0 Einleitung

In diesem Kapitel werden die dieser Vorlesung zu Grunde liegenden Fragestellungen anhand von Beispielen aus den Bereichen *Datenbanken* und *Komplexitätstheorie* vorgestellt. Insbesondere werden hier Formeln der *Logik erster Stufe* – oder Erweiterungen dieser Logik – auf informelle Weise benutzt und erst in späteren Kapiteln präzise eingeführt.

0.1 Logiken als Datenbank-Anfragesprachen

Als Beispiel-Datenbank betrachten wir eine *Bibliothek*, die aus einer Sammlung von Artikeln besteht. Genauer gesagt hat unsere Datenbank u.a. eine Relation $\mathsf{Artikel}$ mit den Attributen Titel, Autor, $\mathsf{Konferenz}$, Jahr.

0.1 Beispiele. In unserer Beispiel-Datenbank besteht die Relation $\mathsf{Artikel}$ aus den in Tabelle 0.1 angegebenen Tupeln.

> **Anfrage 1:** Als erstes wollen wir Titel und Jahr aller bei einer LICS-Konferenz veröffentlichten Arbeiten wissen.

In der in der Praxis weit verbreiteten Datenbank-Anfragesprache SQL lässt sich diese Anfrage beispielsweise durch folgende Anweisung formulieren:

```
SELECT DISTINCT Titel, Jahr
  FROM Artikel
  WHERE Konferenz = 'LICS'
```

Dieselbe Anfrage lässt sich auch durch folgende Formel der Logik erster Stufe (kurz: FO-Formel) $\varphi(t,j)$ beschreiben:

$$\varphi_1(t,j) \;:=\; \exists a\, \exists k\, \big(\mathsf{Artikel}(t,a,k,j) \,\wedge\, k = \text{'LICS'}\big).$$

Als nächstes wollen wir folgende Anfrage stellen:

> **Anfrage 2:** Alle Autoren, die (laut unserer Datenbank) bei keiner LICS-Konferenz veröffentlicht haben.

Formal lässt sich diese Anfrage folgendermaßen ausdrücken:

Relation Artikel:

Titel	Autor	Konferenz	Jahr
Typechecking...	Neven	LICS	2001
Typechecking...	Alon	LICS	2001
Typechecking...	Suciu	LICS	2001
Typechecking...	Vianu	LICS	2001
Typechecking...	Milo	LICS	2001
Learnability...	Grohe	STACS	2002
Learnability...	Turan	STACS	2002
Expressive...	Kreutzer	LICS	2002
Succinctness...	Schweikardt	LICS	2004
Succinctness...	Grohe	LICS	2004
Investing in Research	Gates	SIGMOD	1998
Harry Potter and...	Rowling	none	2003
Relational...	Immerman	STOC	1982
Sure monochromatic...	Erdös	Acta Arith.	1996
Sure monochromatic...	Alon	Acta Arith.	1996
Two lower bounds...	Turan	STOC	1986
Two lower bounds...	Ajtai	STOC	1986
On Turan's theorem...	Ajtai	Combinatorica	1981
On Turan's theorem...	Erdös	Combinatorica	1981
Tailoring...	Grädel	ICALP	1994
Tailoring...	Gurevich	ICALP	1994
An n! lower bound...	Immerman	LICS	2001
An n! lower bound...	Adler	LICS	2001
Two-variable...	Grädel	LICS	1999
Two-variable...	Rosen	LICS	1999

Tabelle 0.1: Beispiel-Relation in einer Bibliotheks-Datenbank

In der Datenbankanfragesprache SQL:

```
SELECT DISTINCT A.Autor
  FROM Artikel A
  WHERE A.Autor NOT IN ( SELECT B.Autor
                           FROM Artikel B
                           WHERE B.Konferenz = 'LICS' )
```

In der Logik erster Stufe (FO):

$$\begin{aligned} \varphi_2(a) \;:=\; & \exists t\,\exists k\,\exists j\;\; \mathsf{Artikel}(t,a,k,j) \;\wedge \\ & \forall t'\,\forall k'\,\forall j'\; (\, \mathsf{Artikel}(t',a,k',j') \;\rightarrow\; \neg\, k' = \text{'LICS'}\,) \end{aligned}$$

Unsere dritte Beispiel-Anfrage lautet:

Anfrage 3: Alle Autoren, deren *Erdös-Nummer* $\leqslant 2$ ist.

Die *Erdös-Nummer* ist nach dem Mathematiker Paul Erdös ($*$1913 in Budapest, †1996 in Warschau) benannt, der mehr als 1500 Arbeiten in den Gebieten Zahlentheorie, Graphentheorie, Kombinatorik, diskrete Mathematik und Grundlagen der Informatik veröffentlicht hat. Paul Erdös selbst hat die *Erdös-Nummer* 0, seine Koautoren haben Erdös-Nummer 1, die Koautoren seiner Koautoren haben Erdös-Nummer 2 usw. Laut unserer Beispiel-Datenbank ist etwa die Erdös-Nummer von Miklos Ajtai 1, die von Martin Grohe 3, die von Bill Gates 4 und die von J.K. Rowling ∞.
Obige Anfrage 3 lässt sich formal folgendermaßen ausdrücken:

In SQL:

```
SELECT DISTINCT D.Autor
  FROM Artikel A, Artikel B, Artikel C, Artikel D
  WHERE ( A.Autor = 'Erdös'  AND
          B.Titel = A.Titel  AND
          C.Autor = B.Autor  AND
          D.Titel = C.Titel )
```

In FO:

$$\begin{aligned} \varphi_3(a) \;:=\; & \exists t_1\,\exists a_1\,\exists k_1\,\exists j_1\;\Big(\\ & \quad \mathsf{Artikel}(t_1,\text{'Erdös'},k_1,j_1) \;\wedge\; \mathsf{Artikel}(t_1,a_1,k_1,j_1) \wedge \\ & \quad \exists t_2\,\exists k_2\,\exists j_2\;(\, \mathsf{Artikel}(t_2,a_1,k_2,j_2) \;\wedge\; \mathsf{Artikel}(t_2,a,k_2,j_2)\,)\,\Big) \end{aligned}$$

Die Logik erster Stufe (FO) wird oft auch als der "logische Kern von SQL" bezeichnet, da sehr viele grundlegende SQL-Anfragen auch in FO ausgedrückt werden können, FO mathematisch gut untersucht ist und – im Gegensatz zur Sprache SQL – eine überschaubare (kurze) Syntax- und Semantik-Definition hat.

Als letztes Beispiel betrachten wir folgende Anfrage:

Anfrage 4: Alle Autoren mit Erdös-Nummer $< \infty$.

Die folgende Formulierung dieser Anfrage in SQL definiert *rekursiv* die Menge E-Autoren aller Autoren mit
– Erdös-Nummer 0
– Erdös-Nummer 1
– Erdös-Nummer 2
– $\vdots$

In SQL:

```
WITH RECURSIVE E-Autoren(Autor) AS
 ( SELECT Autor
     FROM Artikel
     WHERE Autor='Erdös'
   UNION
   SELECT B.Autor
     FROM E-Autoren E, Artikel A, Artikel B
     WHERE ( A.Autor = E.Autor AND B.Titel = A.Titel )
 )
SELECT *
  FROM E-Autoren
```

In FO lässt sich diese Anfrage nicht formulieren (einen Beweis dafür werden wir später im Kapitel über *Ehrenfeucht-Fraïssé Spiele* kennenlernen). Um auch "rekursive" Anfragen definieren zu können, kann man Erweiterungen der Logik erster Stufe, beispielsweise *Fixpunktlogiken* oder die *Logik zweiter Stufe* verwenden.

Wichtige Fragestellungen bzgl. Logik und Datenbanken:

Gegeben eine Datenbank-Anfragesprache bzw. Logik,

- welche Art von Anfragen kann man in der Sprache stellen?
- welche nicht?
- wie aufwendig ist es, eine Anfrage in einer Datenbank auszuwerten?

0.2 Logiken zur Beschreibung von Berechnungsproblemen

0.2 Beispiel (3-Färbbarkeit von Graphen).

Zur Erinnerung:
Ein Graph $G = (V, E)$ heißt *3-färbbar*, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$$\begin{array}{rcl} G \text{ 3-färbbar} & \iff & \text{ex. } R \subseteq V, G \subseteq V, B \subseteq V, \text{ so dass f.a. } u, v \in V \text{ gilt:} \\ & & \text{falls } E(u,v), \text{ so} \\ & & \neg\Big(\big(R(u) \wedge R(v)\big) \vee \big(G(u) \wedge G(v)\big) \vee \big(B(u) \wedge B(v)\big)\Big) \\ & \iff & G \models \Phi_{\text{3-col}}, \end{array}$$

wobei die Formel $\Phi_{\text{3-col}}$ folgendermaßen definiert ist:

$$\begin{aligned} \exists R\, \exists G\, \exists B\, \forall v\, \Big(& \big(R(v) \wedge \neg G(v) \wedge \neg B(v)\big) \vee \\ & \big(\neg R(v) \wedge G(v) \wedge \neg B(v)\big) \ \vee\ \big(\neg R(v) \wedge \neg G(v) \wedge B(v)\big)\Big) \wedge \\ \forall u\, \forall v\ \ E(u,v)\ \rightarrow\ \neg & \Big(\big(R(u) \wedge R(v)\big) \vee \big(G(u) \wedge G(v)\big) \vee \big(B(u) \wedge B(v)\big)\Big). \end{aligned}$$

$\Phi_{\text{3-col}}$ ist eine Formel der *Logik zweiter Stufe* (SO), die eine wichtige Rolle in dieser Vorlesung spielt (…sie wird später noch formal eingeführt). Zusammenfassend haben wir gesehen, dass $\Phi_{\text{3-col}}$ eine Formel ist, die das folgende Berechnungsproblem beschreibt:

> 3-FÄRBBARKEIT
> *Eingabe:* Ein Graph G.
> *Frage:* Ist G 3-färbbar?

0.3 Beispiel (Erreichbarkeit).
Das Erreichbarkeitsproblem ist folgendermaßen definiert:

> ERREICHBARKEIT
> *Eingabe:* Ein Graph G und 2 Knoten s, t von G.
> *Frage:* Gibt es in G einen Pfad von s nach t?

Die Antwort auf obige Frage ist genau dann "ja", wenn gilt: $(G, s, t) \models \Phi_{\text{reach}}$, wobei

$$\Phi_{\text{reach}} \ :=\ \underbrace{[\mathbf{tc}_{x;y}\, E(x,y)]}(s,t)$$

"verallgemeinerter Quantor", der den transitiven Abschluss der Relation E bezeichnet. Der $\mathbf{tc}$-Operator wird später in dieser Vorlesung auch formal eingeführt.

Wichtige Fragestellungen bzgl. Logik und Komplexitätstheorie:

- Welche *Komplexitätsklassen* werden durch welche *Logiken* charakterisiert?
- Gegeben eine Logik (die eine Komplexitätsklasse charakterisiert), wie kann man für ein bestimmtes Problem zeigen, dass es *nicht* in der Logik beschreibbar ist?

0.3 Logiken zur Hardware- und Prozess-Spezifikation und Verifikation

Bestimmte Aspekte von Hardware-Komponenten oder Prozessen werden oft als so genannte *Transitionssysteme* (das sind gefärbte Graphen) modelliert. Dies hat zur Folge, dass Eigenschaften der Hardware-Komponenten bzw. Prozesse durch Formeln einer Logik beschrieben, d.h. *spezifiziert* werden können. Ein Ziel ist nun, automatisch zu testen, ob die zu untersuchende Komponente bestimmte erwünschte Eigenschaften hat – beispielsweise, ob eine Drucker-Warteschlange die Eigenschaft hat, dass jeder Druckjob irgendwann auch wirklich gedruckt wird.

Auch in diesem Bereich treten ähnliche Fragestellungen wie in den beiden vorherigen Bereichen auf:

Wichtige Fragestellungen bzgl. Spezifikation und Verifikation:

- Welche *Eigenschaften* können durch welche *Logiken* spezifiziert werden?
- Gegeben ein Transitionssystem und eine Formel einer bestimmten Logik, wie schwer ist es, zu entscheiden, ob das Transitionssystem die Formel erfüllt?

0.4 Aufbau der Vorlesung

Die Vorlesung *Logik und Komplexität* wird voraussichtlich aus den folgenden Kapiteln bestehen:

0. Einleitung
1. Grundlagen
2. Deskriptive Komplexität
3. Ehrenfeucht-Fraïssé Spiele
4. Fixpunktlogiken
5. Baumautomaten
6. Auswertungsspiele

0.5 Literatur

[EF] H.-D. Ebbinghaus und J. Flum. *Finite Model Theory*. Springer-Verlag, 2te Auflage, 1999.

[I] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[L] L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, 2004.

[G] E. Grädel. Finite Model Theory and Descriptive Complexity. Der Artikel wird im Buch [F] erscheinen.

[K] P. Kolaitis. On the expressive power of logics on finite models. Der Artikel wird im Buch [F] erscheinen.

[F] E. Grädel, P. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema und S. Weinstein. *Finite Model Theory and Its Applications*. Springer-Verlag, 2005. Eine Vorabversion ist online unter `http://www.cis.upenn.edu/~weinstein/fmta.html` erhältlich.

1 Gundlagen

Wir schreiben $\mathbb{Z}$ für die Menge der ganzen Zahlen, $\mathbb{N}$ für die Menge $\{0, 1, 2, \ldots\}$ der natürlichen Zahlen und $\mathbb{N}_{\geqslant 1}$ für die Menge der *positiven* natürlichen Zahlen. $\mathbb{R}$ bezeichnet die Menge der reellen Zahlen und $\mathbb{R}_{\geqslant 0}$ die Menge der reellen Zahlen $\geqslant 0$.
Ist M eine Menge, so schreiben wir $\mathrm{Pot}(M)$ für die *Potenzmenge* von M, d.h.

$$\mathrm{Pot}(M) \;=\; \{X \mid X \subseteq M\}.$$

Sind A und B Mengen, $f : A \to B$ eine Funktion, $\vec{a} = (a_1, .., a_k) \in A^k$ und $R \subseteq A^k$, für ein $k \in \mathbb{N}_{\geqslant 1}$, so setzen wir

$$f(\vec{a}) \;:=\; \big(f(a_1), .., f(a_k)\big) \;\in B^k \qquad \text{und} \qquad f(R) \;:=\; \{f(\vec{a}) \mid \vec{a} \in R\} \;\subseteq B^k.$$

1.1 Logik erster Stufe (FO)

Dieser Abschnitt gibt eine sehr knappe Einführung in einige Grundbegriffe der Logik erster Stufe. Mehr zu diesem Thema findet sich im Skript zur Vorlesung *Theoretische Informatik I* von Prof. Grohe; siehe `http://www.informatik.hu-berlin.de/logik/lehre/WS04-05/ThI1/folien.html`. Für eine umfassende Einführung in die Logik erster Stufe sei auf das Buch

> *Einführung in die mathematische Logik*
> von H.-D. Ebbinghaus, J. Flum und W. Thomas, Springer-Verlag

verwiesen.

1.1 Definition (Signatur, Struktur).

(a) Eine *Signatur* (oder *Symbolmenge*, *Vokabular*) ist eine endliche Menge

$$\sigma \;=\; \{R_1, .., R_k, c_1, .., c_\ell\},$$

für $k, \ell \in \mathbb{N}$.
Die Symbole $R_1, .., R_k$ heißen *Relationssymbole*, die Symbole $c_1, .., c_\ell$ heißen *Konstantensymbole*.
Jedes R_i hat eine feste *Stelligkeit* (Arität) $ar(R_i) \in \mathbb{N}$.

(b) Eine σ-*Struktur*

$$\mathfrak{A} \;=\; (A, \sigma^{\mathfrak{A}}) \;=\; (A, R_1^{\mathfrak{A}}, .., R_k^{\mathfrak{A}}, c_1^{\mathfrak{A}}, .., c_\ell^{\mathfrak{A}})$$

besteht aus

- einer Menge A, dem *Universum* (oder *Träger*) von $\mathfrak{A}$,
- einer $ar(R_i)$-stelligen Relation $R_i^{\mathfrak{A}} \subseteq A^{ar(R_i)}$, für jedes $i \leqslant k$, und
- einem Element $c_j^{\mathfrak{A}} \in A$, für jedes $j \leqslant \ell$.

$\mathfrak{A}$ heißt *endlich*, falls das Universum A endlich ist.

1.2 Beispiele.

- Ein Graph $G = (V, E^G)$ ist eine $\{E\}$-Sturktur, wobei E ein 2-stelliges Relationssymbol ist.
- $\mathfrak{N} := (\mathbb{N}, <^{\mathfrak{N}}, +^{\mathfrak{N}}, \times^{\mathfrak{N}}, 0^{\mathfrak{N}}, 1^{\mathfrak{N}})$ ist eine $\{<, +, \times, 0, 1\}$-Struktur, wobei $<$ ein 2-stelliges Relationssymbol, $+$ und $\times$ 3-stellige Relationssymbole und 0 und 1 Konstantensymbole sind. Hier sollen $0^{\mathfrak{N}}$ und $1^{\mathfrak{N}}$ die natürlichen Zahlen 0 und 1 bezeichnen, $<^{\mathfrak{N}}$ soll die natürliche lineare Ordnung auf $\mathbb{N}$ bezeichnen, und $+^{\mathfrak{N}}$ bzw. $\times^{\mathfrak{N}}$ die Graphen der Addition bzw. Multiplikation, d.h.

$$+^{\mathfrak{N}} := \{(a,b,c) \in \mathbb{N}^3 \mid a + b = c\} \qquad \text{und} \qquad \times^{\mathfrak{N}} := \{(a,b,c) \in \mathbb{N}^3 \mid a \cdot b = c\}.$$

- Unsere Bibliotheksdatenbank aus Beispiel 0.1 ist eine $\{\mathsf{Artikel}\}$-Struktur, wobei $\mathsf{Artikel}$ ein 4-stelliges Relationssymbol ist. Das Universum dieser Struktur ist eine Teilmenge der Menge aller Worte über den Zeichen des ASCII-Alphabets.

1.3 Definition (Homomorphismus, Isomorphismus).
Seien $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen, wobei σ eine Signatur sei.

(a) Ein *Homomorphismus* $h : \mathfrak{A} \to \mathfrak{B}$ ist eine Abbildung $h : A \to B$, für die gilt:
   - $h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$, für alle Konstantensymbole $c \in \sigma$,
   - $h(R^{\mathfrak{A}}) \subseteq R^{\mathfrak{B}}$, für alle Relationssymbole $R \in \sigma$.[1]

(b) Ein Homomorphismus $h : \mathfrak{A} \to \mathfrak{B}$ heißt *Isomorphismus*, falls
   - h eine *Bijektion* von A nach B ist und
   - $\vec{a} \in R^{\mathfrak{A}} \iff h(\vec{a}) \in R^{\mathfrak{B}}$,
     für alle Relationssymbole $R \in \sigma$ und alle $\vec{a} \in A^{ar(R)}$.

(c) $\mathfrak{A}$ und $\mathfrak{B}$ heißen *isomorph*, falls es einen Isomorphismus $h : \mathfrak{A} \to \mathfrak{B}$ gibt.
Wir schreiben dann kurz $\mathfrak{A} \cong \mathfrak{B}$ (oder genauer $h : \mathfrak{A} \cong \mathfrak{B}$).

1.4 Beispiel. Die $\{<, 0, 1\}$-Struktur $(\mathbb{N}, <^{\mathfrak{N}}, 0^{\mathfrak{N}}, 1^{\mathfrak{N}})$ ist isomorph zur Struktur

$$\mathfrak{A} = (\mathbb{Z}_{\leqslant 0}, <^{\mathfrak{A}}, 0^{\mathfrak{A}}, 1^{\mathfrak{A}})$$

mit $\mathbb{Z}_{\leqslant 0} := \{z \in \mathbb{Z} \mid z \leqslant 0\}$, $R^{\mathfrak{A}} := \{(a,b) \in \mathbb{Z}_{\leqslant 0}^2 \mid a > b\}$, $0^{\mathfrak{A}} := 0^{\mathfrak{N}}$ und $1^{\mathfrak{A}} := -1$. Die Abbildung $h : \mathbb{N} \to \mathbb{Z}_{\leqslant 0}$ mit $h(n) := -n$, für alle $n \in \mathbb{N}$, liefert einen Isomorphismus von $(\mathbb{N}, <^{\mathfrak{N}}, 0^{\mathfrak{N}}, 1^{\mathfrak{N}})$ nach $\mathfrak{A}$.

[1] Zur Erinnerung: $h(R^{\mathfrak{A}}) = \big\{ \big(h(a_1), \ldots, h(a_k)\big) \mid (a_1, \ldots, a_k) \in R^{\mathfrak{A}} \big\}$, wobei $k := ar(R)$.

1.5 Definition (Syntax der Logik erster Stufe FO). Sei σ eine Signatur.

(a) $\text{Var}_1 := \{\text{var}_i \mid i \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge alle *Variablen erster Stufe* (oder *Individuenvariablen*).

(b) Die Formelmenge FO$[\sigma]$ ist induktiv wie folgt definiert:

(A1) $R(v_1, .., v_k)$ gehört zu FO$[\sigma]$, für alle Relationssymbole $R \in \sigma$, $k := ar(R)$, $v_1, .., v_k \in \text{Var}_1 \cup \{c \in \sigma \mid c \text{ ist ein Konstantensymbol}\}$.

(A2) $v = v'$ gehört zu FO$[\sigma]$, für alle $v, v' \in \text{Var}_1 \cup \{c \in \sigma \mid c \text{ ist ein Konstantensymbol}\}$.

(BC) Sind $\psi, \varphi_1, \varphi_2$ Formeln in FO$[\sigma]$, so gehören auch die folgenden Formeln zu FO$[\sigma]$:
- $\neg\psi$ (Negation)
- $(\varphi_1 \lor \varphi_2)$ (Oder)
- $(\varphi_1 \land \varphi_2)$ (Und).
- $(\varphi_1 \to \varphi_2)$ (Implikation)
- $(\varphi_1 \leftrightarrow \varphi_2)$ (genau dann wenn)

(Q1) Ist ψ eine Formel in FO$[\sigma]$ und $x \in \text{Var}_1$, so gehören auch folgende Formeln zu FO$[\sigma]$:
- $\exists x\ \psi$ (Existenzquantor)
- $\forall x\ \psi$ (Allquantor).

(c) Die mit $(A1)$ und $(A2)$ gebildeten Formeln heißen *atomare* σ*-Formeln*.

1.6 Definition (Belegungen und Interpretationen). Sei σ eine Signatur.

(a) Eine σ*-Interpretation* $\mathfrak{I} = (\mathfrak{A}, \beta)$ besteht aus einer σ-Struktur $\mathfrak{A}$ und einer *Belegung* $\beta : \text{Var}_1 \to A$, die jeder Variablen einen Wert aus dem Universum von $\mathfrak{A}$ zuordnet.
Oft erweitern wir den Definitionsbereich von β um die Konstantensymbole aus σ und setzen $\beta(c) := c^{\mathfrak{A}}$, für alle Konstantensymbole $c \in \sigma$.

(b) Ist $x \in \text{Var}_1$, $a \in A$ und $\beta : \text{Var}_1 \to A$, so ist die Belegung $\beta\frac{a}{x} : \text{Var}_1 \to A$ folgendermaßen definiert: $\beta\frac{a}{x}(y) := \begin{cases} a & \text{falls } y = x \\ \beta(y) & \text{sonst.} \end{cases}$

1.7 Definition (Semantik der Logik erster Stufe). Sei σ eine Signatur, $\mathfrak{I} = (\mathfrak{A}, \beta)$ eine σ-Interpretation und φ eine FO$[\sigma]$-Formel. Die *Modellbeziehung* $\mathfrak{I} \models \varphi$ (in Worten: $\mathfrak{I}$ *erfüllt* φ) ist folgendermaßen per Induktion über den Formelaufbau definiert:

- Falls φ gemäß Regel (A1) gebildet ist, so $\mathfrak{I} \models \varphi :\iff \big(\beta(v_1), .., \beta(v_k)\big) \in R^{\mathfrak{A}}$.
- Falls φ gemäß Regel (A2) gebildet ist, so $\mathfrak{I} \models \varphi :\iff \beta(v) = \beta(v')$.
- Falls φ gemäß Regel (BC) gebildet ist und φ von der Form
  - $\neg\psi$, so $\mathfrak{I} \models \varphi :\iff$ *nicht* $\mathfrak{I} \models \psi$ (kurz: $\mathfrak{I} \not\models \psi$).
  - $(\varphi_1 \lor \varphi_2)$, so $\mathfrak{I} \models \varphi :\iff \big(\ \mathfrak{I} \models \varphi_1 \text{ oder } \mathfrak{I} \models \varphi_2\ \big)$.
  - $(\varphi_1 \land \varphi_2)$, so $\mathfrak{I} \models \varphi :\iff \big(\ \mathfrak{I} \models \varphi_1 \text{ und } \mathfrak{I} \models \varphi_2\ \big)$.

– $(\varphi_1 \rightarrow \varphi_2)$, so $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\; \big(\text{wenn } \mathfrak{I} \models \varphi_1, \text{ dann auch } \mathfrak{I} \models \varphi_2\,\big)$.
– $(\varphi_1 \leftrightarrow \varphi_2)$, so $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\; \big(\,\mathfrak{I} \models \varphi_1 \text{ genau dann, wenn } \mathfrak{I} \models \varphi_2\,\big)$.

- Falls φ gemäß Regel (Q1) gebildet ist und φ von der Form
  – $\exists x\, \psi$, so $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\;$ es gibt ein $a \in A$, so dass $(\mathfrak{A}, \beta\frac{a}{x}) \models \psi$.
  – $\forall x\, \psi$, so $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\;$ für alle $a \in A$ gilt $(\mathfrak{A}, \beta\frac{a}{x}) \models \psi$.

1.8 Definition (Freie Variablen einer Formel). Sei σ eine Signatur.

(a) Die Menge *frei*(φ) der *freien Variablen* einer FO$[\sigma]$-Formel φ ist induktiv folgendermaßen definiert:

- Falls φ gemäß Regel (A1) gebildet ist, so ist *frei*$(\varphi) := \{v_1, \dots, v_k\} \cap \text{Var}_1$.
- Falls φ gemäß Regel (A2) gebildet ist, so ist *frei*$(\varphi) := \{v, v'\} \cap \text{Var}_1$.
- Falls φ gemäß Regel (BC) gebildet ist, so ist

$$\mathit{frei}(\varphi) := \begin{cases} \mathit{frei}(\psi) & \text{falls } \varphi \text{ von der Form } \neg\psi \\ \mathit{frei}(\varphi_1) \cup \mathit{frei}(\varphi_2) & \text{falls } \varphi \text{ von der Form } (\varphi_1 * \varphi_2), \text{ für } * \in \{\vee, \wedge \rightarrow, \leftrightarrow\}. \end{cases}$$

- Falls φ gemäß Regel (Q1) gebildet ist, so ist *frei*$(\varphi) :=$ *frei*$(\psi) \setminus \{x\}$.

(b) Eine FO$[\sigma]$-Formel φ heißt *Satz*, falls φ keine freien Variablen hat, d.h. *frei*$(\varphi) = \varnothing$.

1.9 Notation.

(a) Wir werden oft Symbole wie $x, y, z, u, v, x_1, x_2, x_3, \dots$ verwenden, um Variablen erster Stufe zu bezeichnen.

(b) Wir schreiben $\varphi(x_1, \dots, x_s)$ um anzudeuten, dass *frei*$(\varphi) = \{x_1, \dots, x_s\}$.

(c) Ist $\varphi(x_1, \dots, x_s)$ eine FO$[\sigma]$-Formel, $\mathfrak{A}$ eine σ-Struktur und $a_1, \dots, a_s \in A$, so schreiben wir

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

falls die Formel φ in der Struktur $\mathfrak{A}$ *erfüllt* ist, wenn die freien Vorkommen der Variablen $x_1, \dots, x_s$ durch die Werte $a_1, \dots, a_s$ belegt werden. Formal heißt das, dass $(\mathfrak{A}, \beta) \models \varphi$, wobei β eine Belegung mit $\beta(x_i) = a_i$, für alle $i \in \{1, \dots, s\}$ ist.

1.10 Definition (Theorie, Modellklasse).
Sei L eine Logik und ψ ein L-Satz. Sei K eine Klasse von Strukturen und sei $\mathfrak{A} \in \mathsf{K}$.

(a) $\text{Th}_L(\mathfrak{A}) := \{\varphi \in L \mid \mathfrak{A} \models \varphi\}$ ist die *L-Theorie* der Struktur $\mathfrak{A}$.

(b) $\text{Mod}_{\mathsf{K}}(\psi) := \{\mathfrak{B} \in \mathsf{K} \mid \mathfrak{B} \models \psi\}$ ist die *Modellklasse* von ψ bezüglich K.