **nicht** erfüllt (d.h. $[\![\varphi]\!]^{\mathcal{I}} = 0$), denn:

$$
\begin{aligned}
[\![\varphi]\!]^{\mathcal{I}} = 1 &\iff \text{Für jedes } b \in B \text{ gilt: } [\![\dot{E}(x,y)]\!]^{\mathcal{I}\frac{b}{y}} = 1 \\
&\iff \text{Für jedes } b \in B \text{ gilt: } \big(\beta\tfrac{b}{y}(x), \beta\tfrac{b}{y}(y)\big) \in \dot{E}^{\mathfrak{B}} \\
&\iff \text{Für jedes } b \in B \text{ gilt: } (1, b) \in \dot{E}^{\mathfrak{B}}.
\end{aligned}
$$

Aber für $b := 1$ gilt: $(1,1) \notin \dot{E}^{\mathfrak{B}}$, und daher ist $[\![\varphi]\!]^{\mathcal{I}} = 0$. $\mathcal{I}$ ist also eine zu φ passende σ-Interpretation, die φ **nicht** erfüllt. Somit ist φ nicht allgemeingültig.

Beobachtung 5.30. Für alle Formeln φ der Logik erster Stufe gilt:

(a) φ ist allgemeingültig $\iff$ $\neg\varphi$ ist unerfüllbar.

(b) φ ist erfüllbar $\iff$ $\neg\varphi$ ist nicht allgemeingültig.

Beweis: Übung. □

Definition 5.31 (semantische Folgerung)**.** Sei σ eine Signatur und seien φ und ψ zwei FO[σ]-Formeln. Wir sagen ψ **folgt aus** φ (kurz: $\varphi \models \psi$, "φ impliziert ψ"), falls für jede zu φ und ψ passende Interpretation $\mathcal{I}$ gilt: — ψ folgt aus φ

$$
\text{Falls} \quad \underbrace{\mathcal{I} \models \varphi}_{\text{d.h. } [\![\varphi]\!]^{\mathcal{I}} = 1} \quad , \quad \text{so auch} \quad \underbrace{\mathcal{I} \models \psi.}_{\text{d.h. } [\![\psi]\!]^{\mathcal{I}} = 1}
$$

Definition 5.32 (logische Äquivalenz)**.** Sei σ eine Signatur. Zwei FO[σ]-Formeln φ und ψ heißen **äquivalent** (kurz: $\varphi \equiv \psi$), wenn für jede zu φ und ψ passende σ-Interpretation $\mathcal{I}$ gilt: — äquivalent

$$
\mathcal{I} \text{ erfüllt } \varphi \iff \mathcal{I} \text{ erfüllt } \psi.
$$

Beobachtung 5.33. Sei σ eine Signatur und seien φ und ψ zwei FO[σ]-Formeln. Es gilt:

(a) $\varphi \equiv \psi \iff \varphi \models \psi$ und $\psi \models \varphi$.

(b) $\varphi \equiv \psi \iff (\varphi \leftrightarrow \psi)$ ist allgemeingültig.

(c) $\varphi \models \psi \iff (\varphi \rightarrow \psi)$ ist allgemeingültig.

Beweis: Übung. □

5.5.2 Grenzen der Logik erster Stufe

In Beispiel 5.18 und 5.19 haben wir viele Beispiele für umgangssprachliche Aussagen kennengelernt, die man durch Formeln der Logik erster Stufe beschreiben kann (siehe auch die Aufgaben zu diesem Kapitel). Es gibt allerdings auch Aussagen, die **nicht** in der Logik erster Stufe formalisiert werden können:

Satz 5.34. *Sei $\sigma_{Graph} := \{\dot{E}\}$ die Signatur, die aus einem 2-stelligen Relationssymbol $\dot{E}$ besteht. Es gilt:*

(a) Es gibt keinen FO[σ_{Graph}]*-Satz φ, so dass für jeden gerichteten Graphen $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ gilt:*

$$\mathfrak{A} \text{ erfüllt } \varphi \iff \mathfrak{A} \text{ ist azyklisch (vgl. Definition 4.14).}$$

(b) Es gibt keine FO[σ_{Graph}]*-Formel ψ mit freien Variablen x und y, so dass für jeden gerichteten Graphen $\mathfrak{A} = (A, \dot{E}^{\mathfrak{A}})$ und jede zu ψ passende Belegung β in $\mathfrak{A}$ gilt:*

$$(\mathfrak{A}, \beta) \text{ erfüllt } \psi \iff \text{es gibt in } \mathfrak{A} \text{ einen Weg von Knoten } \beta(x) \text{ zu Knoten } \beta(y).$$

Einen Beweis dieses Satzes können Sie in der Vorlesung "Logik in der Informatik" kennenlernen.

5.6 Ein Anwendungsbereich der Logik erster Stufe: Datenbanken

Rationale Datenbanken bestehen aus **Tabellen**, die sich als Relationen auffassen lassen. Datenbanken lassen sich daher als **Strukturen** über einer passenden Signatur auffassen. Die in der Praxis gebräuchlichste Datenbankanfragesprache ist **SQL**. Der "Kern" von SQL basiert auf der Logik erster Stufe, die in der Datenbankterminologie oft auch "relationaler Kalkül" (engl.: "relational calculus") bezeichnet wird.

Zur Illustration von Anfragen verwenden wir eine kleine Datenbank mit Kinodaten, bestehend aus:

- einer Tabelle *Orte*, die Informationen über Kinos (Kino, Adresse, Telefonnummer) enthält,
- einer Tabelle *Filme*, die Informationen über Filme enthält (Titel, Regie, Schauspieler).
- eine Tabelle *Programm*, die Informationen zum aktuellen Kinoprogramm enthält (Kino, Titel, Zeit).

Orte-Tabelle:

Kino	Adresse	Telefon
Babylon	Dresdner Str. 2	61609693
Casablanca	Friedenstr. 12	6775752
Cinestar Cubix Alexanderplatz	Rathausstr. 1	2576110
Die Kurbel	Giesebrechtstr. 4	88915998
Filmpalast Berlin	Kurfürstendamm 225	8838551
International	Karl-Marx-Allee 33	24756011
Kino in der Kulturbrauerei	Schönhauser Allee 36	44354422
Moviemento	Kottbusser Damm 22	6924785

Filme-Tabelle:

Titel	Regie	Schauspieler
Capote	Bennet Miller	Philip Seymour Hoffman
Capote	Bennet Miller	Catherine Keener
Das Leben der Anderen	F. Henkel von Donnersmarck	Martina Gedeck
Das Leben der Anderen	F. Henkel von Donnersmarck	Ulrich Tukur
Der ewige Gärtner	Fernando Meirelles	Ralph Fiennes
Der ewige Gärtner	Fernando Meirelles	Rachel Weisz
Good Night and Good Luck	George Clooney	David Strathairn
Good Night and Good Luck	George Clooney	Patricia Clarkson
Knallhart	Detlev Buck	Jenny Elvers
Knallhart	Detlev Buck	Jan Henrik Stahlberg
Raumpatrouille Orion – Rücksturz ins Kino	Michael Braun	Dietmar Schönherr
Raumpatrouille Orion – Rücksturz ins Kino	Theo Mezger	Dietmar Schönherr
Raumpatrouille Orion – Rücksturz ins Kino	Michael Braun	Eva Pflug
Raumpatrouille Orion – Rücksturz ins Kino	Theo Mezger	Eva Pflug
Raumpatrouille Orion – Rücksturz ins Kino	Michael Braun	Wolfgang Völz
Raumpatrouille Orion – Rücksturz ins Kino	Theo Mezger	Wolfgang Völz
Requiem	Hans-Christian Schmid	Sandra Hüller
Sommer vorm Balkon	Andreas Dresen	Nadja Uhl
Sommer vorm Balkon	Andreas Dresen	Inka Friedrich
Sommer vorm Balkon	Andreas Dresen	Andreas Schmidt
Syriana	Stephen Gaghan	George Clooney
Syriana	Stephen Gaghan	Matt Damon
V wie Vendetta	James McTeigue	Natalie Portman
Walk the Line	James Mangold	Joaquin Phoenix
Walk the Line	James Mangold	Reese Witherspoon

Programm-Tabelle:

Kino	Titel	Zeit
Babylon	Capote	17:00
Babylon	Capote	19:30
Kino in der Kulturbrauerei	Capote	17:30
Kino in der Kulturbrauerei	Capote	20:15
International	Das Leben der Anderen	14:30
International	Das Leben der Anderen	17:30
International	Das Leben der Anderen	20:30
Filmpalast Berlin	Good Night and Good Luck	15:30
Filmpalast Berlin	Good Night and Good Luck	17:45
Filmpalast Berlin	Good Night and Good Luck	20:00
Kino in der Kulturbrauerei	Good Night and Good Luck	18:00
Kino in der Kulturbrauerei	Good Night and Good Luck	20:00
Kino in der Kulturbrauerei	Good Night and Good Luck	22:45
Babylon	Sommer vorm Balkon	21:45
Kino in der Kulturbrauerei	Sommer vorm Balkon	21:45
Filmmuseum Potsdam	Raumpatrouille Orion – Rücksturz ins Kino	22:00

Für eine geeignete Signatur σ_{Kino} können wir diese Datenbank durch eine σ_{Kino}-Struktur $\mathfrak{A}_{\text{Kino}}$ folgendermaßen modellieren: Die Signatur σ_{Kino} besteht aus:

- einem 3-stelligen Relationssymbol $\dot{Orte}$
- einem 3-stelligen Relationssymbol $\dot{Filme}$
- einem 3-stelligen Relationssymbol $\dot{Programm}$
- Konstantensymbolen '$\dot{Babylon}$', '$\dot{Casablanca}$', ..., '$\dot{Capote}$', '*Das Leben* $\dot{der}$ *Anderen*', ... usw. – d.h. für jeden Eintrag c in der Beispieldatenbank gibt es ein Konstantensymbol '$\dot{c}$'.

Die σ_{Kino}-Struktur $\mathfrak{A}_{\text{Kino}}$ hat das Universum

$$\begin{aligned} A_{\text{Kino}} := \{&\text{Babylon}, \text{Casablanca}, \text{Cinestar Cubix am Alexanderplatz}, \dots, \\ &\text{Dresdner Str. } 2, \text{Friedenstr. } 12, \dots, 61609693, \dots, \text{Capote}, \dots, 22{:}00\}, \end{aligned}$$

die 3-stelligen Relationen

$$\begin{aligned} \dot{Orte}^{\mathfrak{A}_{\text{Kino}}} := \{&(\text{Babylon}, \text{Dresdner Str. } 2, 61609693), \\ &(\text{Casablanca}, \text{Friedenstr. } 12, 6775752), \\ &\dots, \\ &(\text{Moviemento}, \text{Kottbusser Damm } 22, 6924785)\}, \\ \dot{Filme}^{\mathfrak{A}_{\text{Kino}}} := \{&(\text{Capote}, \text{Bennet Miller}, \text{Philip Seymour Hoffman}), \\ &(\text{Capote}, \text{Bennet Miller}, \text{Catherine Keener}), \\ &\dots, \\ &(\text{Walk the Line}, \text{James Mangold}, \text{Reese Witherspoon})\}, \\ \dot{Programm}^{\mathfrak{A}_{\text{Kino}}} := \{&(\text{Babylon}, \text{Capote}, 17{:}00), \end{aligned}$$

$$
\begin{aligned}
&(\text{Babylon}, \text{Capote}, 19{:}30),\\
&(\text{Kino in der Kulturbrauerei}, \text{Capote}, 17{:}30),\\
&\dots\}
\end{aligned}
$$

sowie für jedes in σ_{Kino} vorkommende Konstantensymbol '$\dot{c}$' die Konstante '$\dot{c}$'$^{\mathfrak{A}_{\text{Kino}}} := c$ (d.h.:

$$
\begin{aligned}
&\text{'}\dot{Babylon}\text{'}^{\mathfrak{A}_{\text{Kino}}} = \text{Babylon},\\
&\text{'}\dot{Capote}\text{'}^{\mathfrak{A}_{\text{Kino}}} = \text{Capote},\\
&\text{'}\dot{George}\ \dot{Clooney}\text{'}^{\mathfrak{A}_{\text{Kino}}} = \text{George Clooney}
\end{aligned}
$$

usw. Anfragen an die Kinodatenbank lassen sich auf unterschiedliche Art formulieren:

Beispiel 5.35. Eine Anfrage an unsere Kinodatenbank:

> "Gib die Titel aller Filme aus, die um 20:30 Uhr laufen."

In der Datenbankanfragesprache SQL lässt sich dies folgendermaßen formulieren:

```
SELECT  Titel
FROM    Programm
WHERE   Zeit = '20:30'
```

Dieselbe Anfrage lässt sich auch durch die folgende Formel der Logik erster Stufe beschreiben:

$$\varphi_{\text{Filme um 20:30 Uhr}}(x_{\text{T}}) := \exists x_{\text{K}}\, \dot{Programm}(x_{\text{K}}, x_{\text{T}}, \text{'}\dot{20{:}30}\text{'}).$$

Notation 5.36. Sei σ eine Signatur und seien $x_1, \dots, x_n$ Variablen.

- Die Notation $\varphi(x_1, \dots, x_n)$ deutet an, dass φ eine FO[σ]-Formel mit frei(φ) $= \{x_1, \dots, x_n\}$ ist, d.h. dass $x_1, \dots, x_n$ diejenigen Variablen sind, die in φ frei vorkommen.
- Ist $\varphi(x_1, \dots, x_n)$ eine FO[σ]-Formel, ist $\mathfrak{A}$ eine σ-Struktur und sind $a_1, \dots, a_n \in A$ Elemente im Universum von $\mathfrak{A}$, so schreiben wir

  $$\mathfrak{A} \models \varphi[a_1, \dots, a_n],$$

  um auszudrücken, dass für die Belegung $\beta\colon \{x_1, \dots, x_n\} \to A$ mit $\beta(x_1) = a_1, \dots, \beta(x_n) = a_n$ gilt:

  $$(\mathfrak{A}, \beta) \models \varphi.$$

Definition 5.37. Sei σ eine Signatur, $\varphi(x_1, \dots, x_n)$ eine FO[σ]-Formel und $\mathfrak{A}$ eine σ-Signatur. Die von φ in $\mathfrak{A}$ definierte n-stellige Relation ist

$$\varphi(\mathfrak{A}) := \{(a_1, \dots, a_n) \in A^n : \mathfrak{A} \models \varphi[a_1, \dots, a_n]\}.$$

Beispiel 5.38. Die FO[σ_{Kino}]-Formel $\varphi_{\text{Filme um 20:30}}(x_{\text{T}})$ aus Beispiel 5.35 definiert in unserer Kinodatenbank $\mathfrak{A}_{\text{Kino}}$ die 1-stellige Relation:

$$
\begin{aligned}
\varphi_{\text{Filme um 20:30}}(\mathfrak{A}_{\text{Kino}}) = \{&(\text{Das Leben der Anderen}),\\
&(\text{Good Night and Good Luck})\}.
\end{aligned}
$$

Darstellung als Tabelle:

Filme um 20:30 Uhr:

Titel
Das Leben der Anderen
Good Night and Good Luck

Beispiel 5.39. Die Anfrage

> "Gib Name und Adresse aller Kinos aus, in denen ein Film läuft, in dem George Clooney mitspielt oder Regie geführt hat."

lässt sich folgendermaßen formulieren:

In SQL:

```
SELECT  Orte.Kino, Orte.Adresse
FROM    Orte, Filme, Programm
WHERE   Orte.Kino = Programm.Kino AND
        Filme.Titel = Programm.Titel AND
        (Filme.Schauspieler = 'George Clooney' OR
         Filme.Regie = 'George Clooney')
```

In Logik erster Stufe:

$$\begin{aligned}\varphi_{\text{Kinos mit George Clooney}}(x_{\mathrm{K}}, x_{\mathrm{A}}) \;:=\; & \\ & \exists x_{\mathrm{Tel}} \exists x_{\mathrm{T}} \exists x_{\mathrm{Z}} \Big(\big(\dot{Orte}(x_{\mathrm{K}}, x_{\mathrm{A}}, x_{\mathrm{Tel}}) \wedge \dot{Programm}(x_{\mathrm{K}}, x_{\mathrm{T}}, x_{\mathrm{Z}}) \big) \wedge \\ & \qquad \big(\exists x_{\mathrm{R}}\, \dot{Filme}(x_{\mathrm{T}}, x_{\mathrm{R}}, \text{'George } \dot{\text{Clooney'}}) \vee \\ & \qquad\quad \exists x_{\mathrm{S}}\, \dot{Filme}(x_{\mathrm{T}}, \text{'George } \dot{\text{Clooney'}}, x_{\mathrm{S}}) \big) \Big)\end{aligned}$$

In unserer konkreten Kinodatenbank $\mathfrak{A}_{\text{Kino}}$ liefert diese Formel die 2-stellige Relation:

$$\begin{aligned}\varphi_{\text{Kinos mit George Clooney}}(\mathfrak{A}_{\text{Kino}}) \;=\; & \{(\text{Filmpalast Berlin}, \text{Kurfürstendamm 225}), \\ & \;(\text{Kino in der Kulturbrauerei}, \text{Schönhauser Allee 36})\}.\end{aligned}$$

Darstellung als Tabelle:

Kinos mit George Clooney:

Kino	Adresse
Filmpalast Berlin	Kurfürstendamm 225
Kino in der Kulturbrauerei	Schönhauser Allee 36

Datails zum Thema Datenbanken und Datenbankanfragesprachen können Sie in den Vorlesungen "Datenbanksysteme I und II" und "Logik und Datenbanken" kennenlernen.

5.7 Übungsaufgaben zu Kapitel 5

Aufgabe 5.1. Sei $\sigma = \{\dot{B}, \dot{S}, \dot{F}, \dot{Nachfolger}, \dot{letzter}\}$ eine Signatur, wobei $\dot{B}, \dot{S}, \dot{F}$ 1-stellige Relationsymbole, $\dot{Nachfolger}$ ein 1-stelliges Funktionssymbol und $\dot{letzter}$ ein Konstantensymbol ist. Sei $\mathfrak{A}$ eine σ-Struktur mit $A = \{1, 2, \ldots, 34\}$ und $\dot{letzter}^{\mathfrak{A}} = 34$, so dass für alle $a \in A$ gilt:

- $a \in \dot{B}^{\mathfrak{A}} \iff$ FC Bayern München ist Tabellenführer an Spieltag a
- $a \in \dot{S}^{\mathfrak{A}} \iff$ FC Schalke 04 ist Tabellenführer an Spieltag a
- $a \in \dot{F}^{\mathfrak{A}} \iff$ Eintracht Frankfurt ist Tabellenführer an Spieltag a
- $\dot{Nachfolger}^{\mathfrak{A}}(a) = \begin{cases} a+1, & \text{falls } a \in \{1, 2, \ldots, 33\} \\ a, & \text{falls } a = 34. \end{cases}$

(a) Geben Sie FO[σ]-Formeln an, die in $\mathfrak{A}$ folgendes aussagen:

(i) Eintracht Frankfurt ist mindestens einmal Tabellenführer.

(ii) Jede der drei Mannschaften ist mindestens einmal Tabellenführer.

(iii) Sind die Bayern an einem Spieltag Erster, so werden sie auch Meister.

(iv) Schalke holt nicht den Titel, wenn sie bereits am vorletzten Spieltag Tabellenführer sind.

(b) Beschreiben Sie umgangssprachlich, was jede der folgenden FO[σ]-Formeln in $\mathfrak{A}$ aussagt:

(i) $\forall x\big(\ \neg \dot{B}(x)\ \rightarrow\ (\dot{S}(x) \vee \dot{F}(x))\ \big)$

(ii) $\neg\exists x\Big(\ \dot{F}(x)\ \wedge\ \big(\ \dot{F}(\dot{Nachfolger}(x))\ \wedge\ \big(\ \dot{F}(\dot{Nachfolger}(\dot{Nachfolger}(x))) \wedge \neg \dot{Nachfolger}(x) \doteq \dot{letzter}\ \big)\ \big)\ \Big)$

(iii) $\Big(\ \neg\exists x\big(\dot{S}(x) \wedge \neg x \doteq \dot{letzter}\big)\ \rightarrow\ \neg\dot{S}(\dot{letzter})\ \Big)$

Aufgabe 5.2. Sei $\sigma = \{\dot{f}, \dot{R}, \dot{c}\}$ eine Signatur mit einem 2-stelligen Funktionssymbol $\dot{f}$, einem 3-stelligen Relationsymbol $\dot{R}$ und einem Konstantensymbol $\dot{c}$. Betrachten Sie die σ-Struktur $\mathfrak{A} = (A, \dot{f}^{\mathfrak{A}}, \dot{R}^{\mathfrak{A}}, \dot{c}^{\mathfrak{A}})$, wobei $A = \{0, 1, 2, 3, 4\}$, $\dot{R}^{\mathfrak{A}} = \{(0, 3, 4), (1, 3, 0), (4, 2, 3)\}$, $\dot{c}^{\mathfrak{A}} = 3$ und die Funktion $\dot{f}^{\mathfrak{A}} \colon A \times A \to A$ definiert ist durch

$\dot{f}^{\mathfrak{A}}$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Zum Beispiel gilt $\dot{f}^{\mathfrak{A}}(2, 3) = 0$ und $\dot{f}^{\mathfrak{A}}(1, 3) = 4$.

Sei $\mathcal{I} = (\mathfrak{A}, \beta)$ die Interpretation mit der Belegung $\beta \colon \textsc{Var} \to A$, für die gilt: $\beta(v_0) = 2$, $\beta(v_1) = 0$, $\beta(v_2) = 1$, $\beta(v_3) = 4$, und $\beta(v_i) = 3$ für alle $i \geq 4$.

(a) Berechnen Sie $[\![t_1]\!]^{\mathcal{I}}$ und $[\![t_2]\!]^{\mathcal{I}}$ für

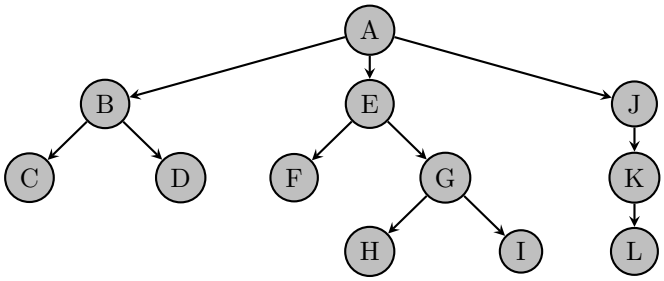
- $t_1 \ := \ \dot{f}(\dot{f}(v_1, v_5), \dot{c})$
- $t_2 \ := \ \dot{f}\big(\dot{f}(\dot{c}, \dot{f}(v_2, \dot{c})), v_1\big)$

(b) Berechnen Sie $[\![\varphi_1]\!]^{\mathcal{I}}$ und $[\![\varphi_2]\!]^{\mathcal{I}}$ für

- $\varphi_1 \ := \ \big(\ \dot{R}(v_1, v_2, \dot{f}(v_0, v_2)) \vee \exists v_0\, \dot{R}(v_0, v_2, v_3)\ \big)$
- $\varphi_2 \ := \ \forall v_1\Big(\ \dot{f}(v_1, \dot{c}) \doteq \dot{f}(v_2, \dot{c})\ \rightarrow\ \exists v_3\big(\dot{R}(v_1, v_2, v_3) \vee \dot{f}(v_1, v_5) \doteq v_4\big)\ \Big)$

Aufgabe 5.3. In dieser Aufgabe sollen gerichtete Bäume durch Strukturen über einer Signatur mit einem 1-stelligen Funktionssymbol *Elternknoten* repräsentiert werden.

(a) Beschreiben Sie, wie ein gegebener gerichteter Baum $B = (V, E)$ durch eine Struktur über der Signatur $\{\dot{Elternknoten}\}$ modelliert werden kann. Geben Sie die entsprechende Struktur für den folgenden Baum an:



(b) Geben Sie je eine Formel $\varphi(x)$ der Logik erster Stufe an, die ausdrückt, dass der Knoten x

(i) ein Blatt ist,
(ii) die Wurzel ist,
(iii) genau zwei Kinder hat.

Aufgabe 5.4. Sei $\sigma := \{\dot{E}, \dot{P}\}$ eine Signatur mit einem 2-stelligen Relationssymbol $\dot{E}$ und einem 1-stelligen Relationssymbol $\dot{P}$. Geben Sie für jede der folgenden Formeln je eine σ-Struktur an, die die Formel erfüllt, und eine, die die Formel nicht erfüllt:

(a) $\forall x \forall y \forall z \big(\big(\dot{E}(x,y) \wedge \dot{E}(y,z)\big) \rightarrow \dot{E}(x,z) \big)$

(b) $\forall x \forall y \Big(\dot{E}(x,y) \rightarrow \big(\big(\dot{P}(y) \wedge \neg\dot{P}(x)\big) \vee \big(\dot{P}(x) \wedge \neg\dot{P}(y)\big) \big) \Big)$

(c) $\Big(\forall x \forall y \big(\dot{E}(x,y) \vee \dot{E}(y,x)\big) \wedge \forall x \forall y \big((\dot{E}(x,y) \wedge \dot{E}(y,x)) \rightarrow x \doteq y \big) \Big)$

Aufgabe 5.5 (freie und gebundene Variablen)**.** Bestimmen Sie für jede der folgenden $\{\dot{P}, \dot{E}, \dot{R}, \dot{f}, \dot{g}, \dot{c}\}$-Formeln, welche Variablen gebunden und welche Variablen frei in der Formel vorkommen:

(a) $\Big(\dot{P}(x) \vee \neg\big(\dot{E}(y,z) \rightarrow \dot{f}(x) \doteq \dot{f}(y)\big) \Big)$

(b) $\forall x \Big(\dot{f}(x) \doteq \dot{g}(y,x) \vee \exists z\, \dot{E}\big(x, \dot{g}(x,z)\big) \Big)$

(c) $\Big(\forall y \neg\dot{E}(y,x) \wedge \exists z \big(\dot{E}(x,z) \wedge \dot{E}(z,y)\big) \Big)$

(d) $\exists x \forall y \exists z \Big(\dot{f}(y) \doteq \dot{g}(x,z) \vee \neg\dot{R}\big(\dot{c}, z, \dot{f}(y)\big) \Big)$

Aufgabe 5.6 (Äquivalenz, Folgerung)**.**

(a) Welche der folgenden Aussagen stimmen, welche stimmen nicht?

(i) $\forall x\, \varphi \equiv \neg\exists x\, \neg\varphi$
(ii) $\exists x(\varphi \wedge \psi) \models \big(\exists x\, \varphi \wedge \exists x\, \psi\big)$
(iii) $\big(\exists x\, \varphi \wedge \exists x\, \psi\big) \models \exists x(\varphi \wedge \psi)$
(iv) $\exists x(\varphi \wedge \psi) \equiv \big(\exists x\, \varphi \wedge \exists x\, \psi\big)$
(v) $\forall x(\varphi \wedge \psi) \equiv \big(\forall x\, \varphi \wedge \forall x\, \psi\big)$

(vi) $\forall x\,\varphi \;\models\; \exists x\,\varphi$

(b) Beweisen Sie, dass Ihre Antworten zu (ii), (iii) und (vi) aus (a) korrekt sind.

Aufgabe 5.7 (Datenbankanfragen)**.** Betrachten Sie die Kinodatenbank $\mathfrak{A}_{\text{Kino}}$ aus der Vorlesung.

(a) Berechnen Sie für jede der folgenden Formeln φ_i die Relation $\varphi_i(\mathfrak{A}_{\text{Kino}})$ und geben Sie umgangssprachlich an, welche Anfrage durch die Formel φ_i beschrieben wird:

$$
\begin{aligned}
\varphi_1(x_{\text{K}}) &= \exists x_{\text{Z}}\, \dot{Programm}(x_{\text{K}}, \text{`}\dot{Capote}\text{'}, x_{\text{Z}}) \\
\varphi_2(x_{\text{S}}) &= \exists x_{\text{T}}\big(\, \exists x_{\text{R}}\, \dot{Filme}(x_{\text{T}}, x_{\text{R}}, x_{\text{S}}) \;\wedge\; \exists x_{\text{K}} \exists x_{\text{Z}}\, \dot{Programm}(x_{\text{K}}, x_{\text{T}}, x_{\text{Z}}) \,\big) \\
\varphi_3(x_{\text{T}}) &= \exists x_{\text{K}} \exists x_{\text{Z}} \Big(\dot{Programm}(x_{\text{K}}, x_{\text{T}}, x_{\text{Z}}) \;\wedge\; \\
&\qquad\qquad \forall y_{\text{K}} \forall y_{\text{Z}} \big(\dot{Programm}(y_{\text{K}}, x_{\text{T}}, y_{\text{Z}}) \rightarrow y_{\text{Z}} \doteq x_{\text{Z}} \big) \Big) \\
\varphi_4(x_{\text{T}}, x_{\text{K}}, x_{\text{A}}) &= \Big(\exists x_{\text{Z}}\, \exists x_{\text{Tel}} \big(\dot{Programm}(x_{\text{K}}, x_{\text{T}}, x_{\text{Z}}) \wedge \dot{Orte}(x_{\text{K}}, x_{\text{A}}, x_{\text{Tel}}) \big) \;\wedge \\
&\qquad\qquad \exists x_{\text{S}}\, \dot{Filme}(x_{\text{T}}, \text{`}\dot{George}\;\dot{Clooney}\text{'}, x_{\text{S}}) \Big)
\end{aligned}
$$

(b) Finden Sie Formeln der Logik erster Stufe, die die folgenden Anfragen beschreiben:

(i) Gib die Titel aller Filme aus, die in mindestens zwei Kinos laufen.

(ii) Gib die Titel aller Filme aus, in denen George Clooney mitspielt, aber nicht selbst Regie führt.

Beachten Sie: Es kann sein, dass ein Film mehr als einen Regisseur hat, z.B. Raumpatrouille Orion – Rücksturz ins Kino.

(iii) Gib die Titel aller Filme aus, deren Schauspieler schon mal in einem Film von Stephen Spielberg mitgespielt haben.

6 Modellierung von Strukturen

In Kapitel 4 haben wir bereits **Graphen** und **Bäume** als Möglichkeiten kennengelernt, mit denen sich Objekte sowie Beziehungen zwischen je 2 Objekten gut modellieren lassen. In Kapitel 5.2 wurden Verallgemeinerungen davon eingeführt, die so genannten σ-Strukturen, wobei σ eine Signatur ist. Abgesehen von Graphen und Bäumen kann man damit beispielsweise auch die natürlichen (oder die rationalen) Zahlen mit arithmetischen Operationen $+$, $\times$ etc. modellieren oder – wie in Kapitel 5.6 gesehen – auch relationale Datenbanken, die z.B. Informationen über Kinofilme und das aktuelle Kinoprogramm enthalten. In Kapitel 6 werden wir nun zwei weitere Kalküle kennenlernen, mit denen man strukturelle Eigenschaften von Systemen beschreiben kann: das Entity-Relationship-Modell und kontextfreie Grammatiken.

6.1 Das Entity-Relationship-Modell

Das Entity-Relationship-Modell (kurz: **ER-Modell**) geht zurück auf einen grundlegenden Artikel von P.P. Chen aus dem Jahr 1976:

> P.P. Chen „The Entity-Relationship-Model – Towards a Unified View of Data." ACM Transactions on Database Systems, Band 1, Nr. 1, Seiten 9–36, 1976.

Es wird heute praktisch als Standardmodell für frühe Entwurfsphasen in der Datenbankentwicklung eingesetzt. Darüber hinaus basiert auch die Spezifikationssprache UML („Unified Modeling Language"), die zur Spezifikation von Strukturen und Beziehungen in Software-Systemen eingesetzt wird, auf dem ER-Modell.

In dieser Vorlesung werden nur einige grundlegende Züge des ER-Modells vorgestellt. Details können Sie z.B. in der Veranstaltung „Datenbanksysteme I" kennenlernen.

Das ER-Modell basiert auf den 3 Grundkonzepten

Entity
- **Entity**: „zu modellierende Informationseinheit" (deutsch: „Objekt", „Ding", „Entität")

Relationship
- **Relationship**: zur Modellierung von Beziehungen zwischen Entities (deutsch: „Beziehung", „Relation")

Attribut
- **Attribut**: Eigenschaft von einem Entity oder einer Beziehung.

Genauer:

Entity
- **Entity**: Objekt der realen oder der Vorstellungswelt, über das Informationen zu speichern sind (z.B. eine Vorlesungsveranstaltung, ein Buch oder eine/n Dozent/in). Auch Informationen über Ereignisse wie Klausuren können Objekte im Sinne des ER-Modells sein.

Entity-Typ
Entity-Menge
- **Entity-Typ** (bzw. **Entity-Menge**): eine Zusammenfassung von Entities, die im Modell als „gleichartig" angesehen werden (z.B. „Vorlesung", „Buch", „Dozent/in"). Im Modell steht ein Entity-Typ für die Menge aller in Frage kommenden Objekte dieser Art.

- **Relationship**: Beziehung zwischen Entities (z.B. welche Dozenten/innen welche Vorlesungen halten). Relationship
- **Attribut**: Eigenschaften von Entities oder Relationships (z.B. die ISBN eines Buchs, der Titel einer Vorlesung oder die Semester, in denen Vorlesung X von Dozent/in Y gehalten wird). Attribut

Beispiel 6.1. Abbildung 6.1 zeigt eine graphische Darstellung für eine Modellierung im ER-Modell – im Beispiel geht es darum, Vorlesungen, Dozenten und für die Vorlesungen empfohlene Bücher darzustellen.

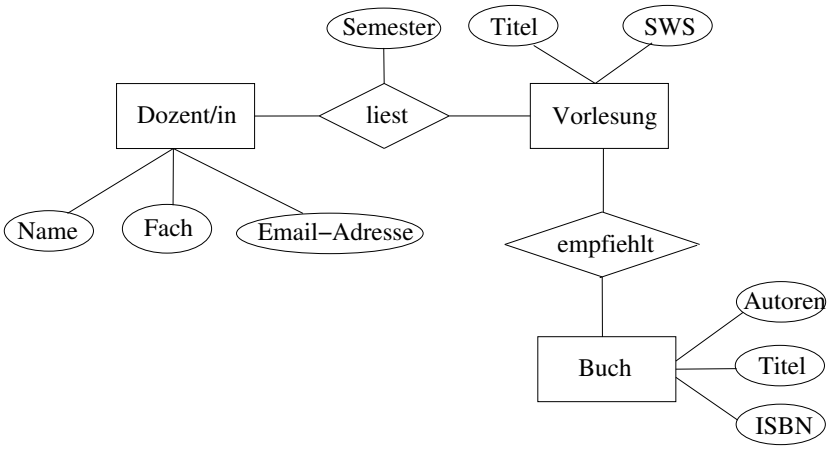
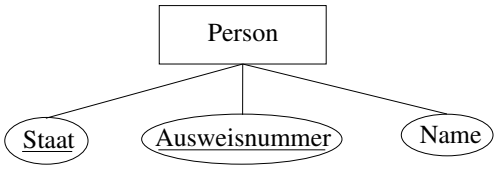


Abbildung 6.1: Ein ER-Modell, das Vorlesungen, Dozenten und für die Vorlesungen empfohlene Bücher darstellt

- **Entity-Typen** werden als Rechtecke dargestellt (hier: Dozent/in, Vorlesung, Buch).
- **Eigenschaften von Entities**, sog. **Attribute**, werden durch Ellipsen dargestellt, die mit dem Rechteck des zugehörigen Entity-Typs verbunden sind (im Beispiel hat jede/r Dozent/in die Attribute „Name“, „Fach“ und „Email-Adresse“).

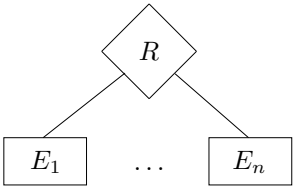
  Ein Attribut ordnet jeder Entity des entsprechenden Entity-Typs einen Wert zu. Ein Attribut, dessen Wert jedes Entity eindeutig bestimmt (z.B. die ISBN von Büchern), heißt **Schlüsselattribut**. Um Schlüsselattribute im ER-Modell explizit zu kennzeichnen, werden sie unterstrichen. Auch mehrere Attribute zusammen können einen Schlüssel bilden, z.B. Schlüsselattribut



- Typen von Relationships, sog. **Relationen-Typen**, werden durch Rauten dargestellt, die mit den betreffenden Entity-Typen durch Striche verbunden sind (z.B. ist in Beispiel 6.1 „liest“ ein Relationen-Typ, der angibt, welche/r Dozent/in welche Vorlesung liest). Relationen-Typen

n-stelliger Relationen-Typ

Allgemein gilt: Ein Relationen-Typ modelliert Beziehungen zwischen den Entities der betroffenen Entity-Typen. Ein ***n*-stelliger Relationen-Typ *R*** (für $n \geq 2$) verknüpft Entities aus n Entity-Typen $E_1, \ldots, E_n$. Er wird graphisch repräsentiert durch



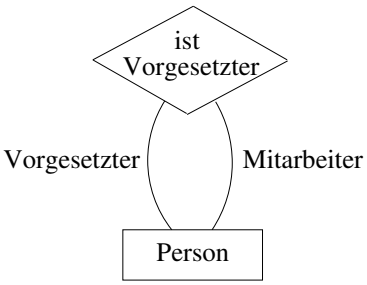
Eine **konkrete Ausprägung des Relationen-Typs** R ist eine Menge von n-Tupeln $(e_1, \ldots, e_n)$, wobei für jedes $i \in \{1, \ldots, n\}$ gilt: e_i ist ein Entity des Entity-Typs E_i.

- Auch Relationen-Typen können Attribute haben (z.B. hat der Relationen-Typ „liest“ in Beispiel 6.1 ein Attribut „Semester“, das angibt, in welchen Semestern Dozent/in X die Vorlesung Y hält).

  Allgemein gilt: Ein Attribut ordnet jedem Tupel des entsprechenden Relationen-Typs einen Wert zu. Beispielsweise ordnet das Attribut „Semester“ jedem Tupel (X, Y) der „liest“-Relationen die Liste aller Semester zu, in denen Dozent/in X die Vorlesung Y hält.

- Für manche Relationen-Typen wird aus ihrem Namen und der graphischen Darstellung zunächst nicht klar, welche Bedeutung die einzelnen Entity-Typen in der Relation haben – insbesondere, wenn ein Entity-Typ mehrfach am Relationen-Typ beteiligt ist. Es können dann **Rollennamen** vergeben werden, etwa um die Beziehung „Person X ist Vorgesetzter von Person Y“ darzustellen:

Rollennamen



Beispiel 6.2. Man beachte die Auswirkung von Modellierungsentscheidungen beim Entwickeln eines ER-Modells: Nutzt ein Reisebüro das ER-Modell aus Abbildung 6.2, so besteht eine konkrete Ausprägung des Relationen-Typs „gebucht für“ aus einer Menge von Tupeln (X, Y), die angibt, dass Person X ein Ticket für Flug Y gebucht hat – und zwar zum Preis Preis(X, Y). Insbesondere heißt dies aber, dass Passagier X für Flug Y nicht zwei verschiedene Buchungen getätigt haben kann. Wenn man solche „Mehrfachbuchungen“ zulassen will, kann man das ER-Modell aus Abbildung 6.3 benutzen.

Ein weiterer Bestandteil von ER-Modellen

Kardinalität

Kardinalität von Relationen-Typen:

Relationen-Typen in der Form, wie wir sie bisher eingeführt haben, sagen über konkrete Ausprägungen nur aus, dass einige Entities aus den beteiligten Entity-Typen in der angegebenen

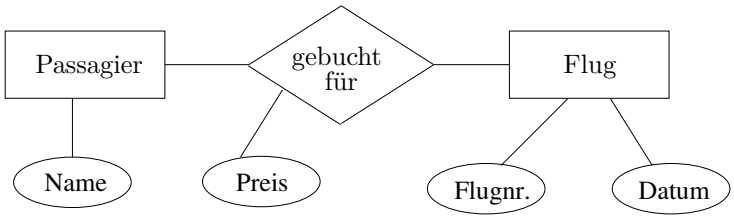


Abbildung 6.2: ER-Modell für ein Reisebüro

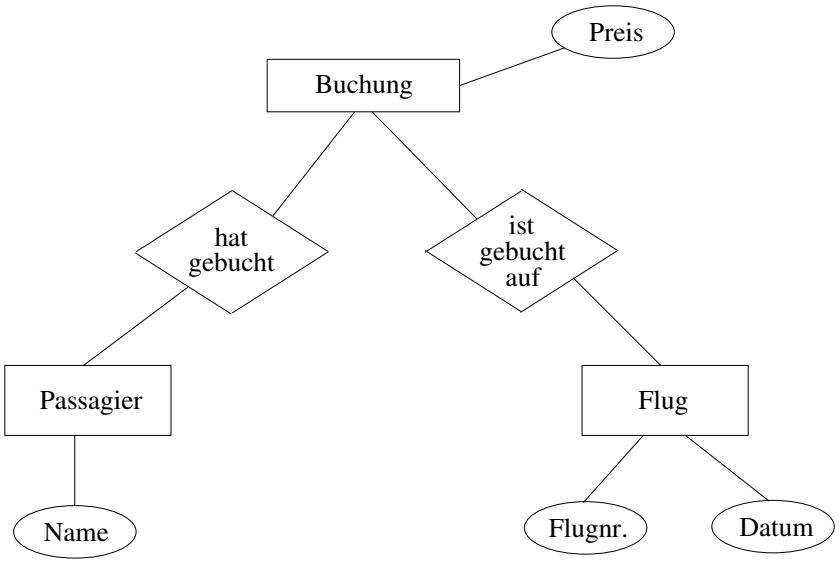
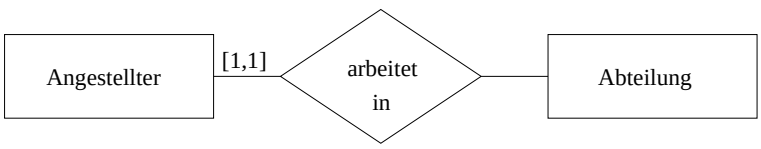
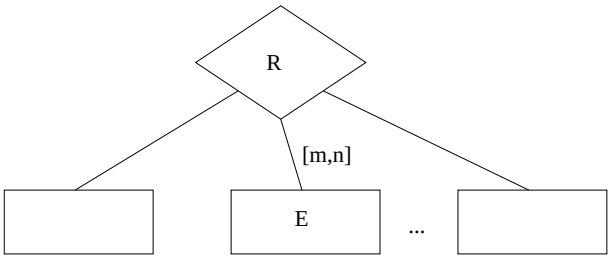


Abbildung 6.3: ER-Modell für ein Reisebüro mit „Mehrfachbuchungen“

Beziehung stehen können. Oft will man aber genauere Angaben (bzw. Einschränkungen) machen – z.B. dass jeder Angestellte durch eine Relation des Relationen-Typs „arbeitet in“ mit genau einer Abteilung verbunden ist. Dies kann graphisch folgendermaßen dargestellt werden:



Allgemein besagt ein Relationen-Typ der Form

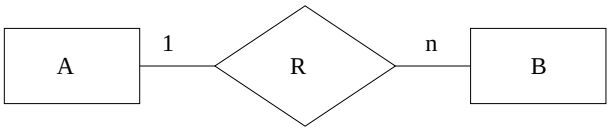


dass für jede konkrete Ausprägung dieses Typs gelten muss: Jedes Entity e der konkreten Ausprägung des Entity-Typs E kommt in mindestens m und höchstens n Tupeln vor.

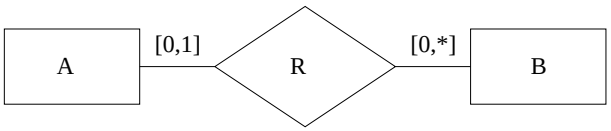
Spezialfälle für $[m, n]$:

- $[1, 1]$ bedeutet: „in genau einem Tupel“.
- $[0, 1]$ bedeutet: „in höchstens einem Tupel“.
- $[0, *]$ bedeutet: „in beliebig vielen Tupeln“.
  Die Angabe $[0, *]$ wird oft auch einfach weggelassen.

Kurznotation für 2-stellige Relationen-Typen:



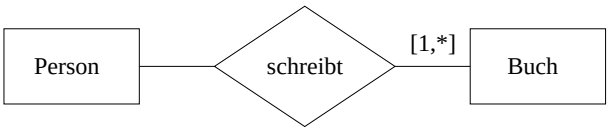
bedeutet



d.h.: „jedes Entity a des Typs A kommt in höchstens einem Tupel von R vor, und jedes Entity b des Typs B kommt in beliebig vielen Tupeln von R vor.“

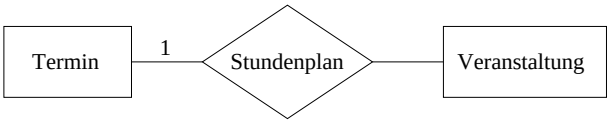
Beispiel 6.3.

(a)



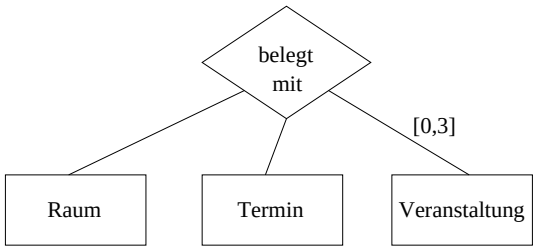
bedeutet: „Jedes Buch wird von mindestens einer Person geschrieben.“

(b)



bedeutet: „Jeder Termin im Stundenplan ist mit höchstens einer Veranstaltung belegt.“

(c)



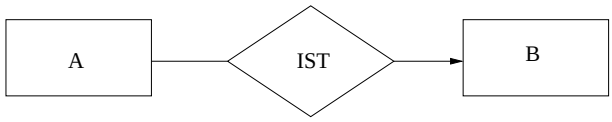
bedeutet: „Jede Veranstaltung wird höchstens dreimal (pro Woche) angeboten.“

Noch ein Bestandteil von ER-Modellen:

Die **IST-Beziehung** (englisch „is-a“): IST-Beziehung

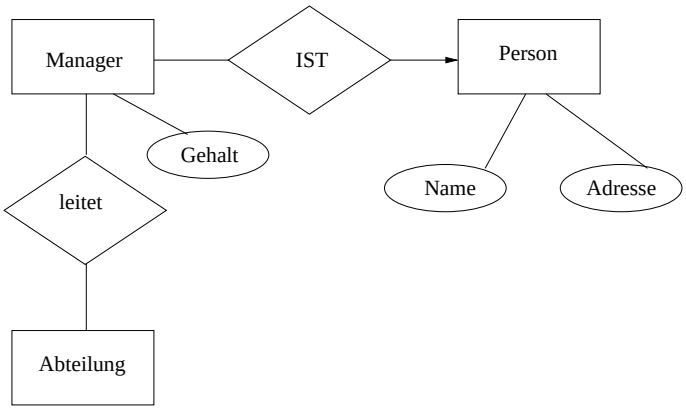
Der spezielle Relationen-Typ IST definiert eine Spezialisierungs-Hierarchie.

Graphische Darstellung:



Bedeutung: Jedes Entity des Typs A ist auch ein Entity des Typs B (d.h. A ist eine Spezialisierung des Typs B).

Beispiel:



Allgemein gilt: Die Entities des Typs A „erben“ alle Attribute von B und können außerdem noch weitere Attribute haben, die spezielle „A-Eigenschaften“ beschreiben. Auch Schlüsselattribute werden als solche geerbt.

Beispiel 6.4. Ein umfangreiches ER-Modell, das einige Aspekte einer Fluggesellschaft modelliert, ist in Abbildung 6.4 dargestellt. In diesem ER-Modell wird u.a. folgendes modelliert:

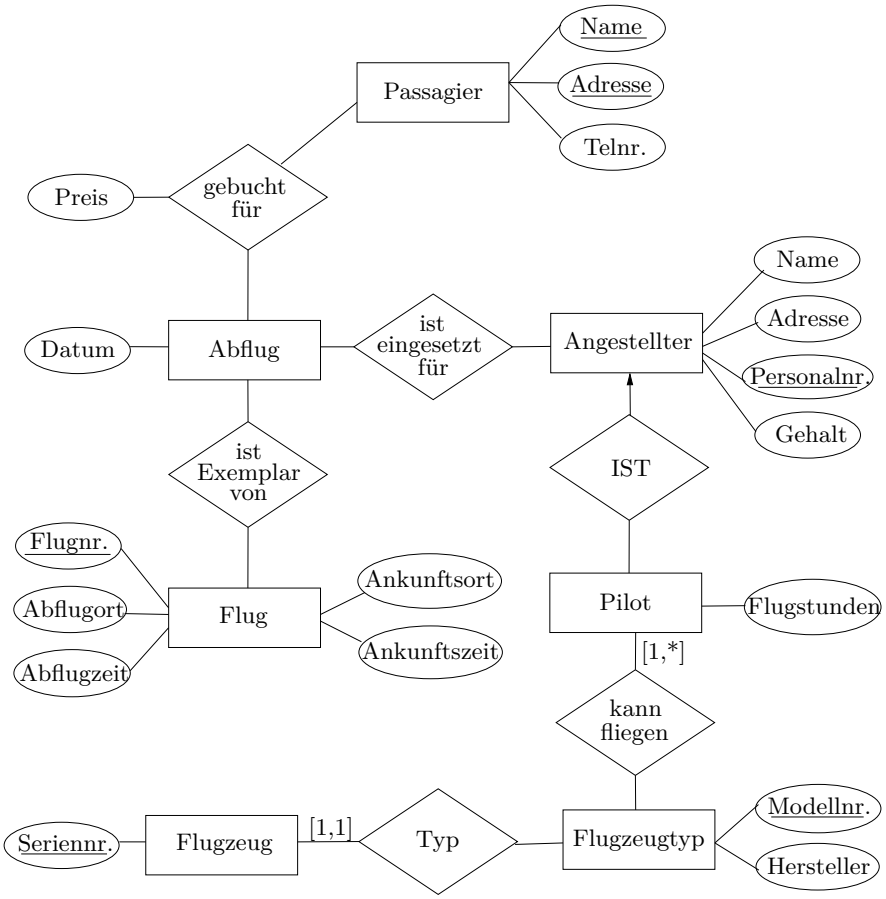


Abbildung 6.4: ER-Modell, das einige Aspekte einer Fluggesellschaft modelliert

(1) Es kommt eine IST-Spezialisierung vor, die besagt, dass Piloten spezielle Angestellte sind. Das wird insbesondere benötigt, um den Relationen-Typ „kann fliegen“ hinreichend präzise formulieren zu können und das Attribut „Flugstunden“ nicht allen Angestellten zuordnen zu müssen.

(2) Entities aller hier aufgeführten Typen (bis auf Abflug) werden durch Schlüsselattribute eindeutig identifiziert. Bei den Passagieren wird angenommen, dass Name und Adresse den jeweiligen Passagier eindeutig festlegen. Die Namen der übrigen Schlüsselattribute deuten an, dass man jeweils eine Nummerierung eingeführt hat, um die Eindeutigkeit zu erreichen (z.B. Personalnr., Flugnr., etc.).

(3) Der Relationen-Typ „Typ“ verbindet konkrete Flugzeuge mit Flugzeugtypen, in dem sie jedem Flugzeug genau einen Flugzeugtypen zuordnet. Solche Unterscheidungen zwischen „Typ“ und „Exemplar“ werden oft in Modellierungen verwendet und sind wichtig, damit Relationen und Attribute sachgerecht zugeordnet werden können. Beispielsweise sind die Fähigkeiten eines Piloten dadurch bestimmt, welche Flugzeugtypen er fliegen kann – und nicht durch die konkreten Flugzeuge (Exemplare), die er fliegen kann.

Vorsicht: Beim ersten Hinsehen mag es verlockend erscheinen, Typ-Exemplar-Beziehungen durch die IST-Spezialisierung zu modellieren. Dies ist aber ein schwerer Entwurfsfehler: „Typen“ und „Exemplare“ bezeichnen verschiedenartige Entity-Typen, zwischen denen i.d.R. keine Teilmengen-Beziehung (wie bei der IST-Spezialisierung) bestehen kann.

6.2 Kontextfreie Grammatik

Kontextfreie Grammatiken (kurz: KFGs) eignen sich besonders gut zur Modellierung von beliebig tief geschachtelten baumartigen Strukturen. KFGs können gleichzeitig

- hierarchische Baumstrukturen und
- Sprachen und deren textuelle Notation

spezifizieren. KFGs werden z.B. angewendet zur Definition von:

- Programmen einer Programmiersprache und deren Struktur, z.B. Java, C, Pascal (KFGs spielen z.B. beim „Compilerbau“ eine wichtige Rolle).
- Datenaustauschformaten, d.h. Sprachen als Schnittstelle zwischen Software-Werkzeugen, z.B. HTML, XML.
- Bäumen zur Repräsentation strukturierter Daten, z.B. XML.
- Strukturen von Protokollen beim Austausch von Nachrichten zwischen Prozessen oder Geräten.

KFGs sind ein grundlegender Kalkül, der für die formale Definition von Sprachen eingesetzt wird.

In dieser Veranstaltung werden nur die Grundbegriffe und einige Beispiele vorgestellt: im Detail werden KFGs in der Veranstaltung „GL-2“ behandelt.

6.2.1 Definition des Begriffs „Kontextfreie Grammatik“

Es gibt 2 Sichtweisen auf KFGs:

(1) Eine KFG ist ein spezielles **Ersetzungssystem**. Seine Regeln geben an, auf welche Art man ein Symbol durch eine Folge von Symbolen ersetzen kann. Auf diese Weise definiert eine KFG eine **Sprache**, d.h. eine **Menge von Worten** über einem bestimmten Alphabet, die mit dem durch die KFG gegebenen Regeln erzeugt werden können.

(2) Gleichzeitig definiert eine KFG eine **Menge von Baumstrukturen**, die sich durch schrittweises Anwenden der Regeln erzeugen lassen.

Für die Modellierung von Strukturen ist die zweite Sichtweise besonderes interessant. Aber es ist oft sehr nützlich, dass derselbe Kalkül auch gleichzeitig eine textuelle Notation für die Baumstrukturen liefern kann und dass Eigenschaften der zugehörigen Sprache untersucht werden können.

Definition 6.5 (KFG)**.** Eine kontextfreie Grammatik $G = (T, N, S, P)$ besteht aus:

- einer endlichen Menge T, der so genannten Menge der **Terminalsymbole** (die Elemente aus T werden auch **Terminale** genannt). Terminalsymbole, Terminale

Nichtterminalsymbole
Nichtterminale

- einer endlichen Menge N, der so genannten Menge der **Nichtterminalsymbole** (die Elemente aus N werden auch **Nichtterminale** genannt).

  Die Mengen T und N sind disjunkt, d.h. $T \cap N = \varnothing$. Die Menge $V := T \cup N$ heißt **Vokabular**; die Elemente in V nennt man auch **Symbole**.

Vokabular
Symbole
Startsymbol

- einem Symbol $S \in N$, dem so genannten **Startsymbol**.

Produktionen

- einer endlichen Menge $P \subseteq N \times V^*$, der so genannten Menge der **Produktionen**. Für eine Produktion $(A, x) \in P$ schreiben wir meistens $A \to x$ (bzw. $A ::= x$).

Beispiel 6.6. Als erstes Beispiel betrachten wir eine KFG für „Wohlgeformte Klammerausdrücke“: $G_{\mathrm{K}} := (T, N, S, P)$ mit

- $T := \{\,(\,,\,)\,\}$

  D.h. G_{K} hat als Terminale die Symbole „(“ („öffnende Klammer“) und „)“ („schließende Klammer“).

- $N := \{\,\mathrm{Klammerung}\,,\,\mathrm{Liste}\,\}$

  D.h. G_{K} hat als Nichtterminale die Symbole „Klammerung“ und „Liste“.

- $S := \mathrm{Klammerung}$

  D.h. „Klammerung“ ist das Startsymbol.

- $P := \{\,\mathrm{Klammerung} \to (\mathrm{Liste}),$
  $\mathrm{Liste} \to \mathrm{Klammerung\ Liste},$
  $\mathrm{Liste} \to \varepsilon\,\}$

  (zur Erinnerung: ε bezeichnet das leere Wort).

6.2.2 Bedeutung der Produktionen/Semantik von KFGs

Jede Produktion einer KFG, etwa die Produktion $A \to x$, kann man auffassen als:

- „Strukturregel“, die besagt „Ein A besteht aus x“ oder als
- „Ersetzungsregel“, die besagt „A kann man durch x ersetzen.“

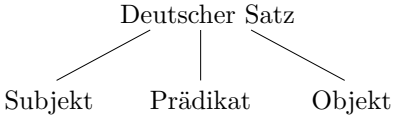
Beispielsweise kann man die Produktion

$$\mathrm{DeutscherSatz} \to \mathrm{Subjekt\ Prädikat\ Objekt}$$

verstehen als Aussage, die besagt:

„Ein Deutscher Satz ist aufgebaut aus Subjekt Prädikat Objekt.“

Graphische Darstellung:



Das Grundkonzept für die Anwendung von Produktionen einer KFG ist die „Ableitung“:

Definition 6.7 (Ableitung)**.** Sei $G = (T, N, S, P)$ eine KFG.

- Falls $A \to x$ eine Produktion in P ist und $u \in V^*$ und $v \in V^*$ beliebige Worte über der Symbolmenge $V = T \cup N$ sind, so schreiben wir

  $$uAv \Longrightarrow_G uxv \quad \text{(bzw. kurz: } uAv \Longrightarrow uxv\text{)}$$

  und sagen, dass uAv in einem **Ableitungsschritt** zu uxv umgeformt werden kann. Ableitungsschritt

- Eine **Ableitung** ist eine Folge von hintereinander angewendeten Ableitungsschritten. Für Worte $w \in V^*$ und $w' \in V^*$ schreiben wir Ableitung

  $$w \Longrightarrow_G^* w' \quad \text{(bzw. kurz: } w \Longrightarrow^* w'\text{)},$$

  um auszusagen, dass es eine endliche Folge von Ableitungsschritten gibt, die w zu w' umformt.

  Spezialfall: Diese Folge darf auch aus 0 Ableitungsschritten bestehen, d.h. f.a. $w \in V^*$ gilt: $w \Longrightarrow^* w$.

Beispiel 6.8. Sei G_K die „Klammer-Grammatik“ aus Beispiel 6.6. Beispiele für einzelne Ableitungsschritte:

- (Liste) $\Longrightarrow$ (Klammerung Liste)
- ((Liste) Liste) $\Longrightarrow$ (() Liste)

Ein Beispiel für eine Ableitung in G_K:

$$\begin{aligned}
\text{Klammerung} &\Longrightarrow \text{(Liste)} \\
&\Longrightarrow \text{(Klammerung Liste)} \\
&\Longrightarrow \text{(Klammerung Klammerung Liste)} \\
&\Longrightarrow \text{(Klammerung (Liste) Liste)} \\
&\Longrightarrow \text{((Liste)(Liste) Liste)} \\
&\Longrightarrow \text{(()(Liste) Liste)} \\
&\Longrightarrow \text{(()()Liste)} \\
&\Longrightarrow \text{(()())}
\end{aligned}$$

In jedem Schritt wird jeweils eine Produktion auf ein Nichtterminal der vorangehenden Symbolfolge angewandt. Obige Kette von Ableitungsschritten zeigt, dass

$$\text{Klammerung} \Longrightarrow^* \text{(()())}$$

Definition 6.9 (Sprache einer KFG)**.** Sei $G = (T, N, S, P)$ eine KFG. Die **von G erzeugte Sprache $L(G)$** ist die Menge aller Worte über dem Alphabet T, die aus dem Startsymbol S abgeleitet werden können. D.h.: Sprache einer KFG, $L(G)$

$$L(G) := \left\{ w \in T^* : S \Longrightarrow_G^* w \right\}.$$

Beispiel 6.10. Die KFG G_K aus Beispiel 6.6 definiert die Sprache $L(G_\text{K})$, die aus allen „wohlgeformten Klammerausdrücken“ besteht, die von einem Klammerpaar umschlossen werden. Beispielsweise gehören folgende Worte zu $L(G_\text{K})$:

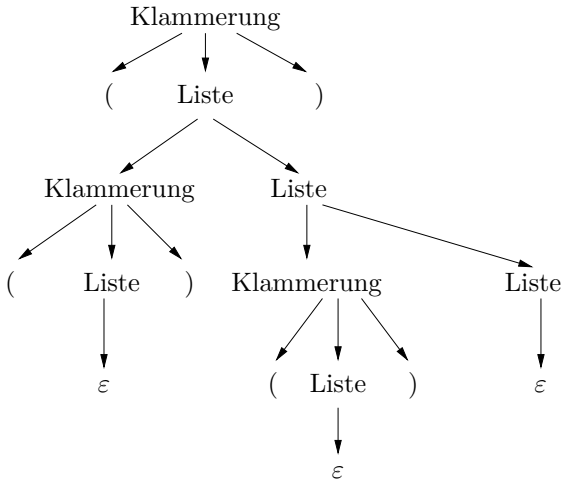
$$\text{()}\,, \qquad \text{(())}\,, \qquad \text{(()())}\,, \qquad \text{(((()))())}\,,$$

aber die Worte)(, (((, (()(, (Liste) gehören **nicht** zu $L(G_\text{K})$.

Beispiel 6.11. Sei G_{K} die „Klammer-Grammatik“ aus Beispiel 6.6. Die Ableitung

$$\begin{aligned}
\text{Klammerung} &\Longrightarrow \text{(Liste)} \\
&\Longrightarrow \text{(Klammerung Liste)} \\
&\Longrightarrow \text{((Liste) Liste)} \\
&\Longrightarrow \text{(() Liste)} \\
&\Longrightarrow \text{(() Klammerung Liste)} \\
&\Longrightarrow \text{(() Klammerung)} \\
&\Longrightarrow \text{(()(Liste))} \\
&\Longrightarrow \text{(()())}
\end{aligned}$$

wird durch den folgenden **Ableitungsbaum** dargestellt:



Ableitungsbäume

Sei $G = (T, N, S, P)$ eine KFG. Jede Ableitung $S \Longrightarrow_G^* w$ lässt sich als gerichteter Baum darstellen, bei dem

- jeder Knoten mit einem Symbol aus $T \cup N \cup \{\varepsilon\}$ markiert ist und
- bei dem die Kinder jedes Knotens eine festgelegte Reihenfolge haben (in der Zeichnung eines Ableitungsbaums werden von links nach rechts zunächst das „erste Kind“ dargestellt, dann das zweite, dritte etc.).

Die Wurzel des Baums ist mit dem Startsymbol S markiert. Jeder Knoten mit seinen Kindern repräsentiert die Anwendung einer Produktion aus P: Die Anwendung einer Produktion der Form $A \to x$ (mit $A \in N$ und $x \in V^*$) wird im Ableitungsbaum repräsentiert durch einen Knoten, der mit dem Symbol A markiert ist und der $|x|$ viele Kinder hat, so dass das i-te Kind mit dem i-ten Symbol von x markiert ist (f.a. $i \in \{1, \ldots, |x|\}$). *Spezialfall:* Ist x das leere Wort ε, so wird eine Anwendung der Produktion $A \to \varepsilon$ repräsentiert durch einen mit A markierten Knoten, der genau ein Kind hat, das mit ε markiert ist.

Beachte: Ein Ableitungsbaum kann mehrere Ableitungen repräsentieren. Beispielsweise repräsentiert der obige Ableitungsbaum auch die Ableitung

$$\begin{aligned}
\text{Klammerung} &\Longrightarrow (\text{Liste}) \\
&\Longrightarrow (\text{Klammerung Liste}) \\
&\Longrightarrow (\text{Klammerung Klammerung Liste}) \\
&\Longrightarrow ((\text{Liste}) \text{ Klammerung Liste}) \\
&\Longrightarrow ((\text{Liste})(\text{Liste}) \text{ Liste}) \\
&\Longrightarrow (()(\text{Liste}) \text{ Liste}) \\
&\Longrightarrow (()() \text{ Liste}) \\
&\Longrightarrow (()()) \ ,
\end{aligned}$$

in der gegenüber der ursprünglichen Ableitung aus Beispiel 6.11 einige Ableitungsschritte vertauscht sind. Im Ableitungsbaum wird von der konkreten Reihenfolge, in der die einzelnen Ableitungsschritte vorkommen, abstrahiert.

Im Folgenden betrachten wir einige weitere Beispiele für kontextfreie Grammatiken.

Beispiel 6.12 (Aussagenlogik)**.** Wir konstruieren eine KFG

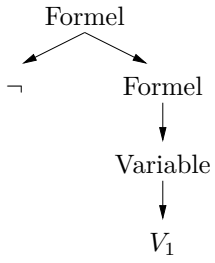
$$G_{\mathrm{AL}} \ = \ (T, N, S, P),$$

deren Sprache $L(G_{\mathrm{AL}})$ gerade die Menge aller aussagenlogischen Formeln ist, in denen nur Variablen aus $\{V_0, V_1, V_2\}$ vorkommen:

- Terminalsymbole $T \ := \ \big\{V_0, V_1, V_2, \mathbf{0}, \mathbf{1}, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\big\}$
- Nichtterminalsymbole $N := \big\{\text{Formel}, \text{Variable}, \text{Junktor}\big\}$
- Startsymbol $S := \text{Formel}$
- Produktionen

$$\begin{aligned}
P \ := \ \big\{&\text{Formel} \to \mathbf{0}, \ \text{Formel} \to \mathbf{1}, \ \text{Formel} \to \text{Variable}, \\
&\text{Formel} \to \neg\text{Formel}, \\
&\text{Formel} \to (\text{Formel Junktor Formel}), \\
&\text{Variable} \to V_0, \ \text{Variable} \to V_1, \ \text{Variable} \to V_2, \\
&\text{Junktor} \to \wedge, \ \text{Junktor} \to \vee, \ \text{Junktor} \to \rightarrow, \ \text{Junktor} \to \leftrightarrow\big\}
\end{aligned}$$

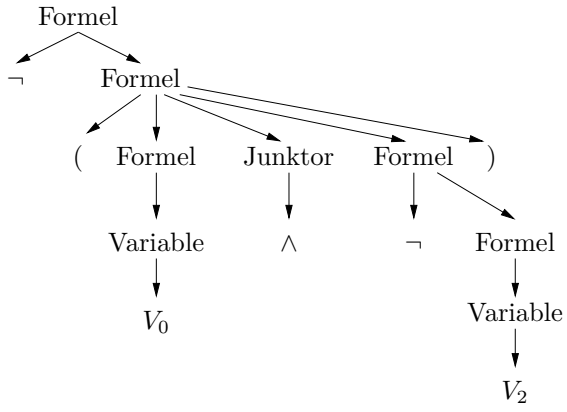
Beispiele für Ableitungsbäume:



Dieser Ableitungsbaum repräsentiert die Ableitung

$$\begin{aligned}
\text{Formel} &\Longrightarrow \neg\text{Formel} \\
&\Longrightarrow \neg\text{Variable} \\
&\Longrightarrow \neg V_1
\end{aligned}$$

Das durch diese(n) Ableitung(sbaum) erzeugte Wort in der Sprache $L(G_{\mathrm{AL}})$ ist die Formel $\neg V_1$.



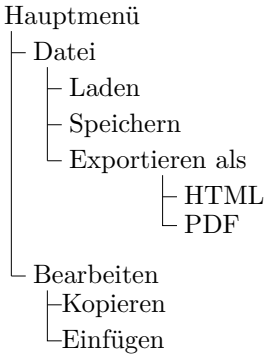
Dieser Ableitungsbaum repräsentiert die Ableitung

$$\begin{aligned}\text{Formel} &\Longrightarrow \neg\text{Formel}\\ &\Longrightarrow \neg(\text{Formel Junktor Formel})\\ &\Longrightarrow \neg(V_0 \text{ Junktor Formel})\\ &\Longrightarrow \neg(V_0 \wedge \text{Formel})\\ &\Longrightarrow \neg(V_0 \wedge \neg\text{Formel})\\ &\Longrightarrow \neg(V_0 \wedge \neg V_2).\end{aligned}$$

Das durch diese(n) Ableitung(sbaum) erzeugte Wort in der Sprache $L(G_{\text{AL}})$ ist die Formel $\neg(V_0 \wedge \neg V_2)$.

Beispiel 6.13 (Menü-Struktur)**.** In der graphischen Benutzungsoberfläche von vielen Software-Systemen werden oftmals „Menüs“ verwendet. Ein Menü besteht aus einem Menünamen und einer Folge von Einträgen. Jeder einzelne Eintrag besteht dabei aus einem Operationsnamen oder selbst wieder einem Menü.

Beispiel:



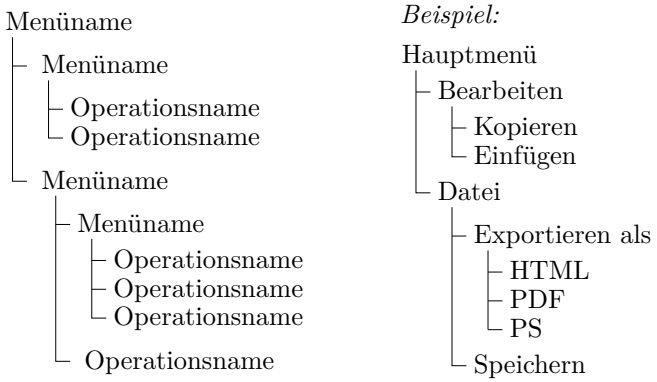
Zur Spezifizierung der Grund-Struktur solcher Menüs kann man folgende Grammatik $G_{\text{Menü}}$ verwenden:

$$G_{\text{Menü}} \;=\; (T, N, S, P)$$

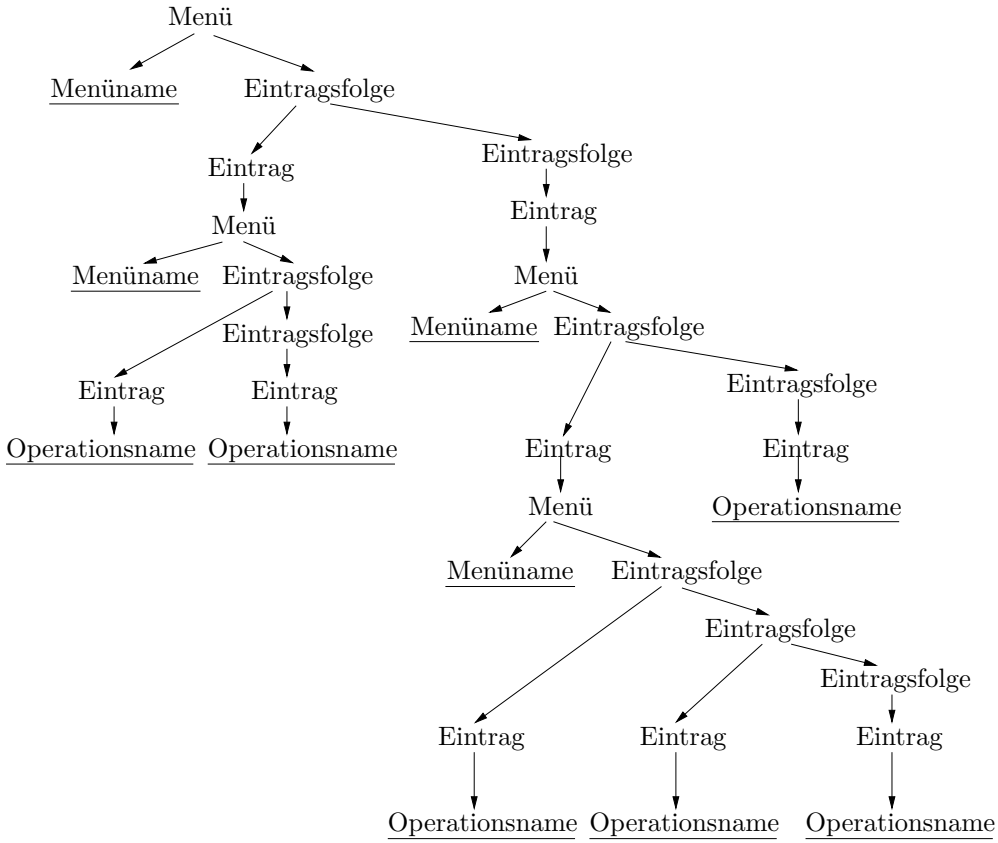
mit

- $T \;:=\; \{\text{Menüname}, \text{Operationsname}\}$
- $N \;:=\; \{\text{Menü}, \text{Eintragsfolge}, \text{Eintrag}\}$
- $S \;:=\; \text{Menü}$
- $P \;:=\; \{\, \text{Menü} \to \text{Menüname Eintragsfolge},$
  $\text{Eintragsfolge} \to \text{Eintrag},$
  $\text{Eintragsfolge} \to \text{Eintrag Eintragsfolge},$
  $\text{Eintrag} \to \text{Operationsname},$
  $\text{Eintrag} \to \text{Menü} \,\}$

Jeder Ableitungsbaum repräsentiert die Struktur eines Menüs. Ein Menü der Struktur



wird z.B. von folgendem Ableitungsbaum repräsentiert:



Beispiel 6.14 (HTML-Tabellen)**.**
HTML: Format zur Beschreibung von verzweigten Dokumenten im WWW.
Ein Bestandteil, der oft im Quell-Code von Internet-Seiten vorkommt, sind Tabellen. Z.B. wird der Eintrag

Tag	Zeit	Raum
Mi	8:00-11:00	Magnus-Hörsaal
Do	14:00-16:00	NM 10

durch HTML-Quelltext der folgenden Form erzeugt:

```
<table>
  <tr>
    <td> Tag </td>
    <td> Zeit </td>
    <td> Raum </td>
  </tr>
  <tr>
    <td> Mi </td>
    <td> 8:00-11:00 </td>
    <td> Magnus-Hörsaal </td>
  </tr>
  <tr>
    <td>Do </td>
    <td>14:00-16:00 </td>
    <td>NM10 </td>
  </tr>
</table>
```

<tr> steht für den Anfang einer Zeile der Tabelle,
</tr> steht für das Ende einer Zeile der Tabelle

<td> steht für den Anfang eines Eintrags,
</td> steht für das Ende eines Eintrags

Als Einträge in einzelnen Zellen der Tabelle kann z.B. Text stehen oder eine weitere Tabelle.

Im Folgenden konstruieren wir eine Grammatik

$$G_{\text{HTML-Tabellen}} \;=\; (T, N, S, P),$$

so dass die von $G_{\text{HTML-Tabellen}}$ erzeugte Sprache aus (möglicherweise geschachtelten) HTML-Tabellen besteht:

- $T \;:=\; \big\{$<table>, </table>, <tr>, </tr>, <td>, </td>, a, …, z, A, …, Z, 0, 1, …, 9, :, -, ␣, ä, ö, ü, ß, Ä, Ö, Ü$\big\}$
- $N \;:=\; \big\{$Tabelle, Zeile, Eintrag, Text, Zeilen, Einträge$\big\}$
- $S \;:=\;$ Tabelle
- $P \;:=\; \big\{$ Tabelle → <table>Zeilen</table>,
  Zeilen → Zeile, Zeilen → Zeile Zeilen,
  Zeile → <tr>Einträge</tr>,
  Einträge → Eintrag, Einträge → Eintrag Einträge,
  Eintrag → <td>Text</td>, Eintrag → Tabelle,
  Text → a, …, Text → z, Text → A, …, Text → Ü,
  Text → a Text, …, Text → z Text, Text → A Text, …, Text → Ü Text $\big\}$

Die oben angegebene Beispiel-HTML-Tabelle wird z.B. durch eine Ableitung erzeugt, die durch den Ableitungsbaum in Abbildung 6.5 repräsentiert wird.

Bemerkung 6.15. Typische Fragen bzgl. kontextfreien Grammatiken:

(a) Welche Sprachen können prinzipiell durch KFGs erzeugt werden, welche nicht?

Beispiel: Die Sprache $L_1 = \{a^n b^n : n \in \mathbb{N}\}$ wird von der KFG $G_1 = (T, N, S, P)$ mit $T = \{a, b\}, N = \{S\}$ und $P = \{S \to aSb, S \to \varepsilon\}$ erzeugt, d.h. $L_1 = L(G_1)$.

Notation: $a^n b^n$ ist eine Abkürzung für $\underbrace{aa\cdots a}_{n \text{ Mal}}\underbrace{bb\cdots b}_{n \text{ Mal}}$.

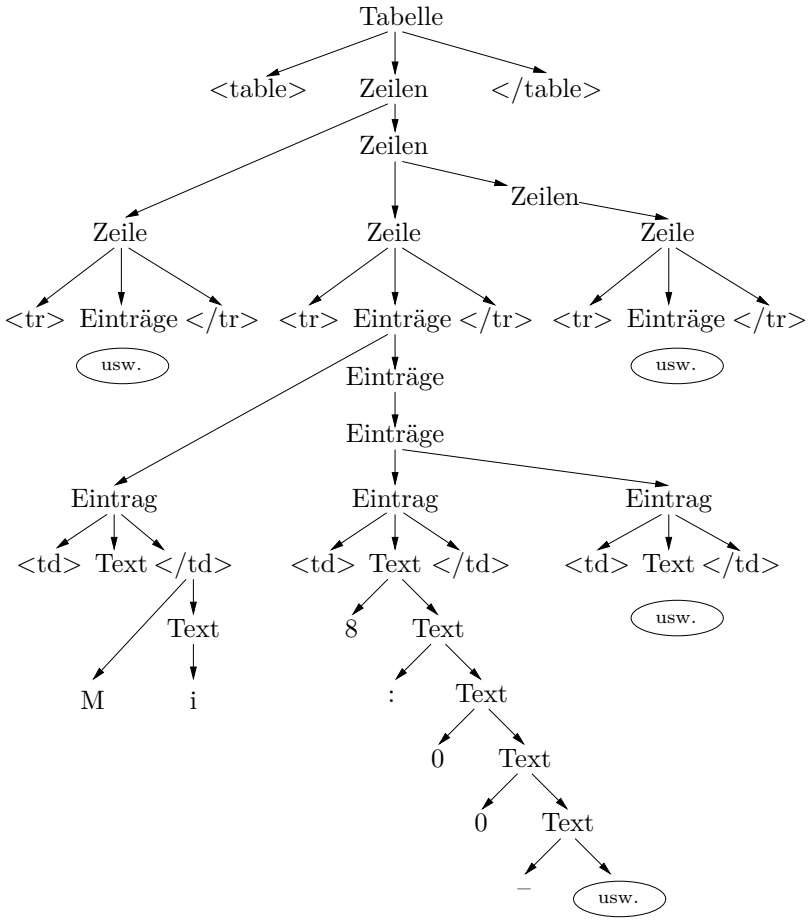


Abbildung 6.5: Ableitungsbaum für eine Beispiel-HTML-Tabelle

Satz. *Es gibt keine KFG, die genau die Sprache* $L_2 := \{a^n b^n c^n : n \in \mathbb{N}\}$ *erzeugt.*

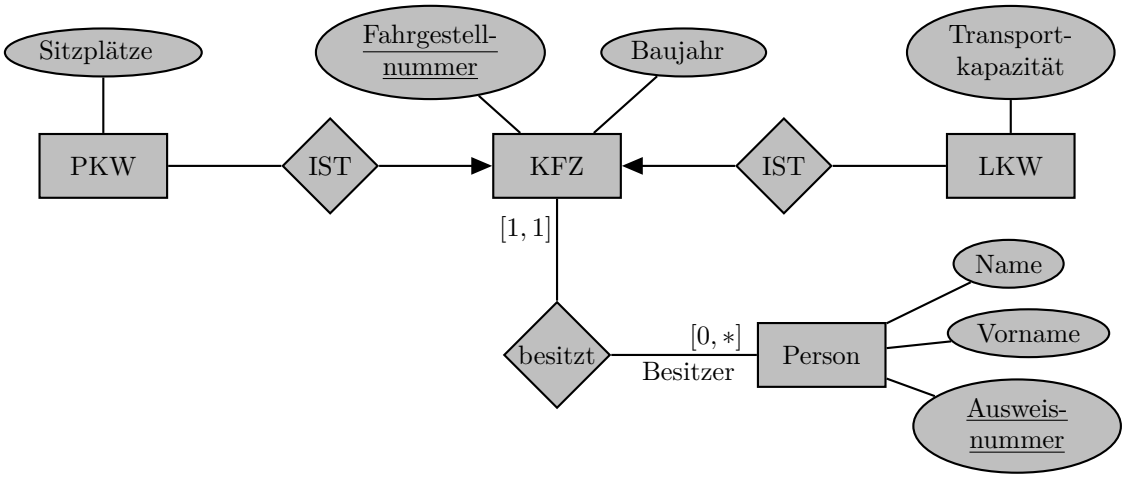
Beweis: In der Vorlesung GL-2.

(b) Gegeben sei eine KFG $G = (T, N, S, P)$ und ein Wort $w \in T^*$. Wie kann man herausfinden, ob $w \in L(G)$ ist, d.h. ob das Wort w zu der von G erzeugten Sprache gehört? Dies ist das so genannte **Wortproblem**.

Ein Algorithmus zum Lösen des Wortproblems für KFGs werden Sie in der Vorlesung GL-2 kennenlernen (den so genannten CYK-Algorithmus, der nach seinen Erfindern Cocke, Younger und Kasami benannt ist).

6.3 Übungsaufgaben zu Kapitel 6

Aufgabe 6.1. Es sei folgendes Entity-Relationship-Modell gegeben:



Weiterhin seien die folgenden konkreten Ausprägungen der einzelnen Entity-Typen PKW, LKW, KFZ und Person gegeben: PKW = $\{PKW1, PKW2\}$, LKW = $\{LKW\}$, KFZ = PKW $\cup$ LKW und Person = $\{Person1, Person2\}$ mit:

- *PKW1*: Fahrgestellnummer: 123421, Baujahr: 1999, Sitzplätze: 4
- *PKW2*: Fahrgestellnummer: 123123, Baujahr: 2003, Sitzplätze: 6
- *LKW*: Fahrgestellnummer: 123131, Baujahr: 1994, Transportkapazität: 7 Tonnen
- *Person1*: Ausweisnummer: 1234567890, Name: Meier, Vorname: Max
- *Person2*: Ausweisnummer: 9876543210, Name: Müller, Vorname: Martha

(a) Welche der folgenden Relationen sind zulässige Ausprägungen des Relationen-Typs „besitzt“?

a) $\{(Person1, PKW1), (Person2, PKW2), (Person1, LKW)\}$
b) $\{(Person1, PKW1), (Person1, PKW2), (Person1, LKW), (Person2, PKW2)\}$
c) $\{(Person1, PKW1), (Person1, PKW2), (Person1, LKW)\}$
d) $\{(Person1, PKW1), (Person2, LKW)\}$

(b) Würde es dem Modell widersprechen, wenn

(a) *Person1* und *Person2* den gleichen (Nach)Namen hätten?
(b) *PKW1* und *PKW2* die gleiche Fahrgestellnummer hätten?
(c) *PKW1* und *LKW* das gleiche Baujahr hätten?
(d) *PKW2* ein Entity vom Typ KFZ (d.h. $PKW2 \in$ KFZ), aber nicht vom Typ PKW (d.h. $PKW2 \notin$ PKW) wäre?

Begründen Sie jeweils Ihre Antworten!

Aufgabe 6.2. In einer Bibliothek gibt es verschiedene Exemplare von Büchern. Die Bücher sind eindeutig über Ihre ISBN gekennzeichnet und besitzen darüberhinaus noch einen Titel, eine Seitenanzahl und ein Erscheinungsjahr. Die Exemplare eines bestimmten Buches sind fortlaufend nummeriert und weiterhin durch eine Inventarnummer und den Standort charakterisiert. Für jedes Buch wird vermerkt, welche Exemplare dieses Buches in der Bibliothek vorhanden sind. Bücher sind in einem Verlag erschienen, von dem der Name und der Verlagsort registriert wird.

Für jeden Bibliotheksbenutzer ist der Name, Vorname, Wohnort und das Geburtsdatum gespeichert. Benutzer können einzelne Buchexemplare ausleihen, wobei jeweils das Datum der Ausleihe registriert wird. Weiterhin können Benutzer einzelne Bücher vorbestellen, wobei auch hier das Datum der Vorbestellung registriert wird.

Geben Sie ein Entity-Relationship-Modell für den oben beschriebenen Sachverhalt an! Geben Sie auch Kardinalitäten und Schlüsselattribute an!

Aufgabe 6.3. Gegeben sei folgende Grammatik $G = (T, N, S, P)$ mit

- $T = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, \mathtt{@}, \mathtt{.}\}$
- $N = \{S, X, Y, Z\}$
- $P = \big\{S \to X\mathtt{@}Y,$
  $Y \to X\,\mathtt{.}\,Z,\ Z \to X\,\mathtt{.}\,Z,\ Z \to X,$
  $X \to \mathtt{a}X,\ X \to \mathtt{b}X,\ \ldots, X \to \mathtt{z}X,\ X \to \mathtt{a},\ X \to \mathtt{b},\ \ldots, X \to \mathtt{z}\big\}$

(a) Überprüfen Sie für jedes der folgenden Worte, ob es in der von G erzeugten Sprache liegt. Wenn ja, dann geben Sie einen Ableitungsbaum für dieses Wort an; ansonsten begründen Sie, warum das Wort nicht zur Sprache gehört.

(i) `meier@web.de`
(ii) `Meier@web.de`
c) `max.meier@web.de`
d) `meier@www.web.de`
e) `root@localhost`

(b) Beschreiben Sie in Worten, welche Sprache $L(G)$ von der Grammatik G erzeugt wird.

Aufgabe 6.4. Sei σ eine Signatur mit einem zweistelligen Relationssymbol $\dot{E}$, einem zweistelligen Funktionssymbol $\dot{f}$, einem einstelligen Funktionssymbol $\dot{g}$ und einem Konstantensymbol $\dot{c}$.

(a) Konstruieren Sie eine Grammatik G_{Terme}, so dass $L(G_{\text{Terme}})$ genau die Menge aller σ-Terme ist, in denen nur Variablen aus $\{v_0, v_1, v_2\}$ vorkommen. Geben Sie einen Ableitungsbaum für den Term $t := \dot{f}\big(v_0, \dot{g}(\dot{f}(v_2, \dot{c}))\big)$ an.

(b) Konstruieren Sie eine Grammatik $G_{\text{FO}[\sigma]}$, so dass $L(G_{\text{FO}[\sigma]})$ genau die Menge aller FO[σ]-Formeln ist, in denen nur Variablen aus $\{v_0, v_1, v_2\}$ vorkommen. Geben Sie einen Ableitungsbaum für die Formel $\varphi := \forall v_0\big(\dot{E}(v_0, \dot{f}(v_2, \dot{c})) \to \dot{g}(v_1) \doteq v_0\big)$ an.