Vorlesung Logik und Komplexität

Sommersemester 2007

Nicole Schweikardt
Institut für Informatik
Humboldt-Universität zu Berlin

Dieses Skript basiert auf dem Skript von Stephan Kreutzer und Nicole Schweikardt zur Vorlesung Logik und Komplexität an der HU Berlin im Sommersemester 2005

Inhaltsverzeichnis

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 Artikel mit den Attributen Titel, Autor, Konferenz, Jahr.

0.1 Beispiele. In unserer Beispiel-Datenbank besteht die Relation 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. Beispielsweise ist 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:

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

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. Der Satz von Gaifman
5. Fixpunktlogiken

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. Kapitel 3 des Buchs [F].

[K] P. Kolaitis. On the expressive power of logics on finite models. Kapitel 2 des Buchs [F].

[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, 2007.

1 Grundlagen

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.

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

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

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

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.

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

- Unsere Bibliotheksdatenbank aus Beispiel 0.1 ist eine $\{\mathsf{Artikel}\}$-Struktur, wobei 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 $\mathbb{Z}_{\leqslant 0} := \{z \in \mathbb{Z} : z \leqslant 0\}$, $<^{\mathfrak{A}} := \{(a,b) \in \mathbb{Z}_{\leqslant 0}^2 : a > b\}$, $0^{\mathfrak{A}} := 0^{\mathfrak{N}}$ 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: $h(R^{\mathfrak{A}}) = \{\, (h(a_1), .., h(a_k)) \mid (a_1, .., a_k) \in R^{\mathfrak{A}} \,\}$, wobei $k := ar(R)$.

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

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

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

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

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

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

(Q1) Ist ψ eine Formel in $\text{FO}[\sigma]$ und $x \in \text{Var}_1$, so gehören auch folgende Formeln zu $\text{FO}[\sigma]$:
- $\exists x\, \psi$ (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 $\text{FO}[\sigma]$-Formel. Die *Modellbeziehung* $\mathfrak{I} \models \varphi$ (in Worten: $\mathfrak{I}$ *erfüllt* φ bzw. $\mathfrak{I}$ ist *Modell* von φ) ist folgendermaßen per Induktion über den Formelaufbau definiert:

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

– $(\varphi_1 \to \varphi_2)$, so $\mathfrak{I} \models \varphi \iff$ (falls $\mathfrak{I} \models \varphi_1$, so auch $\mathfrak{I} \models \varphi_2$).
– $(\varphi_1 \leftrightarrow \varphi_2)$, so $\mathfrak{I} \models \varphi \iff$ ($\mathfrak{I} \models \varphi_1$ genau dann, wenn $\mathfrak{I} \models \varphi_2$).

- Falls φ gemäß Regel (Q1) gebildet ist und φ von der Form
  – $\exists x\, \psi$, so $\mathfrak{I} \models \varphi \iff$ es gibt ein $a \in A$, so dass $(\mathfrak{A}, \beta\frac{a}{x}) \models \psi$.
  – $\forall x\, \psi$, so $\mathfrak{I} \models \varphi \iff$ 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*$(\varphi) := \{v_1, \ldots, v_k\} \cap \text{Var}_1$.
- 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

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

- 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 $x, y, z, u, v, x_1, x_2, x_3, \ldots$ verwenden, um Variablen erster Stufe zu bezeichnen.

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

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

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

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

(d) $\varphi(\mathfrak{A}) := \{(a_1, \ldots, a_s) \in A^s : \mathfrak{A} \models \varphi[a_1, \ldots, a_s]\}$.

(e) Für eine Signatur $\sigma = \{R_1, \ldots, R_k, c_1, \ldots, c_\ell\}$ schreiben wir oft FO$[R_1, \ldots, R_k, c_1, \ldots, c_\ell]$ an Stelle von FO$[\{R_1, \ldots, R_k, c_1, \ldots, c_\ell\}]$, um die Klasse aller FO$[\sigma]$-Formeln zu bezeichnen.

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

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

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

1.11 Definition (Klassen von Strukturen)**.**

Mit $\left\{\begin{array}{l}\mathsf{All}\\ \mathsf{Fin}\\ \mathsf{Ord}\\ \mathsf{FinOrd}\end{array}\right\}$ bezeichnen wir die Klasse $\left\{\begin{array}{l}\textit{aller}\\ \text{aller } \textit{endlichen}\\ \text{aller } \textit{geordneten}\\ \text{aller } \textit{endlichen geordneten}\end{array}\right\}$ Strukturen.

Hierbei heißt eine Struktur $\mathfrak{A}$ *geordnet*, falls es in ihrer Signatur ein 2-stelliges Relationssymbol $<$ gibt, so dass $<^{\mathfrak{A}}$ eine *lineare Ordnung* auf dem Universum A ist, d.h. $<^{\mathfrak{A}}$ ist *transitiv*[2], *antisymmetrisch*[3] und *konnex*[4].

1.2 Einführung in die Komplexitätstheorie

In diesem Abschnitt wiederholen wir einige für diese Vorlesung wichtige Begriffe aus der Komplexitätstheorie. Für eine umfassende Einführung in die Komplexitätstheorie sei auf das Buch

> *Computational Complexity*
> von C. Papadimitriou, Addison-Wesley, 1994

verwiesen.

1.2.1 Turing-Maschinen und Berechenbarkeit

Turing-Maschinen sind von dem englischen Mathematiker Alan Turing (1912–1954) eingeführt worden.

Turing-Maschinen intuitiv:

Das hier benutzte Modell von Turing-Maschinen ist eine 1-Band Turing-Maschine mit linksseitig begrenztem Arbeitsband, bei dem die Eingabe zu Beginn auf dem Arbeitsband gespeichert ist. Es gibt Anweisungen der Form

$$(q, a, q', b, 1),$$

die folgendes besagen: Wenn die Maschine im Zustand q ist und ihr Schreib-/Lesekopf das Symbol a liest, dann kann sie in den Zustand q' wechseln, das gelesene Symbol a durch das Symbol b ersetzen und anschließend den Schreib-/Lesekopf eine Position nach rechts bewegen.
Die Maschine *hält an*, wenn sie einen Endzustand erreicht.

[2] D.h. für alle $a, b, c \in A$ gilt: Falls $a <^{\mathfrak{A}} b$ und $b <^{\mathfrak{A}} c$, so auch $a <^{\mathfrak{A}} c$.
[3] D.h. für alle $a, b \in A$ gilt: Falls $(a, b) \in <^{\mathfrak{A}}$, so $(b, a) \notin <^{\mathfrak{A}}$.
[4] D.h. für alle $a, b \in A$ gilt: $a <^{\mathfrak{A}} b$ oder $b <^{\mathfrak{A}} a$ oder $b = a$.

Formale Definition von Turing-Maschinen:

1.12 Definition (Turing-Maschine)**.**
Eine *nichtdeterministische Turing-Maschine* (kurz: NTM)

$$M \;=\; (Q, \Sigma, \Gamma, \Delta, q_0, F),$$

besteht aus

- einer endlichen Menge Q von Zuständen,
- einem endlichen Arbeitsalphabet Γ mit ausgezeichnetem *Blank-Symbol* $\square$,
- einem Eingabealphabet $\Sigma \subseteq \Gamma \setminus \{\square\}$,
- einem Anfangszustand q_0,
- einer Menge $F = F_{\text{akz}} \dot\cup F_{\text{verw}} \subseteq Q$ von Endzuständen, die aus einer Menge F_{akz} von *akzeptierenden* und einer Menge F_{verw} von *verwerfenden* Endzuständen besteht,
- einer Übergangsrelation $\Delta \subseteq (Q \setminus F) \times \Gamma \times Q \times \Gamma \times \{-1, 0, 1\}$.

M heißt *deterministisch* (kurz: M ist eine DTM), falls für alle $q \in Q \setminus F$ und $a \in \Gamma$ genau ein $q' \in Q$, $a' \in \Gamma$ und $m \in \{-1, 0, 1\}$ mit $(q, a, q', a', m) \in \Delta$ existiert. In diesem Fall schreiben wir oft

$$M \;=\; (Q, \Sigma, \Gamma, \delta, q_0, F),$$

mit *Überführungsfunktion* $\delta : (Q \setminus F) \times \Gamma \;\to\; Q \times \Gamma \times \{-1, 0, 1\}$.

1.13 Definition (Konfigurationen einer TM)**.**

(a) Eine *Konfiguration* einer Turing-Maschine $M \;=\; (Q, \Sigma, \Gamma, \Delta, q_0, F)$ ist ein Tripel

$$c \;=\; (q, p, u) \;\in\; Q \times \mathbb{N} \times \Gamma^*$$

mit $p \leqslant |u|$. D.h. eine Konfiguration ist eine vollständige Beschreibung aller relevanten Daten über einen bestimmten Zeitpunkt während einer Berechnung. Die Konfiguration $c = (q, p, u)$ gibt an, dass die Maschine sich im Zustand q befindet, der Kopf an Position p steht und die Inschrift des Arbeitsbandes gerade u ist.

Beachte: u ist ein Wort *endlicher* Länge. Die unendlich vielen $\square$-Symbole weiter rechts auf dem Arbeitsband werden weggelassen.

(b) Wir bezeichnen die Menge aller möglichen Konfigurationen von M mit $\mathcal{C}_M$.

(c) Die *Startkonfiguration* von M bei Eingabe w ist $C_0(w) := (q_0, 0, w\square)$.

(d) Eine Konfiguration $C = (q, p, u)$ heißt *Endkonfiguration*, falls $q \in F$. Sie heißt *akzeptierend*, falls $q \in F_{\text{akz}}$ und *verwerfend*, falls $q \in F_{\text{verw}}$.

1.14 Definition (Lauf einer TM)**.** Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine Turing-Maschine.

(a) Die Übergangsrelation Δ induziert eine (partielle) Funktion

$$\text{Next}_M : \mathcal{C}_M \rightarrow \text{Pot}(\mathcal{C}_M),$$

wobei für $C = (q, p, u) \in \mathcal{C}_M$ gilt:[5]

$$\text{Next}_M(C) := \left\{ (q', p', u') \;\middle|\; \begin{array}{l} \text{es gibt } a, b \in \Gamma \text{ und } m \in \{-1, 0, 1\}, \text{ so dass} \\ (q, a, q', b, m) \in \Delta, p' = p+m, u_p = a, u'_p = b, \\ u'_i = u_i \text{ für alle } i \neq p, \text{ und } |u'| = |u|, \text{ falls} \\ p' < |u| \text{ bzw. } |u'| = |u| + 1 \text{ und } u'_{p'} = \square, \text{ falls} \\ p' = |u|. \end{array} \right\}.$$

$\text{Next}_M(C)$ enthält also alle Konfigurationen C', so dass M von Konfiguration C in einem Schritt in die Konfiguration C' übergehen kann.

Bemerkung: Ist M deterministisch, so enthält $\text{Next}_M(C)$ höchstens eine Konfiguration, d.h. die Maschine geht von einem Berechnungszustand in einen eindeutig bestimmten nächsten Berechnungszustand über.

(b) Auf Eingabe w definiert die Turing-Maschine M einen *Berechnungsbaum* $B_M(w)$ wie folgt: Der Baum hat einen mit $C_0(w)$ beschrifteten Wurzelknoten. Ist v ein mit C beschrifteter Knoten im Baum, so hat er für jedes $C' \in \text{Next}_M(C)$ einen Kindknoten mit Beschriftung C'.

(c) Ein *Lauf* der TM M auf Eingabe w ist ein Pfad im Berechnungsbaum, der an der Wurzel beginnt und entweder unendlich lang ist oder in einer Endkonfiguration endet (der Lauf heißt dann *endlich* bzw. *terminierend*).

Bemerkung: Ein Lauf entspricht also einer möglichen Berechnung von M auf Eingabe w. Wenn die TM nicht deterministisch ist, kann sie auf der gleichen Eingabe verschiedene Berechnungen ausführen.

(d) Ein Lauf heißt *akzeptierend* (*verwerfend*), falls er in einer akzeptierenden (verwerfenden) Endkonfiguration endet.

1.15 Definition (Sprache einer TM)**.**

(a) Eine Turing-Maschine M *akzeptiert* eine Eingabe w, falls es (mindestens) einen akzeptierenden Lauf von M auf w gibt.
M *verwirft* eine Eingabe w, falls alle terminierenden Läufe von M auf w verwerfen.

(b) Die Sprache $L(M) := \{w \in \Sigma^* : M \text{ akzeptiert } w\}$ heißt die von M *akzeptierte* Sprache.

[5] Hierbei ist $u = u_0 \cdots u_\ell$ mit $u_i \in \Gamma$. Mit $|u|$ bezeichnen wir die *Länge* des Worts u, also $|u| = \ell + 1$.

(c) Eine Sprache $L \subseteq \Sigma^*$ heißt *semi-entscheidbar*, falls es eine Turing-Maschine M gibt, so dass $L = L(M)$.

(d) Eine Sprache $L \subseteq \Sigma^*$ heißt *entscheidbar*, falls es eine Turing-Maschine M gibt, so dass $L = L(M)$ und jeder Lauf von M auf jeder Eingabe w terminiert.

1.16 Bemerkungen.

(a) Man beachte die Asymmetrie in der Definition des *Akzeptierens* einer Eingabe w: Zum Akzeptieren eines Worts w muss es nur (mindestens) einen akzeptierenden Lauf geben. Zum Verwerfen des Worts w müssen hingegen alle Läufe bei Eingeabe w verwerfen oder nicht-terminierend sein.

(b) Ist M *deterministisch*, so besteht der Berechnungsbaum nur aus einem Pfad (d.h. jeder Knoten hat höchstens einen Nachfolger). Die Berechnung von M auf Eingabe w ist also eindeutig.

Es ist bekannt, dass deterministische und nichtdeterministische Turing-Maschinen genau dieselben Sprachen entscheiden können:

1.17 Satz.
Jede durch eine nichtdeterministische Turing-Maschine (semi-)entscheidbare Sprache L ist auch durch eine deterministische Turing-Maschine (semi-)entscheidbar.

(Hier ohne Beweis.)

Das Halteproblem für Turing-Maschinen:

Man kann Turing-Maschinen M und Eingaben w als Wörter $\rho(M)$ und $\rho(w)$ über einem festen, von der TM unabhängigen, Alphabet $\mathscr{A}$ so kodieren, dass sich die Berechnung von M auf w effektiv aus $\rho(M)$ und $\rho(w)$ rekonstruieren lässt (die genaue Wahl der Kodierung ist für diese Vorlesung nicht wichtig und wird deshalb hier weggelassen). Unter Verwendung einer solchen Kodierung kann man Eingaben $\rho(M)\#\rho(w)$ über dem Alphabet $\mathscr{A} \dot\cup \{\#\}$ betrachten und eine Turing-Maschine U bauen, die bei Eingabe $\rho(M)\#\rho(w)$ die Berechnung von M auf w simuliert. Eine solche Maschine U heißt *universelle Turing-Maschine*.

1.18 Definition (Halteproblem)**.** Das *Halteproblem* ist die Sprache

$$H := \{\, \rho(M)\#\rho(w) \,:\, M \text{ ist eine DTM, } w \text{ eine Eingabe für } M,\ M \text{ terminiert auf } w \,\}.$$

Das *Halteproblem auf leerem Eingabewort* ist die Sprache

$$H_\varepsilon := \{\, \rho(M) \,:\, M \text{ ist eine DTM, } M \text{ terminiert bei Eingabe } \varepsilon \,\}.$$

Dabei bezeichnet ε das *leere Wort*.

1.19 Satz.
Das Halteproblem H für Turing-Maschinen und das Halteproblem H_ε auf leerem Eingabewort ist nicht entscheidbar (aber semi-entscheidbar).

(Hier ohne Beweis.)

1.2.2 Komplexitätsklassen

Neben der Frage nach dem absolut Berechenbaren interessieren wir uns für Berechenbarkeit innerhalb einer gegebenen Menge von Ressourcen. Die wichtigsten Ressorcen, deren Verfügbarkeit oft beschränkt ist, sind *Zeit* und *Platz*.

1.20 Definition.
Seien $T, S : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ monoton wachsende Funktionen, und sei M eine TM.

(a) M heißt *T-zeitbeschränkt*, falls jede Berechnung von M auf Eingaben der Länge n höchstens $T(n)$ Schritte macht. D.h. für alle $n \in \mathbb{N}$ und alle Eingabeworte w der Länge n gilt: Jeder Pfad im Berechnungsbaum $B_M(w)$ hat die Länge $\leqslant T(n)$.

(b) M heißt *S-platzbeschränkt*, falls jede Berechnung von M auf Eingaben der Länge n höchstens $S(n)$ Speicherstellen benutzt. D.h. für alle $n \in \mathbb{N}$, alle Eingabeworte w der Länge n und jeden Knoten v im Berechnungsbaum $B_M(w)$ gilt: Ist (q, p, u) die Konfiguration, mit der v beschriftet ist, so hat das Wort u die Länge $|u| \leqslant S(n)$.

1.21 Definition (Komplexitätsklassen)**.**
Seien $S, T : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ monoton wachsende Funktionen.

(a) $\textsc{Dtime}(T)$ (bzw. $\textsc{Dspace}(S)$) ist die Klasse aller Sprachen L, für die es eine *deterministische* Turing-Maschine M gibt, die T-zeitbeschränkt (bzw. S-platzbeschränkt) ist und für die gilt: $L = L(M)$.

(b) $\textsc{Ntime}(T)$ (bzw. $\textsc{Nspace}(S)$) ist die Klasse aller Sprachen L, für die es eine *nichtdeterministische* Turing-Maschine M gibt, die T-zeitbeschränkt (bzw. S-platzbeschränkt) ist und für die gilt: $L = L(M)$.

(c)
$$\begin{aligned}
\textsc{P} := \textsc{Ptime} &:= \bigcup_{k\in\mathbb{N}} \textsc{Dtime}(n^k), & \textsc{Pspace} &:= \bigcup_{k\in\mathbb{N}} \textsc{Dspace}(n^k),\\
\textsc{NP} &:= \bigcup_{k\in\mathbb{N}} \textsc{Ntime}(n^k), & \textsc{NPspace} &:= \bigcup_{k\in\mathbb{N}} \textsc{Nspace}(n^k),\\
\textsc{Exptime} &:= \bigcup_{k\in\mathbb{N}} \textsc{Dtime}(2^{(n^k)}), & \textsc{Expspace} &:= \bigcup_{k\in\mathbb{N}} \textsc{Dspace}(2^{(n^k)}),\\
\textsc{NExptime} &:= \textstyle\bigcup_{k\in\mathbb{N}} \textsc{Ntime}(2^{(n^k)}), & \textsc{NExpspace} &:= \textstyle\bigcup_{k\in\mathbb{N}} \textsc{Nspace}(2^{(n^k)}).
\end{aligned}$$

Beachte: Komplexitätsklassen sind keine Klassen von Turing-Maschinen sondern Klassen von Problemen (d.h. Sprachen, d.h. Mengen von Worten über einem Alphabet).

Zwei weitere wichtige Platzkomplexitätsklassen sind die Klassen $\textsc{Logspace}$ und $\textsc{NLogspace}$ aller Probleme, die durch Turing-Maschinen gelöst werden können, deren Platzverbrauch bei Eingaben der Länge n nur höchstens von der Größe $\mathcal{O}(\log n)$ ist. Da $\log n < n$ ist, heißt das, dass das Arbeitsband viel zu kurz ist um das ganze Eingabewort zu enthalten. Zur

Definition der Klassen LOGSPACE und NLOGSPACE werden daher Turing-Maschinen benutzt, die ein separates *Eingabeband* besitzen, dessen Kopf nur zum Lesen des Eingabeworts, nicht aber zum Schreiben genutzt werden kann. Neben diesem Eingabeband gibt es ein (herkömmliches) *Arbeitsband*, das zu Beginn der Berechnung leer ist. Für eine monoton wachsende Funktion $S : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ schreiben wir

$$\text{DSPACE}'(S) \quad \text{bzw.} \quad \text{NSPACE}'(S),$$

um die Klassen aller Sprachen L zu bezeichnen, für die es eine deterministische bzw. nichtdeterministische Turing-Maschine M mit separatem Eingabeband gibt (von dem gelesen, auf das aber nicht geschrieben werden kann), so dass bei Eingaben der Länge n auf dem *Arbeitsband* von M zu jedem Zeitpunkt der Berechnung höchstens $S(n)$ Speicherstellen benutzt werden.

1.22 Definition (LOGSPACE und NLOGSPACE)**.**

$$\begin{aligned} \text{LOGSPACE} &:= \bigcup_{k\in\mathbb{N}} \text{DSPACE}'(k \cdot \log n), \\ \text{NLOGSPACE} &:= \bigcup_{k\in\mathbb{N}} \text{NSPACE}'(k \cdot \log n). \end{aligned}$$

1.23 Proposition. *Seien $S, T : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ monoton wachsende Funktionen. Es gilt:*

$$\begin{array}{lcllcl} \text{DTIME}(T) & \subseteq & \text{NTIME}(T), & \text{DSPACE}(S) & \subseteq & \text{NSPACE}(S), \\ \text{DTIME}(T) & \subseteq & \text{DSPACE}(T), & \text{NTIME}(T) & \subseteq & \text{NSPACE}(T). \end{array}$$

Beweis: klar. □

1.24 Korollar. LOGSPACE $\subseteq$ NLOGSPACE *und* P $\subseteq$ NP $\subseteq$ NPSPACE**.**

1.25 Satz. *Für jede monoton wachsende Funktion $T : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ mit $T(n) \geqslant \log n$ gilt:*

$$\text{DTIME}(T) \subseteq \text{NTIME}(T) \subseteq \text{NSPACE}(T) \subseteq \text{DTIME}(2^{\mathcal{O}(T)}).$$

Beweisidee: Die ersten beiden Inklusionen sind klar, siehe Proposition 1.23. Zum Beweis der dritten Inklusion sei M eine T-platzbeschränkte NTM. Jede Konfiguration der Turing-Maschine, die während eines Laufs von M bei einer Eingabe der Länge n auftritt, kann durch ein Wort der Länge $\mathcal{O}(T(n))$ kodiert werden. Insbesondere gibt es höchstens $2^{\mathcal{O}(T(n))}$ viele solche Konfigurationen. Die (nichtdeterministische) Maschine M kann daher durch eine *deterministische* Turing-Maschine M' simuliert werden, die bei Eingabe eines Worts w auf ihrem Arbeitsband eine Liste aller Konfigurationen erzeugt, die M bei Eingabe w erreichen kann. □

1.26 Korollar. NLOGSPACE $\subseteq$ P *und* NPSPACE $\subseteq$ EXPTIME**.**

Frage: Welche Inklusionen sind strikt? – Kann man z.B. mit exponentieller Zeit *mehr* Probleme lösen als mit polynomieller Zeit?

Hierarchiesätze:

1.27 Definition (zeit- bzw. platzkonstruierbare Funktionen)**.**
Seien $S, T : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ monoton wachsende Funktionen.

(a) T heißt *zeitkonstruierbar*, falls es eine DTM gibt, die auf Eingaben der Länge $n \in \mathbb{N}$ genau $T(n)$ Schritte macht und dann hält. (Dabei ist egal, was M berechnet.)

(b) S heißt *platzkonstruierbar*, falls es eine DTM gibt, die auf Eingaben der Länge $n \in \mathbb{N}$ irgendwann terminiert und im Laufe der Berechnung auf ihrem Arbeitsband genau $S(n)$ Speicherzellen benutzt.

Bemerkung: Zeit- bzw. platzkonstruierbare Funktionen sind in dem Sinne "schön", dass ihre Komplexität nicht höher ist als ihre Werte. Die meisten üblicherweise verwendeten Funktionen, beispielsweise jedes Polynom sowie die Funktionen $n!$, 2^n und $2^{(n^k)}$, sind zeit- und platzkonstruierbar. Die Funktion $k \cdot \log n$ ist platzkonstruierbar (aber nicht zeitkonstruierbar).

1.28 Theorem (Zeithierarchiesatz)**.** *Seien $T, t : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ monoton wachsende Funktionen mit $t \cdot \log t = o(T)$, T zeitkonstruierbar und $T(n) \geqslant n$ für alle $n \in \mathbb{N}$. Dann gilt:*

$$\textsc{Dtime}(t) \subsetneqq \textsc{Dtime}(T).$$

(Hier ohne Beweis.)

1.29 Korollar. $\textsc{P} \subsetneqq \textsc{Exptime}$.

Analog zum Zeithierarchiesatz gibt es auch einen Platzhierarchiesatz:

1.30 Theorem (Platzhierarchiesatz)**.** *Seien $S, s : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ monoton wachsende Funktionen mit $s = o(S)$, S platzkonstruierbar und $S(n) \geqslant \log n$ für alle $n \in \mathbb{N}$. Dann gilt:*

$$\textsc{Dspace}(s) \subsetneqq \textsc{Dspace}(S).$$

(Hier ohne Beweis.)

1.31 Korollar. $\textsc{Logspace} \subsetneqq \textsc{Pspace} \subsetneqq \textsc{Expspace}$.

Komplementabschluss nichtdeterministischer Komplexitätsklassen:

Frage: Wenn $L \subseteq \Sigma^*$ eine Sprache aus $\textsc{Ntime}(T)$ oder $\textsc{Nspace}(S)$ ist, welche Ressourcen benötigt man, um die Sprache $\overline{L} := \Sigma^* \setminus L$ (d.h. das Komplement von L) zu entscheiden?

1.32 Definition. Sei $\mathscr{K}$ eine Komplexitätsklasse. Wir definieren die Komplementklasse

$$\text{co-}\mathscr{K} := \{\overline{L} : L \in \mathscr{K}\}$$

als die Klasse aller Sprachen, deren Komplement in $\mathscr{K}$ liegt.

Beispielsweise besteht die Klasse co-NP aus allen Sprachen, deren Komplement in NP liegt.

Klar ist, dass *deterministische* Komplexitätsklassen unter Komplementbildung abgeschlossen sind – dazu vertauscht man einfach akzeptierende und verwerfende Endzustände einer DTM. Es gilt also insbesondere P = co-P, PSPACE = co-PSPACE und LOGSPACE = co-LOGSPACE.
Bloßes Vertauschen von akzeptierenden und verwerfenden Endzuständen liefert bei *nichtdeterministischen* Turing-Maschinen in der Regel nicht das gewünschte Ergebnis. In der Tat ist die Frage, ob *nichtdeterministische Zeitkomplexitätsklassen* unter Komplementbildung abgeschlossen sind, ein offenes Forschungsproblem. Man vermutet, dass

$$\text{co-NP} \neq \text{NP}$$

gilt (woraus insbesondere P $\neq$ NP folgen würde), kann dies bisher aber nicht beweisen.

Erstaunlicherweise konnte das Problem für nichtdeterministische *Platz*klassen gelöst werden:

1.33 Theorem (Satz von Immerman und Szelepcsényi)**.**
Sei $S : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ platzkonstruierbar und $S(n) \geqslant \log n$ für alle $n \in \mathbb{N}$. Dann gilt:

$$\text{NSPACE}(S) = \text{co-NSPACE}(S).$$

(Hier ohne Beweis.)

1.34 Korollar. NLOGSPACE = co-NLOGSPACE.

Frage: Wie verhalten sich nichtdeterministische zu deterministischen Komplexitätsklassen? Gilt zum Beispiel

$$\text{P} = \text{NP}\,?$$

Wiederum ist das Problem für Zeitklassen offen, während es für Platzklassen weitgehend gelöst ist:

1.35 Theorem (Satz von Savitch)**.**
Sei $S : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ platzkonstruierbar und $S(n) \geqslant \log n$ für alle $n \in \mathbb{N}$. Dann gilt:

$$\text{NSPACE}(S) \subseteq \text{DSPACE}(S^2).$$

(Hier ohne Beweis.)

1.36 Korollar. PSPACE = NPSPACE *und* EXPSPACE = NEXPSPACE.

Offen bleibt weiterhin, ob

$$\text{LOGSPACE} = \text{NLOGSPACE}\,?$$

Insgesamt erhält man die in Abbildung 1.1 dargestellte Inklusionsstruktur der Komplexitätsklassen.

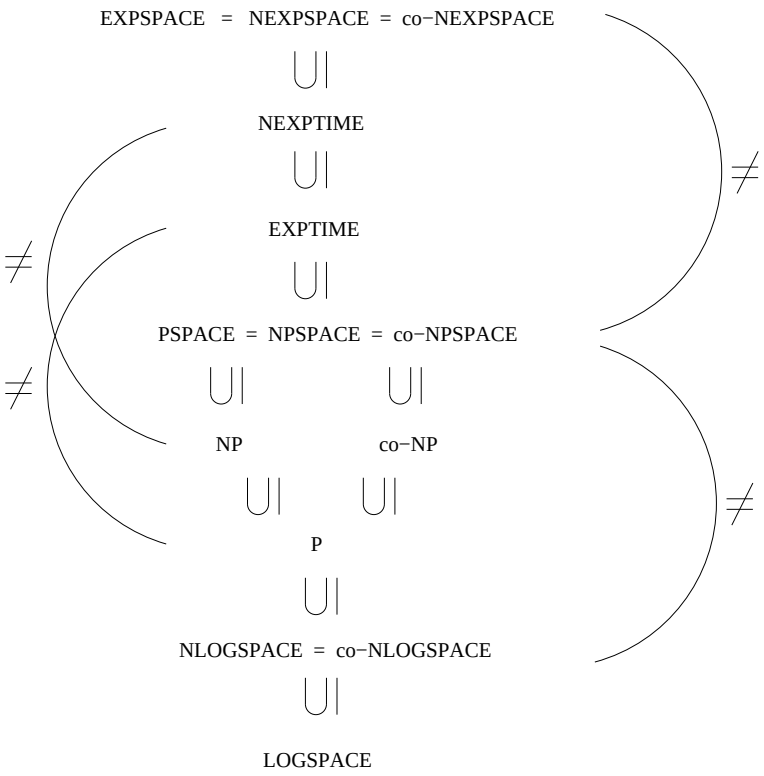


Abbildung 1.1: Inklusionsstruktur der Komplexitätsklassen.

1.2.3 Reduktionen und NP-Vollständigkeit

1.37 Definition (Polynomialzeit-Reduktionen)**.**
Seien Σ_1 und Σ_2 endliche Alphabete, und sei $A \subseteq \Sigma_1^*$ und $B \subseteq \Sigma_2^*$.
Eine *Polynomialzeit-Reduktion* (kurz: Reduktion) von A auf B ist eine (deterministisch) in polynomieller Zeit berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$, so dass für alle $w \in \Sigma_1^*$ gilt:

$$w \in A \iff f(w) \in B.$$

Existiert eine Polynomialzeitreduktion f von A auf B, so sagen wir: A *ist polynomiell reduzierbar auf* B (kurz: $A \leqslant_p B$ bzw. genauer $f : A \leqslant_p B$).

Bemerkung: Der Begriff *Polynomialzeit-Reduktion* ist so definiert, dass folgendes gilt: Falls $A \leqslant_p B$ und $B \in$ PTIME, so auch $A \in$ PTIME. Umgekehrt heißt das: Wenn $A \leqslant_p B$ und $A \notin$ PTIME, so auch $B \notin$ PTIME.
Anschaulich bedeutet $A \leqslant_p B$, dass A "höchstens so schwer" wie B ist.

1.38 Definition (Vollständigkeit)**.** Sei $\mathscr{K}$ eine Komplexitätsklasse und sei $B \subseteq \Sigma^*$.

(a) Das Problem B heißt $\mathscr{K}$*-hart*, falls für alle Probleme $A \in \mathscr{K}$ gilt: $A \leqslant_p B$.

(b) B heißt $\mathscr{K}$*-vollständig*, falls $B \in \mathscr{L}$ und B $\mathscr{K}$-hart ist.

In einem gewissen Sinn sind die vollständigen Probleme die "schwersten" Probleme einer Komplexitätsklasse. Insbesondere gilt für jedes NP-*vollständige* Problem B: Falls B in P liegt, so liegt *jedes* Problem aus NP bereits in P, d.h. es gilt NP = P.
Beispiele für NP-vollständige Probleme sind:

SAT (aussagenlogisches Erfüllbarkeitsproblem)
Eingabe: Eine aussagenlogische Formel α.
Frage: Gibt es eine Variablenbelegung, die α erfüllt?

TSP (Travelling Salesman Problem)
Eingabe: Eine Distanzmatrix D von Städten und eine Distanz k.
Frage: Gibt es eine Rundreise, die jede Stadt genau einmal besucht und bei der insgesamt höchstens die Distanz k zurückgelegt wird?

CLIQUE
Eingabe: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$.
Frage: Gibt es in G eine Clique[6] der Größe k?

[6] d.h. eine Menge $V' \subseteq V$ so dass für alle $u, v \in V'$ mit $u \neq v$ gilt: $E(u, v)$

Vollständigkeit in Komplexitätsklassen unterhalb von P:

Für Komplexitätsklassen $\mathscr{K} \subseteq$ P liefern *Polynomialzeit*-Reduktionen keinen sinnvollen Vollständigkeitsbegriff, denn jedes Problem in $\mathscr{K}$ (außer den "trivialen" Problemen $\varnothing$ und Σ^*) ist vollständig (bzgl. Polynomialzeit-Reduktionen) für $\mathscr{K}$.
Um einen sinnvollen Vollständigkeitsbegriff für Klassen $\mathscr{K}$ mit LOGSPACE $\subset \mathscr{K} \subseteq$ P zu erhalten, werden oft die *Logspace*-Reduktionen verwendet, die analog zu den Polynomialzeit-Reduktionen definiert sind, wobei die Funktion f auf logarithmischem Platz berechenbar sein musss. Ein Beispiel für ein solchermaßen NLOGSPACE-vollständiges Problem ist das bereits in Kapitel 0 (Einleitung) betrachtete Problem ERREICHBARKEIT.

Eine andere Charakterisierung der Klasse NP:

Oft werden nichtdeterministische Polynomialzeit-Algorithmen so beschrieben, dass der Algorithmus eine Lösung "rät" und diese dann überprüft. Beispielsweise kann ein Algorithmus, der TSP löst, eine Rundreise erraten und dann leicht ausrechnen, ob deren Gesamtlänge auch wirklich $\leqslant k$ ist.
Die folgende Charakterisierung von NP macht diesen Ansatz explizit:

1.39 Satz. *Ein Problem $L \subseteq \Sigma^*$ ist genau dann in* NP*, wenn es ein Polynom $p(n)$ und ein Problem $L' \in$ P gibt, so dass*

$$L = \{w \in \Sigma^* : \textit{es gibt ein Wort } z\textit{, so dass } |z| \leqslant p(|w|) \textit{ und } w\#z \in L'\}.$$

Das Wort z wird oft *polynomieller Zeuge* genannt.

Beweis: Ein direkter Beweis über Turing-Maschinen ist sehr leicht.
Der Satz folgt aber auch aus dem *Satz von Fagin*, der in Kapitel 2 (Deskriptive Komplexität) bewiesen wird. □

1.2.4 Orakel-Turing-Maschinen und die Polynomialzeit-Hierarchie

Die *Polynomialzeit-Hierarchie* (auch: *Polynomielle Hierarchie* bzw. *Polynomiale Hierarchie*) ist wegen ihres engen Zusammenhangs zur Logik zweiter Stufe für diese Vorlesung von Interesse; siehe Kapitel 2 (Deskriptive Komplexität). Ähnlich wie bei der Klasse NP gibt es viele äquivalente Charakterisierungen der Polynomialzeit-Hierarchie, u.a. die folgende, die auf der Charakterisierung von NP aus Satz 1.39 aufbaut.

1.40 Definition (Polynomialzeit-Hierarchie)**.** Für jedes $k \in \mathbb{N}$ definieren wir induktiv die Komplexitätsklassen Σ_k^p und Π_k^p folgendermaßen:

- $\Sigma_0^p := \Pi_0^p :=$ P.
- Ein Problem $L \subseteq \Sigma^*$ gehört genau dann zur Klasse Σ_{k+1}^p, wenn es ein Polynom $p(n)$ und ein Problem $L' \in \Pi_k^p$ gibt, so dass

$$L = \{w \in \Sigma^* : \text{es gibt ein Wort } z\text{, so dass } |z| \leqslant p(|w|) \text{ und } w\#z \in L'\}.$$

- Ein Problem $L \subseteq \Sigma^*$ gehört genau dann zur Klasse Π^p_{k+1}, wenn es ein Polynom $p(n)$ und ein Problem $L' \in \Sigma^p_k$ gibt, so dass

$$L = \{w \in \Sigma^* : \text{für alle Worte } z \text{ mit } |z| \leqslant p(|w|) \text{ gilt } w\#z \in L'\}.$$

Des Weiteren setzen wir $\text{PH} := \bigcup_{k\in\mathbb{N}} \Sigma^p_k$.

1.41 Bemerkung. Man sieht leicht, dass folgendes gilt:

- $\Sigma^p_1 = \text{NP}$ und $\Pi^p_1 = \text{co-NP}$
- $\Pi^p_k = \text{co-}\Sigma^p_k$
- $\Sigma^p_k \subseteq \Pi^p_{k+1} \subseteq \Sigma^p_{k+2}$
- $\text{PH} = \bigcup_{k\in\mathbb{N}} \Pi^p_k$.

Um Turing-Maschinen basierte Charakterisierungen der Komplexitätsklassen der Polynomialzeit-Hierarchie einführen zu können, benötigen wir den Begriff der *Orakel-Turing-Maschinen*.

1.42 Definition (Orakel-TM)**.**
Sei Σ ein Alphabet und $B \subseteq \Sigma^*$ eine Sprache.
Eine *Orakel-Turing-Maschine M mit Orakel B* ist eine Turing-Maschine mit einem zusätzlichen Band, dem so genannten *Orakel-Band* und 3 zusätzlichen Zuständen q_{ja}, q_{nein} und $q_?$. Berechnungen und Konfigurationen von M sind wie üblich definiert, nur hat die Orakel-TM zusätzlich die Möglichkeit, Anfragen an das Orakel zu stellen. Dazu kann sie ein Wort w auf das Orakel-Band schreiben und dann in den Zustand $q_?$ gehen. Falls $w \in B$ liegt, so geht die Maschine direkt in den Zustand q_{ja} über; andernfalls in q_{nein}. In jedem der beiden Fälle wird der Inhalt des Orakel-Bands gelöscht.

Die Sprache B dient hier also als "Orakel". Die Entscheidung, ob $w \in B$ ist, ist "atomar", d.h. sie benötigt nur *einen* Schritt. Dies kann hilfreich sein, falls die Komplexität von B höher ist als die Maschine M selbst berechnen kann, z.B. wenn M deterministisch und polynomiell zeitbeschränkt ist und $B \in \text{NP}$.

Das Befragen eines "Orakels" kann man sich als Anfragen an einen Netzwerk-Server vorstellen: Das Programm auf einem Netzwerk-Client darf wegen der geringen Rechenkapazität des Clients nur polynomiell lange laufen, wohingegen der Server komplexere Aufgaben bewältigen kann. Der Client kann daher in seiner Berechnung Anfragen an den Server stellen, den Server also als "Orakel" benutzen.

Orakel-Turing-Maschinen können benutzt werden, um neue Komplexitätsklassen einzuführen:

1.43 Definition. Sei $B \subseteq \Sigma^*$.

(a) Sei M eine Orakel-TM mit Orakel B. Die von M mit Orakel B akzeptierte Sprache ist

$$L(M^B) := \{w : M \text{ akzeptiert Eingabe } w \text{ mit Orakel } B\}.$$

Des Weiteren werden folgende Komplexitätsklassen definiert:

(b) $\mathrm{P}^B := \left\{ L \;\middle|\; \begin{array}{l}\text{es gibt eine polynomiell zeitbeschränkte deterministische}\\ \text{Orakel-TM } M \text{ mit Orakel } B, \text{ die } L \text{ entscheidet}\end{array} \right\}.$

(b) $\mathrm{NP}^B := \left\{ L \;\middle|\; \begin{array}{l}\text{es gibt eine polynomiell zeitbeschränkte nichtdeterministische}\\ \text{Orakel-TM } M \text{ mit Orakel } B, \text{ die } L \text{ entscheidet}\end{array} \right\}.$

(c) Für eine Komplexitätsklasse $\mathscr{K}$ ist

$$\mathrm{P}^{\mathscr{K}} := \bigcup_{B \in \mathscr{K}} \mathrm{P}^B \qquad \text{und} \qquad \mathrm{NP}^{\mathscr{K}} := \bigcup_{B \in \mathscr{K}} \mathrm{NP}^B.$$

1.44 Bemerkung. Offensichtlich gilt:

- $\mathrm{P}^{\mathrm{P}} = \mathrm{P}$, denn anstatt das Orakel zu befragen kann eine TM die Antwort des Orakels auch selbst berechnen.
- $\mathrm{P}^{\mathrm{NP}} = \mathrm{P}^{\text{co-NP}}$, $\mathrm{NP}^{\mathrm{NP}} = \mathrm{NP}^{\text{co-NP}}$, $\mathrm{NP} \subseteq \mathrm{P}^{\mathrm{NP}}$ und $\text{co-NP} \subseteq \mathrm{P}^{\mathrm{NP}}$.
- $\mathrm{P}^{\mathrm{NP}} = \mathrm{P}^B$ für jedes NP-vollständige Problem B.
  Insbesondere gilt also $\mathrm{P}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{SAT}} = \mathrm{P}^{\mathrm{TSP}} = \mathrm{P}^{\mathrm{CLIQUE}}$.
- $\mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{PSPACE}$.

Erweitert man P um NP-Orakel, so erhält man vermutlich echt mehr als NP und co-NP. Dies liegt daran, dass nichtdeterministische Zeitklassen wie NP vermutlich nicht unter Komplementbildung abgeschlossen sind (wegen der Asymmetrie in der Definition des Akzeptierens nichtdeterministischer Turing-Maschinen).

Sprachen, die durch Orakel-Maschinen definiert sind, können natürlich selbst wieder als Orakel verwendet werden. Dies führt zur folgenden Charakterisierung der Polynomialzeit-Hierarchie.

1.45 Proposition. *Für jedes $k \in \mathbb{N}$ gilt* $\Sigma^p_{k+1} = \mathrm{NP}^{\Pi^p_k} = \mathrm{NP}^{\Sigma^p_k}$.

Abgesehen von Σ^P_k und Π^P_k werden manchmal auch die folgenden Teilklassen von PH betrachtet.

1.46 Definition. $\Delta^p_0 := \mathrm{P}$, und für jedes $k \in \mathbb{N}$ setze $\Delta^p_{k+1} := \mathrm{P}^{\Sigma^p_k} \left(= \mathrm{P}^{\Pi^p_k} \right)$.

1.47 Bemerkung. Man sieht leicht, dass

$$\Delta_1^p = \text{P}, \quad \Sigma_1^p = \text{NP} \quad \text{und} \quad \Pi_1^p = \text{co-NP},$$

$$\Delta_k^p \subseteq \Sigma_k^p \subseteq \Delta_{k+1}^p \quad \text{und} \quad \Delta_k^p \subseteq \Pi_k^p \subseteq \Delta_{k+1}^p,$$

und

$$\text{PH} \subseteq \text{PSPACE}.$$

Insgesamt ergibt sich die in Abbildung 1.2 dargestellte Inklusionsstruktur der Klassen der Polynomiellen Hierarchie.

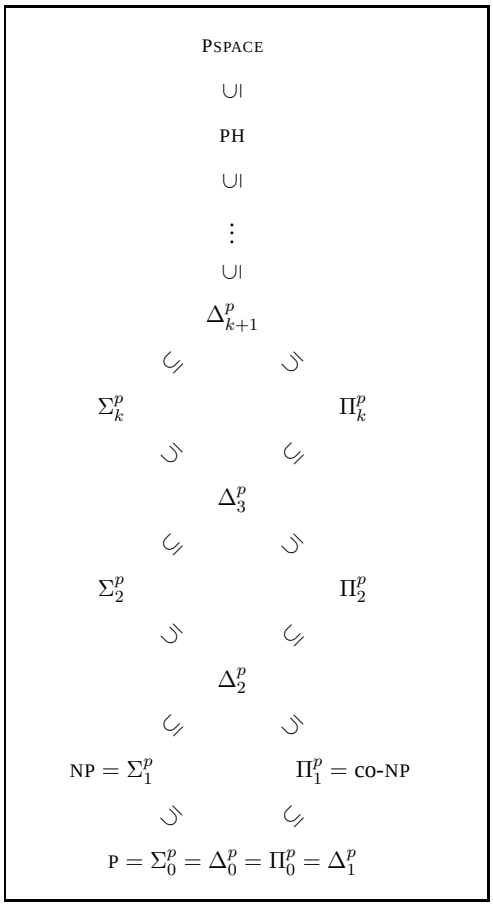


Abbildung 1.2: Inklusionsstruktur der Komplexitätsklassen der Polynomialzeit-Hierarchie.

1.3 Die Komplexität der Logik erster Stufe und der Satz von Trakhtenbrot

Wie in Kapitel 0 (Einleitung) dargestellt, sind die folgenden Logikprobleme von besonderem Interesse für die verschiedensten Bereiche der Informatik.

1.48 Definition (Logikprobleme)**.** Sei $\mathscr{L}$ eine Logik und K eine Klasse von Strukturen.

(a) ERFÜLLBARKEITSPROBLEM FÜR $\mathscr{L}$ IN K:
Eingabe: Ein Satz $\varphi \in \mathscr{L}$.
Frage: Gibt es eine Struktur $\mathfrak{A} \in \mathsf{K}$ mit $\mathfrak{A} \models \varphi$?

(b) AUSWERTUNGSPROBLEM FÜR $\mathscr{L}$ UND K:
Eingabe: Ein Satz $\varphi \in \mathscr{L}$ und eine Struktur $\mathfrak{A} \in \mathsf{K}$.
Frage: Gilt $\mathfrak{A} \models \varphi$?

(c) BERECHNUNGSPROBLEM FÜR $\mathscr{L}$ UND K:
Eingabe: Eine Formel $\varphi(x_1, .., x_r) \in \mathscr{L}$ und eine Struktur $\mathfrak{A} \in \mathsf{K}$.
Ziel: Berechne $\varphi(\mathfrak{A}) := \{(a_1, .., a_r) \in A^r : \mathfrak{A} \models \varphi[a_1, .., a_r]\}$.

(d) QUERY CONTAINMENT PROBLEM FÜR $\mathscr{L}$ IN K (auch: *Subsumptionsproblem*):
Eingabe: Zwei Formeln $\varphi_1(x_1, .., x_r)$ und $\varphi_2(x_1, .., x_r) \in \mathscr{L}$.
Frage: Gilt für alle $\mathfrak{A} \in \mathsf{K}$, dass $\varphi_1(\mathfrak{A}) \subseteq \varphi_2(\mathfrak{A})$?

In diesem Abschnitt untersuchen wir die Komplexität der obigen Probleme für die *Logik erster Stufe* FO und die Klasse Fin aller *endlichen Strukturen*.

1.3.1 Die Standardkodierung von Strukturen und Formeln

Komplexitätsanalysen der Komplexitätstheorie basieren auf Wortsprachen, d.h. Klassen von Wörtern über einem endlichen Alphabet. Logikprobleme hingegen sind für Formeln einer Logik und Klassen von Strukturen definiert. Um über die Komplexität von Logikproblemen sprechen zu können, müssen wir deshalb Strukturen und Formeln als *Wörter* über einem geeigneten Alphabet kodieren.

Formeln lassen sich ganz einfach als Wörter kodieren — jede Formel φ der Logik erster Stufe lässt sich beispielsweise direkt als Wort *enc*(φ) über dem (endlichen) Alphabet

$$\{\ \wedge,\ \vee,\ \neg,\ \rightarrow,\ \leftrightarrow,\ (,\ ),\ \exists,\ \forall,\ =,\ \text{var},\ R,\ c,\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\ \}$$

auffassen.

Um auch *endliche Strukturen* durch Wörter über einem endlichen Alphabet repräsentieren zu können, benötigen wir zunächst ein paar grundlegende Begriffe.
Auf endlichen linearen Ordnungen kann man die Elemente der Ordnung eindeutig mit

natürlichen Zahlen identifizieren, nämlich ihrem *Rang* (d.h. ihrer "Höhe") innerhalb der Ordnung:

1.49 Definition (Rang eines Elements bzgl. einer Ordnung)**.** Sei A eine endliche, durch eine Relation $<$ linear geordnete Menge. Für ein Element $a \in A$ definieren wir den *Rang* von a bzgl. $<$ als

$$rg_<(a) \;:=\; |\{b \in A \,:\, b < a\}|.$$

Falls $A = \{a_0 < a_1 < \cdots < a_{n-1}\}$, so ist $rg(a_i)$ also gerade die Zahl i.

1.50 Bemerkung. Alternativ lässt sich der Rang auch induktiv definieren als

$$rg_<(a) \;:=\; \begin{cases} 0 & \text{falls es kein } b \in A \text{ mit } b < a \text{ gibt} \\ \max\{rg(b) + 1 \,:\, b \in A \text{ mit } b < a\} & \text{sonst.} \end{cases}$$

Diese Definition ist auf *endlichen* Mengen A äquivalent zur obigen, funktioniert allerdings auch für bestimmte unendliche und partielle Ordnungen (so genannte *Wohlordnungen*).

1.51 Definition (Lexikographische Ordnung)**.** Sei $A := \{a_0 < a_1 < \cdots < a_{n-1}\}$ eine durch die Relation $<$ linear geordnete Menge, und sei r eine natürliche Zahl. Die *lexikographische Ordnung* $<_{lex}$ auf der Menge A^r aller r-Tupel über A ist wie folgt definiert:
Für $\vec{a} = (a_1, .., a_r) \in A^r$ und $\vec{b} = (b_1, .., b_r) \in A^r$ gilt

$$\vec{a} <_{lex} \vec{b} \quad :\Longleftrightarrow \quad \begin{array}{l} \text{es gibt ein } i \in \{1, .., r\}, \text{ so dass } a_i < b_i, \\ \text{und für alle } j < i \text{ gilt } a_j = b_j. \end{array}$$

Wie man leicht sieht, ist $<_{lex}$ eine *lineare Ordnung* auf A^r.

1.52 Beispiel. Sei $A = \{0, 1, 2\}$ und sei $<$ die natürliche lineare Ordnung auf A. Für die daraus resultierende lexikographische Ordnung $<_{lex}$ auf A^2 gilt:

$\vec{a} \in A^2$	$(0,0)$	$(0,1)$	$(0,2)$	$(1,0)$	$(1,1)$	$(1,2)$	$(2,0)$	$(2,1)$	$(2,2)$
$rg_{<_{lex}}(\vec{a})$	0	1	2	3	4	5	6	7	8

1.53 Definition (Die Standardkodierung $enc(\mathfrak{A})$ einer Struktur $\mathfrak{A}$)**.**
Sei $\sigma = \{R_1, .., R_k, c_1, .., c_\ell\}$ eine Signatur, wobei R_i ein Relationssymbol der Stelligkeit $r_i := ar(R_i)$ und c_i ein Konstantensymbol ist (für alle $i \leqslant k$ bzw. $i \leqslant \ell$).
Sei $r := \max\{r_1, .., r_k\}$ die maximale Stelligkeit eines Relationssymbols in σ.
Sei $\mathfrak{A}$ eine σ-Struktur mit einem Universum A der Mächtigkeit $n := |A|$.

Wir wählen zunächst eine beliebige lineare Ordnung $<$ auf der Menge A, es sei also $A = \{a_0, .., a_{n-1}\}$ mit $a_0 < a_1 < \cdots < a_{n-1}$. Im Folgenden werden wir jedes Element a_i mit seinem Rang $rg_<(a_i)$, also mit der Zahl i identifizieren. Des Weiteren identifizieren wir jedes r_i-Tupel $\vec{a} \in A^{r_i}$ mit seinem Rang $rg_{<_{lex}}(\vec{a}) \in \{0, 1, .., n^{r_i} - 1\}$.

Für jedes $i \in \{1, .., k\}$ kodieren wir die Relation $R_i^{\mathfrak{A}}$ durch das Wort

$$enc(R_i^{\mathfrak{A}}) \;:=\; w_0 w_1 \cdots w_{n^r - 1} \;\in\; \{0,1\}^{n^r},$$

wobei

- für alle $\vec{a} \in A^{r_i}$ gilt: $\vec{a} \in R_i^{\mathfrak{A}} \iff w_{rg_{<_{lex}}(\vec{a})} = 1$,
- für alle $j \geqslant n^{r_i}$ gilt: $w_j = 0$.

Konstanten $c_i^{\mathfrak{A}}$ werden analog als Wörter

$$enc(c_i^{\mathfrak{A}}) := w_0 w_1 \cdots w_{n^r-1} \in \{0,1\}^{n^r},$$

kodiert, wobei für alle $j \in \{0, \ldots, n^r - 1\}$ gilt:

$$w_j = 1 \iff j = rg_<(c_i^{\mathfrak{A}}).$$

Schließlich kodieren wir die Struktur $\mathfrak{A}$ durch das Wort

$$enc(\mathfrak{A}) \quad := \quad 1^n\, 0^{(n^r-n)}\ enc(R_1^{\mathfrak{A}})\ \cdots\ enc(R_k^{\mathfrak{A}})\ enc(c_1^{\mathfrak{A}})\ \cdots\ enc(c_\ell^{\mathfrak{A}}).$$

1.54 Beispiel. Sei $\sigma = \{R_1, c_1\}$ die Signatur, die aus einem Konstantensymbol c_1 und einem 2-stelligen Relationssymbol R_1 besteht. Wir betrachten die σ-Struktur $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, c_1^{\mathfrak{A}})$ mit $A = \{0,1,2\}$, $R_1^{\mathfrak{A}} = \{(0,1),(1,1)\}$ und $c_1^{\mathfrak{A}} = 2$. Dann gilt:

$$enc(\mathfrak{A}) = \underbrace{111000000}_{1^n 0^{n^2-n}}\ \underbrace{010010000}_{enc(R_1^{\mathfrak{A}})}\ \underbrace{001000000}_{enc(c_1^{\mathfrak{A}})}.$$

1.55 Bemerkung. Die Standardkodierung $enc(\mathfrak{A})$ hat folgende Eigenschaften:

(a) Die Kodierung hängt von der gewählten Ordnung des Universums ab.

(b) Wir können schnell auslesen, ob ein Tupel $\vec{a}$ in $R_i^{\mathfrak{A}}$ liegt. Dazu braucht man nur den Schreib-/Lesekopf einer Turing-Maschine in den richtigen Block zu bewegen und dort die $rg_{<_{lex}}(\vec{a})$-te Speicherzelle zu lesen. Insgesamt heißt das:

$$\vec{a} \in R_i^{\mathfrak{A}} \quad \iff \quad \text{auf Bandposition } i \cdot n^r + rg_{<_{lex}}(\vec{a}) \text{ steht eine } 1.$$

(c) Die Größe der Kodierung, d.h. die Länge des Worts $enc(\mathfrak{A})$, ist polynomiell in der Größe des Universums der Struktur $\mathfrak{A}$ (der Exponent hängt von der Signatur, nicht aber vom Universum A ab). Genauer: $|enc(\mathfrak{A})| = (1 + k + \ell) \cdot |A|^r$.

1.3.2 Datenkomplexität, Programmkomplexität, kombinierte Komplexität

1.56 Definition. Sei $\mathscr{L}$ eine Logik und $\mathscr{K}$ eine Komplexitätsklasse. Wir sagen:

(a) Die *kombinierte Komplexität des Auswertungsproblems für* $\mathscr{L}$ liegt in $\mathscr{K}$, falls gilt:

$$L := \{enc(\mathfrak{A}) \# enc(\varphi) : \mathfrak{A} \in \mathsf{Fin},\ \varphi \in \mathscr{L} \text{ und } \mathfrak{A} \models \varphi\} \in \mathscr{K}.$$

(Man beachte, dass L eine Menge von Worten über einem endlichen Alphabet ist.)

(b) Die *Datenkomplexität des Auswertungsproblems für* $\mathscr{L}$ liegt in $\mathscr{K}$, falls für alle Sätze $\varphi \in \mathscr{L}$ gilt:

$$L_\varphi \;:=\; \{enc(\mathfrak{A}) \,:\, \mathfrak{A} \in \mathsf{Fin} \text{ und } \mathfrak{A} \models \varphi\} \;\in\; \mathscr{K}.$$

(c) Die *Programmkomplexität des Auswertungsproblems für* $\mathscr{L}$ liegt in $\mathscr{K}$, falls für alle endlichen Strukturen $\mathfrak{A}$ gilt:

$$L_\mathfrak{A} \;:=\; \{enc(\varphi) \,:\, \varphi \in \mathscr{L} \text{ und } \mathfrak{A} \models \varphi\} \;\in\; \mathscr{K}.$$

1.57 Bemerkung. Bei der *Datenkomplexität* besteht die Eingabe nur aus (der Standardkodierung) einer Struktur $\mathfrak{A}$, während die betrachtete Formel *fest* ist. Die "Komplexität" wird also in der Größe der Struktur gemessen.
Der Name "Datenkomplexität" kommt aus dem Bereich der Datenbanken, in dem die Anfragen (also Formeln) typischerweise klein, die Datenbanken (also Strukturen) hingegen sehr groß sind. Daher interessiert man sich in diesem Kontext besonders dafür, wie der Ressourcenverbrauch mit der Größe der Datenbank wächst.

Oft ist die *Datenkomplexität* einer Logik viel kleiner als die kombinierte Komplexität, wohingegen kombinierte Komplexität und Programmkomplexität oft zusammenfallen. In dieser Vorlesung werden wir in erster Linie die Datenkomplexität und die kombinierte Komplexität betrachten.

1.58 Definition. Sei $\mathscr{L}$ eine Logik und $\mathscr{K}$ eine Komplexitätsklasse.

(a) Die *kombinierte Komplexität* vom Auswertungsproblem für $\mathscr{L}$ heißt $\mathscr{K}$*-vollständig*, falls die Sprache

$$L \;:=\; \{enc(\mathfrak{A})\#enc(\varphi) \,:\, \mathfrak{A} \in \mathsf{Fin},\ \varphi \in \mathscr{L} \text{ und } \mathfrak{A} \models \varphi\}$$

$\mathscr{K}$-vollständig ist.

(b) Die *Datenkomplexität* vom Auswertungsproblem für $\mathscr{L}$ heißt $\mathscr{K}$*-vollständig*, falls sie in $\mathscr{K}$ liegt und es (mindestens) einen Satz $\varphi \in \mathscr{L}$ gibt, so dass die Sprache

$$L_\varphi \;:=\; \{enc(\mathfrak{A}) \,:\, \mathfrak{A} \in \mathsf{Fin} \text{ und } \mathfrak{A} \models \varphi\}$$

$\mathscr{K}$-vollständig ist.

(c) Die *Programmkomplexität* vom Auswertungsproblem für $\mathscr{L}$ heißt $\mathscr{K}$*-vollständig*, falls sie in $\mathscr{K}$ liegt und es (mindestens) eine endliche Struktur $\mathfrak{A}$ gibt, so dass die Sprache

$$L_\mathfrak{A} \;:=\; \{enc(\varphi) \,:\, \varphi \in \mathscr{L} \text{ und } \mathfrak{A} \models \varphi\}$$

$\mathscr{K}$-vollständig ist.

1.3.3 Die Komplexität des Auswertungsproblems für FO

1.59 Lemma. *Es gibt eine deterministische Turing-Maschine, die das Problem*

$$L := \{\mathit{enc}(\mathfrak{A})\#\mathit{enc}(\varphi) : \mathfrak{A} \in \mathsf{Fin},\ \varphi \in \text{FO } \textit{und } \mathfrak{A} \models \varphi\}$$

löst und bei Eingabe eines Worts $\mathit{enc}(\mathfrak{A})\#\mathit{enc}(\varphi)$ *mit* $n := |\mathit{enc}(\mathfrak{A})|$ *und* $m := |\mathit{enc}(\varphi)|$ *nur Platz* $\mathcal{O}\big(m \cdot (\log m + m \cdot \log n)\big)$ *benutzt.*

Beweisidee: Man kann leicht einen rekursiven Algorithmus

$$\text{EVAL}(\mathfrak{A}, \varphi(x_1, .., x_s), a_1, .., a_s)$$

angeben, der bei Eingabe einer endlichen Struktur $\mathfrak{A}$, einer FO-Formel φ und Belegungen $a_1, .., a_s \in A$ für die freien Variablen von φ entscheidet, ob $\mathfrak{A} \models \varphi[a_1, .., a_s]$; siehe z.B.

> § *6.5 Auswertung von Formeln in endlichen Strukturen* in Prof. Grohes Skript zur Vorlesung *Theoretische Informatik I* vom Wintersemester 2004/05, erhältlich unter `http://www.informatik.hu-berlin.de/logik/lehre/WS04-05/ThI1/folien.html`

In jedem rekursiven Aufruf muss dieser Algorithmus sich höchstens folgendes merken:

1. Welche Teilformel ψ von φ wird gerade bearbeitet?

   … Dazu muss eine DTM sich nur merken, an der wievielten Position von $\mathit{enc}(\varphi)$ diese Teilformel beginnt; es reicht also ein Bitstring der Länge $\log m$.

2. Mit welchen Werten aus dem Universum von $\mathfrak{A}$ werden die freien Variablen von ψ gerade belegt?

   … Dazu muss eine DTM sich für jede der höchstens m freien Variablen einen Bitstring der Länge $\log n$ merken, der das jeweilige Element im Universum von $\mathfrak{A}$ repräsentiert.

Für jeden Rekursionsschritt wird also höchstens Platz $\mathcal{O}(\log m + m \cdot \log n)$ benötigt.
Da die Rekursionstiefe $\leqslant m$ ist, reicht insgesamt also Platz $\mathcal{O}\big(m \cdot (\log m + m \cdot \log n)\big)$ aus. □

Aus Lemma 1.59 ergibt sich direkt:

1.60 Satz (Die Komplexität des Auswertungsproblems für FO)**.**

(a) Die kombinierte Komplexität des Auswertungsproblems für FO *liegt in* PSPACE*.*

(b) Die Datenkomplexität des Auswertungsproblems für FO *liegt in* LOGSPACE*.*

Beweis: Sei M die DTM aus Lemma 1.59, die das Problem

$$L := \{\mathit{enc}(\mathfrak{A})\#\mathit{enc}(\varphi) : \mathfrak{A} \in \mathsf{Fin},\ \varphi \in \text{FO und } \mathfrak{A} \models \varphi\}$$

löst und bei Eingaben $\mathit{enc}(\mathfrak{A})\#\mathit{enc}(\varphi)$ mit $n := |\mathit{enc}(\mathfrak{A})|$ und $m := |\mathit{enc}(\varphi)|$ nur Platz $\mathcal{O}\big(m \cdot (\log m + m \cdot \log n)\big)$ benötigt. Wenn N die Länge der gesamten Eingabe bezeichnet, so ist $N = n + m + 1$ und die DTM M ist daher S-platzbeschränkt für eine Funktion $S : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ mit $S(N) \in$

$$\mathcal{O}\big(N \cdot (\log N + N \cdot \log N)\big) \;=\; \mathcal{O}\big(N \cdot \log N \;+\; N^2 \cdot \log N\big) \;=\; \mathcal{O}\big(N^2 \cdot \log N\big).$$

Somit ist M eine polynomiell platzbeschränkte Turing-Maschine, die das Problem L löst. Also: $L \in$ PSPACE, d.h. die kombinierte Komplexität des Auswertungsproblems für FO liegt in PSPACE. Somit ist *(a)* bewiesen.

Zum Beweis von *(b)* sei φ ein FO-Satz und

$$L_\varphi := \{\mathit{enc}(\mathfrak{A}) : \mathfrak{A} \in \mathsf{Fin} \text{ und } \mathfrak{A} \models \varphi\}.$$

Man kann leicht eine DTM M_φ bauen, die bei Eingabe $\mathit{enc}(\mathfrak{A})$ genau das tut, was M bei Eingabe $\mathit{enc}(\mathfrak{A})\#\mathit{enc}(\varphi)$ tun würde. M_φ löst dann das Problem L_φ, und da die Formel φ fest ist, ist $m = |\mathit{enc}(\varphi)|$ eine Konstante, d.h., M_φ hat Platzschranke $\mathcal{O}(\log n)$. Für jedes feste $\varphi \in$ FO ist also $L_\varphi \in$ LOGSPACE. Somit liegt die Datenkomplexität des Auswertungsproblems für FO in LOGSPACE. □

Die *Datenkomplexität* des Auswertungsproblems für FO ist tatsächlich noch geringer. Man kann nämlich zeigen, dass sie in einer Schaltkreiskomplexitätsklasse namens AC^0 liegt, von der man weiß, dass $\text{AC}^0 \subsetneqq$ LOGSPACE. (Wir werden das in dieser Vorlesung allerdings nicht beweisen.)
Hinsichtlich der *kombinierten Komplexität* ist ein PSPACE-Algorithmus aber das bestmögliche:

1.61 Satz.
Die kombinierte Komplexität des Auswertungsproblems für FO *ist* PSPACE*-vollständig.*

Beweis: Diesen Satz werden wir in Kapitel 2 (Deskriptive Komplexität), Satz 2.51 beweisen. □

1.3.4 Der Satz von Trakhtenbrot

Im vorherigen Abschnitt haben wir gesehen, dass das Auswertungsproblem für FO in PSPACE (kombinierte Komplexität) bzw. LOGSPACE (Datenkomplexität) lösbar ist. Nun werden wir uns dem *endlichen Erfüllbarkeitsproblem für* FO zuwenden, also dem Problem

ERFÜLLBARKEITSPROBLEM FÜR FO$[\sigma]$ IN Fin:
Eingabe: Ein FO$[\sigma]$-Satz φ.
Frage: Gibt es eine σ-Struktur $\mathfrak{A} \in \mathsf{Fin}$ mit $\mathfrak{A} \models \varphi$?

Formal wird dieses Problem durch die folgende Wortsprache kodiert:

$$L_{\text{ERF-FO}[\sigma]\text{-Fin}} := \left\{ \mathit{enc}(\varphi) \;\middle|\; \begin{array}{l} \varphi \text{ ist ein FO}[\sigma]\text{-Satz, für den es eine } \sigma\text{-Struktur} \\ \mathfrak{A} \in \mathsf{Fin} \text{ mit } \mathfrak{A} \models \varphi \text{ gibt} \end{array} \right\}.$$

1.62 Theorem (Satz von Trakhtenbrot, 1950)**.**
Es gibt eine endliche Signatur σ, so dass das Erfüllbarkeitsproblem für FO$[\sigma]$ *in* Fin *unentscheidbar ist.*

Beweis: Wir müssen zeigen, dass es keine Turing-Maschine geben kann, die das Problem $L_{\text{ERF-FO}[\sigma]\text{-Fin}}$ entscheidet, d.h., die bei Eingabe der Kodierung eines FO$[\sigma]$-Satzes entscheidet, ob dieser ein endliches Modell hat.

Idee: Wir wissen, dass das Halteproblem auf leerem Eingabewort, H_ε, unentscheidbar ist (siehe Satz 1.19). Wir kodieren nun die Berechnung einer beliebigen Turing-Maschine M (von der wir wissen wollen, ob sie bei leerer Eingabe anhält) durch einen FO-Satz φ_M, der genau dann ein *endliches* Modell hat, wenn die Berechnung von M bei leerem Eingabewort endlich ist, die TM also anhält. Wenn das endliche Erfüllbarkeitsproblem für FO entscheidbar wäre, wäre demnach auch das Halteproblem bei leerem Eingabewort entscheidbar, was im Widerspruch zu Satz 1.19 steht.

Diese Idee wird wie folgt ausgeführt:
Sei M eine DTM. Wir kodieren zunächst die Berechnung von M auf leerem Eingabewort durch eine Struktur $\mathfrak{A}_M$ über einer geeigneten Signatur σ. Anschließend konstruieren wir eine FO$[\sigma]$-Formel φ_M, die von genau denjenigen σ-Strukturen erfüllt wird, die die Struktur $\mathfrak{A}_M$ als Unterstruktur enthalten. Insbesondere gilt dann: φ_M hat genau dann ein endliches Modell, wenn $\mathfrak{A}_M$ endlich ist, d.h. wenn die Berechnung von M auf leerem Eingabewort endlich ist.

O.B.d.A. können wir annehmen, dass $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine DTM ist mit

- $Q = \{0, 1, \ldots, s_Q\}$ für ein $s_Q \in \mathbb{N}$, Anfangszustand $q_0 = 0$, $F \subseteq Q$,
- $\Gamma = \{0, 1, \ldots, s_\Gamma\}$ für ein $s_\Gamma \in \mathbb{N}$, Blank-Symbol $\square = 0$,
- $s := \max\{s_Q, s_\Gamma\}$,
- falls die Berechnung von M bei leerer Eingabe terminiert, so nach genau n Schritten, wobei n eine natürliche Zahl $\geqslant s$ ist.

Die Signatur σ ist (unabhängig von der konkreten DTM M) wie folgt gewählt:

$$\sigma := \{ <,\ \mathit{succ},\ 0,\ B,\ K,\ Z \},$$

wobei

- $<$ und *succ* zwei 2-stellige Relationssymbole, 0 ein Konstantensymbol,

- B ein 3-stelliges Relationssymbol,
- K und Z zwei 2-stellige Relationssymbole

sind. Wir geben FO$[\sigma]$-Formeln an, die in ihren Modellen $\mathfrak{A}$ folgende Interpretationen der Prädikate erzwingen:

(1.) $<^{\mathfrak{A}}$ ist eine lineare Ordnung, $0^{\mathfrak{A}}$ deren kleinstes Element und $\mathit{succ}^{\mathfrak{A}}$ ist die *Nachfolger*-Relation (engl: successor) bzgl. $<^{\mathfrak{A}}$, d.h.

$$(a,b) \in \mathit{succ}^{\mathfrak{A}} \iff a <^{\mathfrak{A}} b \text{ und es gibt kein } c \text{ mit } a <^{\mathfrak{A}} c <^{\mathfrak{A}} b.$$

(2.) $(t,p,\gamma) \in B^{\mathfrak{A}} \iff$ auf Bandposition p steht zum Zeitpunkt t der Berechnung von M bei leerem Eingabewort das Symbol γ.

(3.) $(t,p) \in K^{\mathfrak{A}} \iff$ der Schreib-/Lesekopf von M steht zum Zeitpunkt t (der Berechnung bei leerem Eingabewort) auf Bandposition p.

(4.) $(t,q) \in Z^{\mathfrak{A}} \iff$ M ist zum Zeitpunkt t (der Berechnung bei leerem Eingabewort) in Zustand q.

Die Struktur $\mathfrak{A}_M$, die die Berechnung von M bei leerer Eingabe kodiert, ist folgendermaßen definiert: Ihr Universum ist

$$A_M := \begin{cases} \{0,\ldots,n\}\,, & \text{falls die Berechnung nach genau } n \text{ Schritten terminiert} \\ \mathbb{N}\,, & \text{falls die Berechnung unendlich ist.} \end{cases}$$

Die Relationen $<^{\mathfrak{A}_M}$ und $\mathit{succ}^{\mathfrak{A}_M}$ sowie die Konstante $0^{\mathfrak{A}_M}$ sind durch die natürliche lineare Ordnung auf A_M, deren Nachfolger-Relation sowie die Zahl 0 belegt. Die Relationen $B^{\mathfrak{A}_M}$, $K^{\mathfrak{A}_M}$ und $Z^{\mathfrak{A}_M}$ sind genau so gewählt, wie in den Punkten (2.), (3.) und (4.) beschrieben.

Nun zur Definition der Formel φ_M, die erzwingen soll, dass ihre Modelle $\mathfrak{A}$ die Struktur $\mathfrak{A}_M$ als Substruktur enthalten:
Punkt (1.) wird durch folgende Formel gewährleistet:

$$\begin{aligned} \varphi_{<,\mathit{succ},0} \quad := \quad & \\ \forall x\,\forall y\,\forall z \quad & \big((x<y \;\wedge\; y<z) \;\rightarrow\; x<z\big) && \text{(transitiv)} \\ \wedge\; & \big(x<y \;\rightarrow\; \neg\, y<x\big) && \text{(antisymmetrisch)} \\ \wedge\; & \big(x<y \;\vee\; y<x \;\vee\; y=x\big) && \text{(konnex)} \\ \wedge\; & \big(0<x \;\vee\; 0=x\big) && \text{(0 ist kleinstes Elt.)} \\ \wedge\; & \big(\mathit{succ}(x,y) \;\leftrightarrow\; (x<y \;\wedge\; \neg\exists u\,(x<u \wedge u<y))\big) && (\mathit{succ} \text{ ist Nachf.-Rel.}) \end{aligned}$$

In jedem Modell $\mathfrak{A}$ der Formel $\varphi_{<,\mathit{succ},0}$ können wir die Elemente $a_0, a_1, a_2, a_3, \ldots$, für die

$$a_0 = 0^{\mathfrak{A}} \text{ und } (a_0,a_1) \in \mathit{succ}^{\mathfrak{A}},\ (a_1,a_2) \in \mathit{succ}^{\mathfrak{A}},\ (a_2,a_3) \in \mathit{succ}^{\mathfrak{A}},\ \ldots$$

mit den natürlichen Zahlen $0, 1, 2, 3, \ldots$ identifizieren. Die folgende Formel erzwingt in ihren Modellen, dass die Variablen $z_0, z_1, .., z_s$ mit diesen Elementen $a_0, a_1, .., a_s$ (also im Grunde mit den natürlichen Zahlen $0, 1, .., s$) belegt werden:

$$\varphi_{\textit{Zahlen}}(z_0, z_1, .., z_s) \quad := \quad z_0 = 0 \ \wedge \bigwedge_{i=1}^{s} \textit{succ}(z_{i-1}, z_i).$$

Der FO$[\sigma]$-Satz φ_M wird nun folgendermaßen gewählt:

$$\begin{aligned}\varphi_M \quad &:= \\ &\varphi_{<,\textit{succ},0} \ \wedge \ \exists z_0 \cdots \exists z_s \ \big(\varphi_{\textit{Zahlen}}(z_0, .., z_s) \wedge \\ &\qquad\qquad \varphi_{\textit{Band}} \ \wedge \ \varphi_{\textit{Kopf}} \ \wedge \ \varphi_{\textit{Zustand}} \ \wedge \ \varphi_{\textit{Start}} \ \wedge \ \varphi_{\textit{Schritt}} \big) \ ,\end{aligned}$$

wobei die Formeln der letzten Zeile wie folgt gewählt sind:

$\varphi_{\textit{Band}}$ besagt, dass zu jedem Zeitpunkt auf jeder Bandposition genau ein Symbol des Arbeitsalphabets $\Gamma = \{0, .., s_\Gamma\}$ steht:

$$\varphi_{\textit{Band}} \ := \ \forall x \, \forall y \, \exists z \ \Big(B(x, y, z) \ \wedge \ \big(\bigvee_{i=0}^{s_\Gamma} z = z_i \big) \ \wedge \ \forall z' \big(B(x, y, z') \rightarrow z' = z \big) \Big).$$

$\varphi_{\textit{Kopf}}$ besagt, dass zu jedem Zeitpunkt der Schreib-/Lesekopf auf genau einer Bandposition steht:

$$\varphi_{\textit{Kopf}} \ := \ \forall x \, \exists y \ \Big(K(x, y) \ \wedge \ \forall y' \big(K(x, y') \rightarrow y' = y \big) \Big).$$

$\varphi_{\textit{Zustand}}$ besagt, dass die Maschine zu jedem Zeitpunkt in genau einem Zustand aus der Zustandsmenge $Q = \{0, 1, .., s_Q\}$ ist:

$$\varphi_{\textit{Zustand}} \ := \ \forall x \, \exists z \ \Big(Z(x, z) \ \wedge \ \big(\bigvee_{i=0}^{s_Q} z = z_i \big) \ \wedge \ \forall z' \big(Z(x, z') \rightarrow z' = z \big) \Big).$$

Die Formel $\varphi_{\textit{Start}}$ besagt, dass M zum Zeitpunkt 0 in der Anfangskonfiguration bei Eingabe des leeren Worts ist, d.h. sie ist im Startzustand $q_0 = 0$, ihr Schreib-/Lesekopf steht auf Bandposition 0, und auf jeder Bandposition steht das Blank-Symbol $\square = 0$:

$$\varphi_{\textit{Start}} \ := \ Z(0, 0) \ \wedge \ K(0, 0) \ \wedge \ \forall y \, B(0, y, 0).$$

Die Formel $\varphi_{\textit{Schritt}}$ besagt für jeden Zeitpunkt t: Falls M zum Zeitpunkt t nicht in einem Endzustand ist, so ist sie im Zeitpunkt $t' := t + 1$ in einem laut Übergangsrelation zulässigen Zustand q' und hat das entsprechende Symbol auf's Band geschrieben, den Kopf an die richtige Stelle bewegt und die Beschriftung aller anderen Bandpositionen nicht verändert.

(Zur besseren Lesbarkeit werden in der folgenden Formel die Symbole t, p, t', p' etc. benutzt, um Variablen der Logik erster Stufe zu bezeichnen.)

$\varphi_{\textit{Schritt}} :=$

$$\forall t\, \forall p \bigwedge_{q \in Q \setminus F} \bigwedge_{\gamma \in \Gamma} \Big(\Big(K(t,p) \ \wedge\ Z(t,z_q) \ \wedge\ B(t,p,z_\gamma) \Big) \ \rightarrow \Big(\exists t'\, \exists p' \, \big(\textit{succ}(t,t') \ \wedge\ K(t',p') \ \wedge \big(\forall p'' \, \big(p'' = p \ \vee \bigwedge_{\gamma'' \in \Gamma} (B(t',p'',z_{\gamma''}) \leftrightarrow B(t,p'',z_{\gamma''})) \big) \big) \wedge \bigvee_{\substack{q',\gamma',m\,:\\ (q,\gamma,q',\gamma',m) \in \Delta}} \big(Z(t',z_{q'}) \ \wedge\ B(t',p,z_{\gamma'}) \ \wedge\ \chi_m(p,p') \big) \big) \Big) \Big),$$

wobei die Formel $\chi_m(p,p')$ die jeweilige Kopfbewegung für $m \in \{1,-1,0\}$ beschreibt:

$$\chi_1(p,p') \ :=\ \textit{succ}(p,p'), \qquad \chi_{-1}(p,p') \ :=\ \textit{succ}(p',p), \qquad \chi_0(p,p') \ :=\ p = p'.$$

Insgesamt sind wir nun fertig mit der Konstruktion der Formel φ_M, die die Berechnung von M bei leerem Eingabewort beschreibt. Gemäß der Konstruktion von φ_M und $\mathfrak{A}_M$ gilt:

- $\mathfrak{A}_M \models \varphi_M$, und
- in jeder σ-Struktur $\mathfrak{A}$ mit $\mathfrak{A} \models \varphi_M$ gibt es Elemente $a_0, a_1, a_2, \ldots$, die den natürlichen Zahlen $0, 1, 2, \ldots$ entsprechen, und schränkt man $\mathfrak{A}$ ein auf das Teiluniversum

  $$\{a_i \,:\, i \in A_M\},$$

  so erhält man eine Struktur, die isomorph zu $\mathfrak{A}_M$ ist.

Insbesondere heißt das, dass das Universum von jedem Modell von φ_M mindestens so groß wie das Universum von $\mathfrak{A}_M$ ist. D.h. φ_M hat genau dann ein endliches Modell, wenn A_M endlich, d.h. die Berechnung von M bei leerem Eingabewort endlich ist.
Wenn das Erfüllbarkeitsproblem für $\text{FO}[\sigma]$ auf Fin entscheidbar wäre, wäre also auch das Halteproblem H_ε entscheidbar, denn bei Eingabe einer DTM M könnte man zunächst den Satz φ_M konstruieren und dann auf endliche Erfüllbarkeit testen. Dies widerspricht aber der Tatsache, dass H_ε nicht entscheidbar ist. □

1.63 Bemerkung. Man kann sogar zeigen, dass der Satz von Trakhtenbrot für die Signatur $\{E\}$ gilt, die aus nur einem 2-stelligen Relationssymbol besteht. (Daraus folgt dann natürlich auch, dass der Satz von Trakhtenbrot für *jede* Signatur σ gilt, die mindestens ein 2-stelliges Relationssymbol enthält.)
Um dies zu beweisen, braucht man lediglich Strukturen über der im Beweis von Theorem 1.62 benutzten Signatur σ auf geeignete Weise durch Strukturen über der Signatur $\{E\}$ zu kodieren. Näheres dazu am Ende von Kapitel 3 in Verbindung mit Satz 3.72.

Anwendungen des Satzes von Trakhtenbrot:

Unter Verwendung des Satzes von Trakhtenbrot kann man die Unentscheidbarkeit vieler konkreter Logikprobleme beweisen. Im Folgenden wird dies exemplarisch an der Eigenschaft der *Ordnungsinvarianz* von Formeln dargelegt.

Ordnungsinvarianz:

1.64 Definition. Sei σ eine Signatur und sei $<$ ein 2-stelliges Relationssymbol, das *nicht* zu σ gehört. Ein $\text{FO}[\sigma \dot\cup \{<\}]$-Satz φ heißt *ordnungsinvariant auf* Fin, falls für alle endlichen σ-Strukturen $\mathfrak{A}$ und alle linearen Ordnungen $<_1^{\mathfrak{A}}$ und $<_2^{\mathfrak{A}}$ auf dem Universum von $\mathfrak{A}$ gilt:

$$(\mathfrak{A}, <_1^{\mathfrak{A}}) \models \varphi \quad \Longleftrightarrow \quad (\mathfrak{A}, <_2^{\mathfrak{A}}) \models \varphi.$$

Mit Hilfe des Satzes von Trakhtenbrot kann man leicht einen Beweis für den folgenden Satz finden.

1.65 Satz. *Für jede Signatur σ, die mindestens ein 2-stelliges Relationssymbol und mindestens ein 1-stelliges Relationssymbol enthält, ist das folgende Problem unentscheidbar:*

ORDNUNGSINVARIANZ:
Eingabe: Ein $\text{FO}[\sigma \dot\cup \{<\}]$-Satz φ.
Frage: Ist φ ordnungsinvariant auf Fin ?

Beweis: O.B.d.A. sei $\sigma = \{E, Q\}$, wobei E ein 2-stelliges und Q ein 1-stelliges Relationssymbol ist. Angenommen, das Problem der Ordnungsinvarianz ist doch entscheidbar, d.h. es gibt eine TM M, die bei Eingabe eines $\text{FO}[\sigma \dot\cup \{<\}]$-Satzes φ entscheidet, ob φ ordnungsinvariant auf Fin ist.
Wir nutzen diese TM M, um eine neue TM M' zu bauen, die bei Eingabe eines $\text{FO}[\{E\}]$-Satzes ψ entscheidet, ob ψ ein endliches Modell hat, d.h., die das Erfüllbarkeitsproblem für $\text{FO}[\{E\}]$ auf Fin löst. Dies steht aber im Widerspruch zum Satz von Trakhtenbrot (Theorem 1.62 und Bemerkung 1.63).
Bei Eingabe eines $\text{FO}[\{E\}]$-Satzes ψ tut M' folgendes: Zunächst schreibt sie die Formel

$$\varphi \quad := \quad (\,\psi \;\rightarrow\; \chi\,),$$

wobei

$$\chi \quad := \quad \exists x \forall y \,\big(Q(x) \;\wedge\; (x < y \;\vee\; x = y)\big)$$

auf das Arbeitsband und startet dann die TM M, die (laut unserer Annahme) bei Eingabe φ entscheidet, ob φ ordnungsinvariant auf Fin ist.
Die Formel φ ist gerade so konstruiert, dass gilt:

φ ist ordnungsinvariant auf Fin $\quad\Longleftrightarrow\quad$ ψ hat kein endliches Modell,

denn: Für alle endlichen σ-Strukturen $\mathfrak{A}$ und jede lineare Ordnung $<^{\mathfrak{A}}$ auf A gilt

$$(\mathfrak{A}, <^{\mathfrak{A}}) \models \chi \quad \Longleftrightarrow \quad \text{das minimale Element in } \mathfrak{A} \text{ bzgl. der Ordnung } <^{\mathfrak{A}} \text{ liegt in } Q^{\mathfrak{A}}.$$

Insbesondere ist χ nicht ordnungsinvariant auf Fin. Dies hat zur Folge, dass die Formel φ genau dann ordnungsinvariant auf Fin ist, wenn es gar keine endliche Struktur $\mathfrak{A}$ mit $\mathfrak{A} \models \psi$ gibt, d.h., wenn ψ kein endliches Modell hat.
Somit ist M' eine Turing-Maschine, die das endliche Erfüllbarkeitsproblem für FO$[\{E\}]$ entscheidet. Dies ist aber ein Widerspruch zum Satz von Trakhtenbrot. □

Das Query Containment Problem:

Ähnlich wie bei der Ordnungsinvarianz (nur viel einfacher) kann man auch folgendes zeigen:

1.66 Satz. *Sei σ eine Signatur, die mindestens ein 2-stelliges Relationssymbol enthält. Dann ist das folgende Problem unentscheidbar:*

> QUERY CONTAINMENT PROBLEM FÜR FO$[\sigma]$ IN Fin:
> *Eingabe: Zwei* FO$[\sigma]$*-Formeln $\varphi_1(x_1, .., x_r)$ und $\varphi_2(x_1, .., x_r)$ (mit $r \in \mathbb{N}$).*
> *Frage: Gilt für alle $\mathfrak{A} \in$ Fin, dass $\varphi_1(\mathfrak{A}) \subseteq \varphi_2(\mathfrak{A})$?*

Beweis: Übung. □

Monotonie:

Ähnlich wie bei der Ordnungsinvarianz kann man auch zeigen, dass man für eine gegebene FO-Formel nicht entscheiden kann, ob diese *monoton* in dem folgenden Sinne ist:

1.67 Definition. Seien $r, s \in \mathbb{N}$, und sei σ eine Signatur und R ein r-stelliges Relationssymbol, das *nicht* in σ liegt.
Eine Formel $\varphi(x_1, .., x_s) \in$ FO$[\sigma \dot\cup \{R\}]$ heißt *monoton in R auf* Fin, wenn für alle endlichen σ-Strukturen $\mathfrak{A}$ und alle Relationen $R_1^{\mathfrak{A}}, R_2^{\mathfrak{A}} \subseteq A^r$ gilt:

$$\text{Falls } \; R_1^{\mathfrak{A}} \subseteq R_2^{\mathfrak{A}}, \;\; \text{so} \quad \varphi\big(\mathfrak{A}, R_1^{\mathfrak{A}}\big) \subseteq \varphi\big(\mathfrak{A}, R_2^{\mathfrak{A}}\big),$$

wobei $\varphi(\mathfrak{A}, R_i^{\mathfrak{A}}) := \{\vec{a} \in A^s \,:\, (\mathfrak{A}, R_i^{\mathfrak{A}}) \models \varphi[\vec{a}]\}$.

1.68 Satz. *Sei σ eine Signatur, die mindestens ein 2-stelliges Relationssymbol enthält, und sei R ein Relationssymbol, das nicht zu σ gehört. Dann ist das folgende Problem unentscheidbar:*

> MONOTONIE:
> *Eingabe: Eine* FO$[\sigma \dot\cup \{R\}]$*-Formel $\varphi(x_1, .., x_s)$.*
> *Frage: Ist $\varphi(x_1, .., x_s)$ monoton in R auf* Fin *?*

Beweis: Übung. □

Allgemeingültigkeit:

Ein wichtiges Resultat aus der *mathematischen Logik* ist die Existenz eines korrekten und *vollständigen* Beweissystems für die Logik erster Stufe. Insbesondere folgt daraus (für jede Signatur σ), dass das Problem

> ALLGEMEINGÜLTIGKEIT:
> *Eingabe:* Ein $\mathrm{FO}[\sigma]$-Satz φ.
> *Frage:* Ist φ *allgemeingültig*, d.h. gilt für alle (endlichen oder unendlichen) σ-Strukturen $\mathfrak{A}$, dass $\mathfrak{A} \models \varphi$?

semi-entscheidbar ist.

Als wichtige Folgerung aus dem Satz von Trakhtenbrot erhält man, dass die Einschränkung des Allgemeingültigkeitsproblems auf die Klasse Fin aller *endlichen* Strukturen weder entscheidbar noch semi-entscheidbar ist:

1.69 Korollar. *Für jede Signatur σ, die mindestens ein 2-stelliges Relationssymbol enthält, ist das folgende Problem nicht semi-entscheidbar:*

> ALLGEMEINGÜLTIGKEIT AUF Fin:
> *Eingabe:* Ein $\mathrm{FO}[\sigma]$-Satz φ.
> *Frage:* Gilt für alle *endlichen* σ-Strukturen $\mathfrak{A}$, dass $\mathfrak{A} \models \varphi$?

Beweis: Übung. □

2 Deskriptive Komplexität

Klassische Komplexitätstheorie:

Einordnen von Problemen hinsichtlich der zu ihrer *Lösung* benötigten *Rechenressourcen* wie z.B. Zeit oder Platz.

Deskriptive Komplexitätstheorie:

Einordnen der Probleme hinsichtlich der zu ihrer *Beschreibung* nötigen *logischen* Ressourcen wie z.B.

- Art und Anzahl der Quantoren,
- Anzahl der Quantorenalternierungen, d.h. Anzahl der Wechsel zwischen Existenzquantoren und Allquantoren
- Schachtelung von Negationen

Ziel dieses Kapitels:

Zu einer Komplexitätsklasse $\mathscr{K}$ eine "natürliche" Logik $\mathscr{L}$ finden, so dass man mit Sätzen aus $\mathscr{L}$ genau diejenigen Probleme defineren kann, die zur Komplexitätsklasse $\mathscr{K}$ gehören.

Vereinbarung: Innerhalb von Kapitel 2 sind alle Strukturen *endlich*!

2.1 Definition (Logische Beschreibung von Komplexitätsklassen)**.**
Sei $\mathscr{L}$ eine Logik, $\mathscr{K}$ eine Komplexitätsklasse und S eine unter Isomorphie abgeschlossene[1] Klasse endlicher Strukturen.
$\mathscr{L}$ *beschreibt* $\mathscr{K}$ *auf* S (engl.: $\mathscr{L}$ *captures* $\mathscr{K}$ auf S), falls die folgenden Bedingungen (a) und (b) erfüllt sind:

(a) Für jeden Satz $\varphi \in \mathscr{L}$ ist die Sprache

$$L_{\mathsf{S}}(\varphi) \;:=\; \{\mathit{enc}(\mathfrak{A}) \,:\, \mathfrak{A} \in \mathsf{S} \text{ und } \mathfrak{A} \models \varphi\} \;\in\; \mathscr{K}.$$

(D.h.: Die Datenkomplexität des Auswertungsproblems für $\mathscr{L}$ auf S liegt in $\mathscr{K}$.)

[1] d.h. für alle Strukturen $\mathfrak{A}, \mathfrak{B}$ gilt: Falls $\mathfrak{A} \in \mathsf{S}$ und $\mathfrak{A} \cong \mathfrak{B}$, so auch $\mathfrak{B} \in \mathsf{S}$.

(b) Für jede Signatur σ und jede unter Isomorphie abgeschlossene Klasse $\mathsf{C} \subseteq \mathsf{S}$ von σ-Strukturen gilt: Falls

$$L_{\mathsf{C}} := \{enc(\mathfrak{A}) : \mathfrak{A} \in \mathsf{C}\} \in \mathscr{K},$$

so gibt es einen Satz $\varphi \in \mathscr{L}$, so dass[2]

$$\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\varphi).$$

Falls eine Logik $\mathscr{L}$ eine Komplexitätsklasse $\mathscr{K}$ auf der Klasse Fin aller endlichen Strukturen beschreibt, so sagen wir auch kurz: $\mathscr{L}$ *beschreibt* $\mathscr{K}$.

2.2 Bemerkungen.

(a) Man beachte, dass jede Struktur $\mathfrak{A}$ *mehrere* mögliche Kodierungen *enc*$(\mathfrak{A})$ hat, nämlich i.d.R. für jede lineare Ordnung $<$ auf ihrem Universum eine andere.
Die Sprache L_{C} ist so definiert, dass sie aus *sämtlichen* Kodierungen jeder Struktur in C besteht.

(b) Isomorphe Strukturen haben dieselben Kodierungen, denn:
Seien $\mathfrak{A}$ und $\mathfrak{B}$ isomorphe Strukturen und sei h ein Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$. Ferner sei $<^{\mathfrak{A}}$ eine lineare Ordnung auf A, d.h. $A = \{a_0 <^{\mathfrak{A}} a_1 <^{\mathfrak{A}} \cdots <^{\mathfrak{A}} a_{n-1}\}$. Wählt man für B die lineare Ordnung $<^{\mathfrak{B}}$ so, dass

$$h(a_0) <^{\mathfrak{B}} h(a_1) <^{\mathfrak{B}} \cdots <^{\mathfrak{B}} h(a_{n-1}),$$

so ist das Wort *enc*$(\mathfrak{B})$, das man mit der linearen Ordnung $<^{\mathfrak{B}}$ erhält, identisch zu dem Wort *enc*$(\mathfrak{A})$, das man mit der linearen Ordnung $<^{\mathfrak{A}}$ erhält.

(c) Wir werden im Folgenden oft Strukturklassen C, die unter Isomorphie abgeschlossen sind, mit der Menge L_{C} ihrer Kodierungen identifizieren. Insbesondere identifizieren wir die Sprache $L_{\mathsf{S}}(\varphi)$ mit der Modellklasse $\mathrm{Mod}_{\mathsf{S}}(\varphi)$.
Wir schreiben dann kurz "$\mathsf{C} \in \mathscr{K}$" an Stelle von "$L_{\mathsf{C}} \in \mathscr{K}$", und wir schreiben "$\mathrm{Mod}_{\mathsf{S}}(\varphi) \in \mathscr{K}$" an Stelle von "$L_{\mathsf{S}}(\varphi) \in \mathscr{K}$" (vgl. Definition 2.1).

Wir suchen nun Logiken, die Komplexitätsklassen wie NP, P, PSPACE, LOGSPACE beschreiben. Die *Logik erster Stufe* ist dafür zu ausdrucksschwach, da sie keine Konstrukte enthält, die z.B. rekursive Definitionen oder Aussagen der Art "es gibt einen Pfad, der die Eigenschaft XYZ hat" ermöglichen. (Später, in Kapitel 3 (Ehrenfeucht-Fraïssé Spiele) werden wir diese Ausdrucksschwäche der Logik erster Stufe auch beweisen.) Um Komplexitätsklassen beschreiben zu können, betrachten wir daher geeignete Erweiterungen von FO.

[2] Zur Erinnerung: Gemäß Definition 1.10 gilt: $\mathrm{Mod}_{\mathsf{S}}(\varphi) = \{\mathfrak{A} \in \mathsf{S} : \mathfrak{A} \models \varphi\}$.

2.1 Die Logik zweiter Stufe und der Satz von Fagin

Die erste solche Erweiterung ist die folgendermaßen definierte *Logik zweiter Stufe* (SO), die die Logik erster Stufe um Quantoren der Form $\exists X$ bzw. $\forall X$ erweitert, wobei X eine Relationsvariable ist.

2.1.1 Syntax und Semantik der Logik zweiter Stufe (SO)

2.3 Definition (Syntax der Logik zweiter Stufe SO)**.** Sei σ eine Signatur.

(a) $\text{Var}_2 := \{\text{Var}_i^k : i, k \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge alle *Variablen zweiter Stufe* (oder *Relationsvariablen*). Die Relationsvariable Var_i^k hat die *Stelligkeit* $ar(\text{Var}_i^k) = k$.
D.h. für jede Zahl k gibt es abzählbar viele Relationsvariablen der Stelligkeit k.

(b) Die Formelmenge $\text{SO}[\sigma]$ ist induktiv durch die Regeln (A1), (A2), (BC) und (Q1) der Logik erster Stufe (wobei "$\text{FO}[\sigma]$" jeweils durch "$\text{SO}[\sigma]$" ersetzt werden muss), sowie die beiden folgenden Regeln (A3) und (Q2) definiert:

(A3) Ist $X \in \text{Var}_2$ mit Stelligkeit $k := ar(X)$ und sind $v_1, \ldots, v_k \in \text{Var}_1 \cup \{c \in \sigma : c \text{ ist ein Konstantensymbol}\}$, so gehört $X(v_1, \ldots, v_k)$ zu $\text{SO}[\sigma]$.

(Q2) Ist ψ eine Formel in $\text{SO}[\sigma]$ und $X \in \text{Var}_2$, so gehören auch folgende Formeln zu $\text{SO}[\sigma]$:
- $\exists X\ \psi$ (Existenzquantor)
- $\forall X\ \psi$ (Allquantor).

(c) Die mit $(A1)$, $(A2)$ und $(A3)$ gebildeten Formeln heißen *atomare σ-Formeln*.

2.4 Bemerkungen.

(a) Analog zu Definition 1.8 bezeichnen wir mit

$$\textit{frei}(\varphi)$$

die Menge aller Variablen erster oder zweiter Stufe, die frei in einer $\text{SO}[\sigma]$-Formel φ vorkommen. Wir schreiben oft $\varphi(x_1, \ldots, x_s, X_1, \ldots, X_t)$ um anzudeuten, dass

$$\textit{frei}(\varphi) \ = \ \{x_1, \ldots, x_s, X_1, \ldots, X_t\}.$$

Eine $\text{SO}[\sigma]$-Formel φ heißt *Satz*, falls $\textit{frei}(\varphi) = \varnothing$, φ also keine freien Variablen erster Stufe und keine freien Relationsvariablen hat.

(b) Die *Semantik* der Logik zweiter Stufe ist auf die offensichtliche Weise definiert, z.B. gilt für eine σ-Struktur $\mathfrak{A}$, eine k-stellige Relationsvariable X und eine $\text{SO}[\sigma]$-Formel φ

$$\mathfrak{A} \models \exists X\, \varphi \quad \Longleftrightarrow \quad \text{es gibt eine Relation } X^{\mathfrak{A}} \subseteq A^k, \text{ so dass } (\mathfrak{A}, X^{\mathfrak{A}}) \models \varphi.$$

Ist $\varphi(x_1, .., x_s, X_1, .., X_t)$ eine SO$[\sigma]$-Formel, $\mathfrak{A}$ eine σ-Struktur, $a_1, .., a_s \in A$ und $A_1, .., A_t$ Relationen über A mit $A_i \subseteq A^{ar(X_i)}$, so schreiben wir

$$\mathfrak{A} \models \varphi[a_1, .., a_s, A_1, .., A_t] \quad \text{bzw.} \quad (\mathfrak{A}, A_1, .., A_t) \models \varphi[a_1, .., a_s]$$

falls die Formel φ in der Struktur $\mathfrak{A}$ *erfüllt* ist, wenn die freien Vorkommen der Variablen $x_1, .., x_s, X_1, .., X_t$ durch die Werte $a_1, .., a_s, A_1, .., A_t$ belegt werden.

(c) Für eine Signatur $\sigma = \{R_1, .., R_k, c_1, .., c_\ell\}$ schreiben wir oft SO$[R_1, .., R_k, c_1, .., c_\ell]$ an Stelle von SO$[\{R_1, .., R_k, c_1, .., c_\ell\}]$, um die Klasse aller SO$[\sigma]$-Formeln zu bezeichnen.

2.5 Beispiele (Graphprobleme)**.**
Sei $\sigma := \{E\}$ die Signatur für Graphen, E also ein 2-stelliges Relationssymbol.

(a) Die Formel $\Phi_{\text{3-col}}$ aus Beispiel 0.2 ist ein SO$[\sigma]$-Satz, der in einem Graphen $G = (V, E)$ genau dann gilt, wenn der Graph 3-färbbar ist.
Also gilt: $\text{Mod}_{\mathsf{Fin}}(\Phi_{\text{3-col}}) = \{G : G \text{ ist ein 3-färbbarer Graph}\}$.

(b) Die Formel

$$\Phi_{\text{conn}} := \forall X \Big(\big(\exists x\, X(x) \wedge \exists y\, \neg X(y)\big) \rightarrow \exists u\, \exists v \big((E(u,v) \vee E(v,u)) \wedge X(u) \wedge \neg X(v)\big)\Big)$$

besagt in ihren Modellen $G = (V, E)$, dass für jede nicht-leere Knotenmenge $X \neq V$ gilt: Es gibt eine Kante zwischen einem Knoten in X und einem Knoten außerhalb von X. Somit gilt für alle Graphen G:

$$\begin{aligned} G \not\models \Phi_{\text{conn}} \quad &\Longleftrightarrow \quad \text{es gibt eine Knotenmenge } X \text{ mit } \varnothing \neq X \neq V, \text{ so} \\ &\qquad\quad \text{dass keine Kante zwischen } X \text{ und } V \setminus X \text{ verläuft} \\ &\Longleftrightarrow \quad G \text{ ist nicht zusammenhängend.} \end{aligned}$$

Also gilt: $\text{Mod}_{\mathsf{Fin}}(\Phi_{\text{conn}}) = \{G : G \text{ ist ein zusammenhängender Graph}\}$.

(c) Ein *Hamiltonkreis* in einem Graphen $G = (V, E)$ ist eine Folge $v_0, .., v_n$ von Knoten, so dass $V = \{v_0, .., v_n\}$ und es eine Kante von v_n nach v_0 und eine Kante von v_i nach v_{i+1}, für alle $i < n$, gibt. Aus der Komplexitätstheorie ist bekannt, dass das Problem

> HAMILTONKREIS:
> *Eingabe:* Ein Graph G.
> *Frage:* Hat G einen Hamiltonkreis?

NP-vollständig ist.
Wir geben nun einen SO-Satz Φ_{Ham} an, so dass

$$\text{Mod}_{\mathsf{Fin}}(\Phi_{\text{Ham}}) = \{G : G \text{ hat einen Hamiltonkreis}\}.$$

Die Formel Φ_{Ham} ist von der Form

$$\exists R_< \, \exists R_{succ} \, \exists z_0 \, \exists z_{max} \, \Big(\varphi_{<,succ,0,max}(R_<, R_{succ}, z_0, z_{max}) \; \wedge \\ E(z_{max}, z_0) \; \wedge \; \forall x \, \forall y \, \big(R_{succ}(x,y) \rightarrow E(x,y)\big) \Big),$$

wobei

$$\varphi_{<,succ,0,max}(R_<, R_{succ}, z_0, z_{max}) \; := \\ \varphi_{<,succ,0}(R_<, R_{succ}, z_0) \; \wedge \; \forall x \, \big(R_<(x, z_{max}) \vee \; x{=}z_{max} \big)$$

die Formel $\varphi_{<,succ,0}$ aus dem Beweis von Theorem 1.62 benutzt, um in ihren Modellen zu erzwingen, dass $R_<$ eine lineare Ordnung, R_{succ} deren Nachfolger-Relation, z_0 deren kleinstes und z_{max} deren größtes Element ist. Die letzte Zeile der Formel Φ_{Ham} besagt, dass die Folge $v_0, v_1, \ldots, v_n$, die man beim Durchlaufen der Knotenmenge entlang der Nachfolger-Relation R_{succ} erhält, einen Hamiltonkreis bildet. Insgesamt gilt für jeden endlichen Graph G:

$$G \models \Phi_{\text{Ham}} \iff G \text{ hat einen Hamiltonkreis.}$$

2.1.2 Existentielle Logik zweiter Stufe und der Satz von Fagin

In diesem Abschnitt beweisen wir den Satz von Fagin, der besagt, dass die Komplexitätsklasse NP aus genau den Problemen besteht, die durch Formeln der *existentiellen Logik zweiter Stufe* beschrieben werden können.

2.6 Definition (Existentielle Logik zweiter Stufe ESO)**.** Sei σ eine Signatur.
Die Formelmenge $\text{ESO}[\sigma]$ aller *existentiellen* $\text{SO}[\sigma]$*-Formeln* besteht aus allen $\text{SO}[\sigma]$-Formeln Φ, die von der Gestalt

$$\exists X_1 \exists X_2 \cdots \exists X_d \, \varphi$$

sind, wobei $d \geqslant 0$, $X_1, \ldots, X_d$ Relationsvariablen und φ eine $\text{FO}[\sigma \dot\cup \text{Var}_2]$-Formel ist.

2.7 Beispiele. Die $\text{SO}[E]$-Formeln $\Phi_{\text{3-col}}$ und Φ_{Ham} aus Beispiel 2.5 sind $\text{ESO}[E]$-Formeln. Die Formel Φ_{conn} aus Beispiel 2.5 ist keine $\text{ESO}[E]$-Formel.

2.8 Theorem (Satz von Fagin, 1974)**.** ESO *beschreibt* NP *auf* Fin.

Beweis: Sei σ eine Signatur.

"$\subseteq$": Zunächst zeigen wir, dass für jeden $\text{ESO}[\sigma]$-Satz Φ gilt:

$$L_{\text{Fin}}(\Phi) \; := \; \{\mathit{enc}(\mathfrak{A}) \, : \, \mathfrak{A} \in \text{Fin und } \mathfrak{A} \models \Phi\} \; \in \; \text{NP},$$

d.h. wir müssen zeigen, dass das Problem

$\mathrm{Mod}_{\mathsf{Fin}}(\Phi)$:
Eingabe: Eine endliche Struktur $\mathfrak{A}$.
Frage: Gilt $\mathfrak{A} \models \Phi$?

durch eine nichtdeterministische Turing-Maschine lösbar ist, deren Zeitschranke polynomiell in der Größe des Universums der Eingabe-Struktur $\mathfrak{A}$ ist.
Sei dazu der gegebene ESO$[\sigma]$-Satz Φ von der Form

$$\exists X_1 \cdots \exists X_d \, \varphi$$

mit $d \geqslant 0$ und $\varphi \in \mathrm{FO}[\sigma \dot\cup \{X_1, \ldots, X_d\}]$.
Aus Lemma 1.59 folgt, dass es eine deterministische Turing-Maschine M' gibt, die das Problem

$$L' \;:=\; \{\mathit{enc}(\mathfrak{A}') \,:\, \mathfrak{A}' \text{ ist eine } (\sigma \dot\cup \{X_1, \ldots, X_d\})\text{-Struktur mit } \mathfrak{A}' \models \varphi\}$$

löst und dabei nur Platz $\mathcal{O}(\log n)$ benutzt, wobei n die Größe des Universums von $\mathfrak{A}'$ bezeichnet. Aus Satz 1.25 folgt, dass es eine Konstante $d' \in \mathbb{N}$ gibt, so dass M' Zeitschranke $n^{d'}$ hat. Unter Verwendung von M' kann man leicht eine NTM M bauen, die dem folgenden (nichtdeterministischen) Algorithmus gemäß das Problem $\mathrm{Mod}_{\mathsf{Fin}}(\Phi)$ löst:

Eingabe: Eine endliche σ-Struktur $\mathfrak{A}$.
Frage: Gilt $\mathfrak{A} \models \Phi$?
Lösung:

1. Rate Belegungen $X_1^{\mathfrak{A}}, \ldots, X_d^{\mathfrak{A}}$ für die Relationsvariablen $X_1, \ldots, X_d$,
   d.h. rate (Kodierungen von) Relationen $X_i^{\mathfrak{A}} \subseteq A^{ar(X_i)}$, für $i = 1, \ldots, d$.
2. Entscheide, ob
   $$(\mathfrak{A}, X_1^{\mathfrak{A}}, \ldots, X_d^{\mathfrak{A}}) \;\models\; \varphi,$$
   indem die Turing-Maschine M' mit Eingabe $\mathit{enc}(\mathfrak{A}')$, für $\mathfrak{A}' := (\mathfrak{A}, X_1^{\mathfrak{A}}, \ldots, X_d^{\mathfrak{A}})$ gestartet wird.

Für Schritt 1 wird der Nicht-Determinismus genutzt. Da jede Relation $X_i^{\mathfrak{A}} \subseteq A^{ar(X_i)}$ durch ein 0-1-Wort der Länge $n^{ar(X_i)}$ kodiert werden kann, benötigt eine NTM für Schritt 1 nur polynomiell viele Schritte. Da die TM M' polynomiell zeitbeschränkt ist, reicht auch für Schritt 2 eine polynomielle Anzahl von Schritten aus.
Die so konstruierte NTM M löst also das Problem $L_{\mathsf{Fin}}(\Phi)$ und ist polynomiell zeitbeschränkt. Es gilt also: $L_{\mathsf{Fin}}(\Phi) \in \mathrm{NP}$.

"$\supseteq$": Sei C eine unter Isomorphie abgeschlossene Klasse endlicher σ-Strukturen, so dass

$$L_{\mathsf{C}} \;:=\; \{\mathit{enc}(\mathfrak{A}) \,:\, \mathfrak{A} \in \mathsf{C}\} \;\in\; \mathrm{NP}.$$

D.h. es gibt eine NTM $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ mit $F = F_{\text{akz}} \dot\cup F_{\text{verw}}$ und eine Konstante $k \in \mathbb{N}$, so dass M bei Eingabe (der Kodierung) einer endlichen σ-Struktur $\mathfrak{A}$ entscheidet,

ob $\mathfrak{A} \in \mathsf{C}$ und dabei weniger als n^k Schritte macht. Mit n bezeichnen wir hierbei — und bis zum Ende dieses Beweises immer — die Größe des Universums der Eingabe-Struktur $\mathfrak{A}$. O.B.d.A. können wir annehmen, dass

- F_{akz} aus genau einem akzeptierenden Zustand q_{akz} besteht,
- jeder Lauf von M bei jeder Eingabe-Struktur $\mathfrak{A}$ mit $|A| \geqslant 2$ nach *genau* $n^k - 1$ Schritten in einem (akzeptierenden oder verwerfenden) Endzustand endet und
- $n^k \geqslant |\mathit{enc}(\mathfrak{A})|$.

Unser Ziel ist, einen $\text{ESO}[\sigma]$-Satz Φ zu finden, so dass $\mathsf{C} = \text{Mod}_{\mathsf{Fin}}(\Phi)$, d.h. für jede endliche σ-Struktur $\mathfrak{A}$ gilt:

$$\begin{aligned} \mathfrak{A} \models \Phi \quad &\iff \quad \mathfrak{A} \in \mathsf{C} \\ &\iff \quad M \text{ akzeptiert } \mathfrak{A} \\ &\iff \quad \text{es gibt einen Lauf von } M \text{ bei Eingabe } \mathit{enc}(\mathfrak{A}), \text{ der nach} \\ &\qquad\quad\ n^k - 1 \text{ Schritten in Zustand } q_{\text{akz}} \text{ endet.} \end{aligned}$$

Idee zur Konstruktion von Φ: Ähnlich wie im Beweis des Satzes von Trachtenbrot. Jetzt wird aber jeder Zeitpunkt $t \in \{0, 1, \ldots, n^k - 1\}$ durch das k-Tupel $\vec{t} := (t_1, \ldots, t_k) \in \{0, \ldots, n-1\}^k$ kodiert, für das gilt:

$$\mathit{rg}_{<_{lex}}(\vec{t}) \;=\; t.$$

Analog wird jede Bandposition $p \in \{0, 1, \ldots, n^k - 1\}$ durch ein k-Tupel $\vec{p} := (p_1, \ldots, p_k) \in \{0, \ldots, n-1\}^k$ kodiert. Ein Lauf von M bei Eingabe $\mathit{enc}(\mathfrak{A})$ wird durch folgende Relationen repräsentiert:

- eine $2k$-stellige Relation $K^{\mathfrak{A}}$ mit der Bedeutung

  $(\vec{t}, \vec{p}) \in K^{\mathfrak{A}} \iff$ der Schreib-/Lesekopf steht zum Zeitpunkt $\vec{t}$ auf Bandposition $\vec{p}$.

- für jedes Symbol $\gamma \in \Gamma$ eine $2k$-stellige Relation $B_\gamma^{\mathfrak{A}}$ mit der Bedeutung

  $(\vec{t}, \vec{p}) \in B_\gamma^{\mathfrak{A}} \iff$ auf Bandposition $\vec{p}$ steht zum Zeitpunkt $\vec{t}$ das Symbol γ.

- für jeden Zustand $q \in Q$ eine k-stellige Relation $Z_q^{\mathfrak{A}}$ mit der Bedeutung

  $\vec{t} \in Z_q^{\mathfrak{A}} \iff M$ ist zum Zeitpunkt $\vec{t}$ in Zustand q.

Damit k-Tupel $\vec{t} \in A^k$ bzw. $\vec{p} \in A^k$ überhaupt mit Tupeln aus $\{0, \ldots, n-1\}^k$, und somit mit Zahlen aus $\{0, 1, \ldots, n^k - 1\}$ identifiziert werden können, benötigen wir außerdem eine lineare Ordnung $<^{\mathfrak{A}}$ auf dem Universum A von $\mathfrak{A}$, deren Nachfolger-Relation $\mathit{succ}^{\mathfrak{A}}$, sowie Elemente $z_0^{\mathfrak{A}}$ und $z_{\mathit{max}}^{\mathfrak{A}}$ für das kleinste und das größte Element in A bzgl. der Ordnung $<^{\mathfrak{A}}$.

Der ESO$[\sigma]$-Satz Φ, der in jeder Struktur $\mathfrak{A}$ die Berechnung von M bei Eingabe *enc*$(\mathfrak{A})$ beschreibt, wird nun folgendermaßen gewählt:

$$\Phi \;:=\; \exists K \, \big(\exists B_\gamma\big)_{\gamma\in\Gamma} \, \big(\exists Z_q\big)_{q\in Q} \, \exists R_< \, \exists R_{succ} \, \exists z_0 \, \exists z_{max} \, \Big(\varphi_{<,succ,0,max}(R_<, R_{succ}, z_0, z_{max}) \;\wedge\; \varphi_{Band} \;\wedge\; \varphi_{Kopf} \;\wedge\; \varphi_{Zustand} \;\wedge\; \varphi_{Start} \;\wedge\; \varphi_{Schritt} \;\wedge\; \varphi_{Akzeptiere} \Big) \,,$$

wobei $\varphi_{<,succ,0,max}(R_<, R_{succ}, z_0, z_{max})$ die bereits in Beispiel 2.5(c) benutzte FO-Formel ist, die in ihren Modellen erzwingt, dass $R_<$ mit einer linearen Ordnung, R_{succ} mit deren Nachfolger-Relation und z_0 bzw. z_{max} mit deren kleinstem bzw. größtem Element belegt wird. Die FO-Formeln der letzten Zeile von Φ sind wie folgt gewählt:

φ_{Band} besagt, dass zu jedem Zeitpunkt auf jeder Bandposition genau ein Symbol des Arbeitsalphabets Γ steht:

$$\varphi_{Band} \;:=\; \forall x_1 \cdots \forall x_k \, \forall y_1 \cdots \forall y_k \bigvee_{\gamma\in\Gamma} \Big(B_\gamma(\vec{x},\vec{y}) \;\wedge \bigwedge_{\substack{\gamma'\in\Gamma \\ \gamma'\neq\gamma}} \neg B_{\gamma'}(\vec{x},\vec{y}) \Big).$$

φ_{Kopf} besagt, dass zu jedem Zeitpunkt der Schreib-/Lesekopf auf genau einer Bandposition steht:

$$\varphi_{Kopf} \;:=\; \forall x_1 \cdots \forall x_k \, \exists y_1 \cdots \exists y_k \Big(K(\vec{x},\vec{y}) \;\wedge\; \forall y'_1 \cdots \forall y'_k \, \big(K(\vec{x},\vec{y}\,') \leftrightarrow \bigwedge_{i=1}^{k} y_i = y'_i \big) \Big).$$

$\varphi_{Zustand}$ besagt, dass die Maschine zu jedem Zeitpunkt in genau einem Zustand aus der Zustandsmenge Q ist:

$$\varphi_{Zustand} \;:=\; \forall x_1 \cdots \forall x_k \bigvee_{q\in Q} \Big(Z_q(\vec{x}) \;\wedge \bigwedge_{\substack{q'\in Q \\ q'\neq q}} \neg Z_{q'}(\vec{x}) \Big).$$

$\varphi_{Akzeptiere}$ besagt, dass M zum Zeitpunkt n^k-1, der durch das k-Tupel $(z_{max},..,z_{max})$ repräsentiert wird, in dem *akzeptierenden* Zustand q_{akz} ist:

$$\varphi_{Akzeptiere} \;:=\; Z_{q_{\text{akz}}}(\underbrace{z_{max},..,z_{max}}_{k}).$$

Die Formel $\varphi_{Schritt}$ besagt für jeden Zeitpunkt $\vec{t}$: Falls $\vec{t}$ nicht das Ende der Berechnung ist, so ist M im Zeitpunkt $\vec{t}\,' := \vec{t}+1$ in einem laut Übergangsrelation Δ zulässigen Zustand q' und hat das entsprechende Symbol auf's Band geschrieben, den Kopf an die richtige Stelle bewegt und die Beschriftung aller anderen Bandpositionen nicht verändert.
Die Formel $\varphi_{Schritt}$ benutzt die FO-Formel

$$\textit{equal}(x_1,..,x_k,y_1,..,y_k) \;:=\; \bigwedge_{i=1}^{k} x_i = y_i \,,$$

sowie eine FO-Formel

$$\mathit{succ}_{<_{lex}}(x_1, .., x_k, y_1, .., y_k)\,,$$

die besagt, dass das k-Tupel $\vec{y}$ mit dem unmittelbaren Nachfolger (bzgl. der aus $R_<$ gebildeten lexikographischen Ordnung) des Tupels $\vec{x}$ belegt ist (genaue Konstruktion der Formel $\mathit{succ}_{<_{lex}}(\vec{x},\vec{y})$: *Übung*).
Zur besseren Lesbarkeit schreiben wir in der folgenden Formel

$$\exists \vec{t} \qquad \text{bzw.} \qquad \forall \vec{p}$$

als Abkürzung für die Quantifizierungen

$$\exists t_1 \cdots \exists t_k \qquad \text{bzw.} \qquad \forall p_1 \cdots \forall p_k\,.$$

$\varphi_{\mathit{Schritt}}$:=

$$\begin{aligned}\forall \vec{t}\,\forall \vec{p} \bigwedge_{q\in Q\setminus F}\ \bigwedge_{\gamma\in\Gamma} \Big(\Big(K(\vec{t},\vec{p}) \ \wedge\ Z_q(\vec{t}) \ \wedge\ B_\gamma(\vec{t},\vec{p}) \Big) \ \rightarrow \\ \Big(\exists \vec{t}\,' \exists \vec{p}\,' \ \big(\ \mathit{succ}_{<_{lex}}(\vec{t},\vec{t}\,') \ \wedge\ K(\vec{t}\,',\vec{p}\,') \ \wedge \\ \big(\forall \vec{p}\,'' \ \big(\mathit{equal}(\vec{p}\,'',\vec{p}) \ \vee \bigwedge_{\gamma''\in\Gamma} (B_{\gamma''}(\vec{t}\,',\vec{p}\,'') \leftrightarrow B_{\gamma''}(\vec{t},\vec{p}\,'')) \big) \big) \ \wedge \\ \bigvee_{\substack{q',\gamma',m\,:\\ (q,\gamma,q',\gamma',m)\in\Delta}} \big(\ Z_{q'}(\vec{t}\,') \ \wedge\ B_{\gamma'}(\vec{t}\,',\vec{p}) \ \wedge\ \chi_m(\vec{p},\vec{p}\,') \big) \big) \Big) \Big),\end{aligned}$$

wobei die Formel $\chi_m(\vec{p},\vec{p}\,')$ die jeweilige Kopfbewegung für $m \in \{1,-1,0\}$ beschreibt:

$$\begin{aligned}\chi_1(\vec{p},\vec{p}\,') &:= \mathit{succ}_{<_{lex}}(\vec{p},\vec{p}\,'),\\ \chi_{-1}(\vec{p},\vec{p}\,') &:= \mathit{succ}_{<_{lex}}(\vec{p}\,',\vec{p}),\\ \chi_0(\vec{p},\vec{p}\,') &:= \mathit{equal}(\vec{p},\vec{p}\,').\end{aligned}$$

Nun fehlt nur noch die Formel φ_{Start}, die besagt, dass M zum Zeitpunkt 0 (der durch das k-Tupel $\vec{z}_0 := (z_0, .., z_0)$ kodiert wird) in der Anfangskonfiguration bei Eingabe des Worts $\mathit{enc}(\mathfrak{A})$ ist, d.h. M ist im Startzustand q_0, der Schreib-/Lesekopf steht auf Bandposition 0 (die durch das k-Tupel $\vec{z}_0 := (z_0, .., z_0)$ kodiert wird), und für jedes $\vec{p} \in \{0, .., n-1\}^k$ gilt: Auf Bandposition $i := \mathit{rg}_{<_{lex}}(\vec{p}) \in \{0, 1, .., n^k-1\}$ steht

- das Symbol 0, falls an der i-ten Position des Worts $\mathit{enc}(\mathfrak{A})$ eine 0 steht,
- das Symbol 1, falls an der i-ten Position des Worts $\mathit{enc}(\mathfrak{A})$ eine 1 steht,
- das Blank-Symbol $\Box$, falls $i \geqslant |\mathit{enc}(\mathfrak{A})|$.

Dies wird durch die FO-Formel

$$\varphi_{\mathit{Start}} \;:=\; Z_{q_0}(\vec{z}_0) \;\wedge\; K(\vec{z}_0,\vec{z}_0) \;\wedge\; \forall y_1 \cdots \forall y_k \Big(\big(B_0(\vec{z}_0,\vec{y}) \leftrightarrow \zeta_0(\vec{y})\big) \;\wedge\; \big(B_1(\vec{z}_0,\vec{y}) \leftrightarrow \zeta_1(\vec{y})\big) \;\wedge\; \big(B_\square(\vec{z}_0,\vec{y}) \leftrightarrow \neg(\zeta_0(\vec{y}) \vee \zeta_1(\vec{y}))\big) \Big)$$

ausgedrückt, wobei $\zeta_0(\vec{y})$ und $\zeta_1(\vec{y})$ geeignete FO-Formeln sind, die ausdücken, dass in dem 0-1-Wort $\mathit{enc}(\mathfrak{A})$ an Position $\mathit{rg}_{<_{\mathit{lex}}}(\vec{y})$ eine 0 bzw. eine 1 steht. Im Folgenden werden die Formeln ζ_0 und ζ_1 für den speziellen Fall angegeben, dass die Signatur σ aus einem 2-stelligen Relationssymbol R_1 und einem Konstantensymbol c_1 besteht. Der allgemeine Fall, in dem σ aus mehreren Relationssymbolen verschiedener Stelligkeiten und mehreren Konstantensymbolen besteht, kann ganz analog behandelt werden.

Ist $\sigma = \{R_1, c_1\}$ (für ein 2-stelliges Relationssymbol R_1 und ein Konstantensymbol c_1), so gilt für jede σ-Struktur $\mathfrak{A}$, dass das Wort $\mathit{enc}(\mathfrak{A})$ die Länge $3n^2$ hat, wobei $n := |A|$ ist und $\mathit{enc}(\mathfrak{A})$ von der Form

$$\mathit{enc}(\mathfrak{A}) \;=\; \underbrace{1^n 0^{n^2-n}}_{\text{Länge } n^2} \; \underbrace{\mathit{enc}(R_1^{\mathfrak{A}})}_{\text{Länge } n^2} \; \underbrace{\mathit{enc}(c_1^{\mathfrak{A}})}_{\text{Länge } n^2} \;\in\; \{0,1\}^{3n^2}.$$

Die Positionen $0,..,3n^2-1$ des Worts $\mathit{enc}(\mathfrak{A})$ werden durch genau diejenigen k-Tupel $\vec{p} = (p_1,..,p_k) \in \{0,..,n-1\}^k$ kodiert, für die gilt:

$$p_1 = \cdots = p_{k-3} = 0, \quad p_{k-2} \in \{0,1,2\} \quad \text{und} \quad p_{k-1}, p_k \in \{0,..,n-1\}.$$

Daher erzwingen die folgenden Formeln $\zeta_1(\vec{y})$ und $\zeta_0(\vec{y})$, dass in ihren Modellen die Variablen $\vec{y}$ mit Werten belegt sind, die Positionen kodieren, an denen in $\mathit{enc}(\mathfrak{A})$ eine 1 bzw. eine 0 steht:

$$\zeta_1(y_1,..,y_k) \;:=\; \bigwedge_{i=1}^{k-3} y_i = z_0 \;\wedge\; \Big(\big(y_{k-2} = z_0 \;\wedge\; y_{k-1} = z_0\big) \qquad \text{(Anfangsblock } 1^n0^{n^2-n}\text{)}$$
$$\vee\; \big(R_{\mathit{succ}}(z_0, y_{k-2}) \;\wedge\; R_1(y_{k-1}, y_k)\big) \qquad \text{(Teilstück } \mathit{enc}(R_1^{\mathfrak{A}})\text{)}$$
$$\vee\; \big(\text{“}y_{k-2} = 2\text{”} \;\wedge\; y_{k-1} = z_0 \;\wedge\; y_k = c_1\big) \Big) \qquad \text{(Teilstück } \mathit{enc}(c_1^{\mathfrak{A}})\text{)}$$

$$\zeta_0(y_1,..,y_k) \;:=\; \bigwedge_{i=1}^{k-3} y_i = z_0 \;\wedge\; \Big(y_{k-2} = z_0 \;\vee\; R_{\mathit{succ}}(z_0, y_{k-2}) \;\vee\; \text{“}y_{k-2} = 2\text{”}\Big) \;\wedge\; \neg\, \zeta_1(y_1,..,y_k),$$

wobei “$y_{k-2} = 2$” für die Formel $\exists x\big(R_{\mathit{succ}}(z_0,x) \wedge R_{\mathit{succ}}(x, y_{k-2})\big)$ steht.

Insgesamt sind wir nun fertig mit der Konstruktion der ESO-Formel Φ. Man kann leicht

nachprüfen (per Induktion nach den Zeitpunkten $0, 1, \ldots, n^k - 1$), dass für jede endliche σ-Struktur $\mathfrak{A}$ gilt:

$$\mathfrak{A} \models \Phi \iff \text{es gibt einen Lauf von } M \text{ bei Eingabe } \mathit{enc}(\mathfrak{A}), \text{ der nach } n^k - 1 \text{ Schritten in dem akzeptierenden Zustand } q_{\text{akz}} \text{ endet.}$$

Da M genau die σ-Strukturen akzeptiert, die in C liegen, gilt demnach:
$\mathfrak{A} \models \Phi \iff \mathfrak{A} \in \mathsf{C}$. Somit sind wir fertig mit dem Beweis von Theorem 2.8. □

Unter Verwendung des Satzes von Fagin erhält man relativ leicht einen Beweis für das folgende Resultat (das allerdings bereits vor dem Satz von Fagin bekannt war):

2.9 Theorem (Satz von Cook, 1971)**.** *Das aussagenlogische Erfüllbarkeitsproblem*

SAT:
Eingabe: Eine aussagenlogische Formel α.
Frage: Gibt es eine Variablenbelegung, die α erfüllt?

ist NP*-vollständig.*

Beweis: SAT $\in$ NP, denn man kann leicht einen nichtdeterministischen Polynomialzeit-Algorithmus angeben, der bei Eingabe einer aussagenlogischen Formel α zunächst eine Belegung der in α vorkommenden Variablen mit Werten aus $\{0, 1\}$ "rät" und danach überprüft, ob diese Belegung die Formel α tatsächlich erfüllt.

SAT ist NP-hart: Dazu muss man für jedes Problem L in NP zeigen, dass es eine Polynomialzeit-Reduktion von L auf SAT gibt. Jedes Problem L kann man, für eine geeignete Signatur σ, mit einer Klasse C endlicher σ-Strukturen (bzw. der zugehörigen Menge L_{C}) identifizieren. Da nach Voraussetzung $L_{\mathsf{C}} \in$ NP ist, liefert der Satz von Fagin (Theorem 2.8), dass es einen ESO$[\sigma]$-Satz Φ gibt, so dass für jede endliche σ-Struktur $\mathfrak{A}$ gilt: $\mathfrak{A} \in \mathsf{C} \iff \mathfrak{A} \models \Phi$.
Der Satz Φ sei von der Form

$$\Phi \;=\; \exists X_1 \cdots \exists X_d \, \varphi,$$

wobei $d \geqslant 0$, $X_1, \ldots, X_d$ Relationsvariablen und φ ein FO$[\sigma \dot\cup \{X_1, \ldots, X_d\}]$-Satz ist.

Wir nutzen die Formel φ, um für jede endliche σ-Struktur $\mathfrak{A}$ eine aussagenlogische Formel $\alpha_{\Phi,\mathfrak{A}}$ zu konstruieren, so dass gilt

$$\mathfrak{A} \models \Phi \iff \alpha_{\Phi,\mathfrak{A}} \text{ hat eine erfüllende Belegung.}$$

Die aussagenlogische Formel $\alpha_{\Phi,\mathfrak{A}}$ benutzt aussagenlogische Variablen aus der Menge

$$V \;:=\; \{\, v_{X_i,\vec{a}} \;:\; i \in \{1, \ldots, d\} \text{ und } \vec{a} \in A^{ar(X_i)} \,\}.$$

Die Idee dabei ist, dass eine Belegung, die der Variablen $v_{X_i,\vec{a}}$ den Wahrheitswert 1 zuordnet, kodiert, dass das Tupel $\vec{a}$ in der Relation X_i liegt. Somit entspricht eine Belegung

der aussagenlogischen Variablen aus V mit Werten aus $\{0,1\}$ gerade einer Belegung der Relationsvariablen $X_1, \ldots, X_d$ mit Relationen über dem Universum von $\mathfrak{A}$.

Die aussagenlogische Formel $\alpha_{\Phi,\mathfrak{A}}$ entsteht nun aus φ, indem man nacheinander die folgenden Ersetzungen durchführt:

1. Ersetze jede Teilformel der Form $\exists x\, \psi(x, \cdots)$ durch $\bigvee_{a \in A} \psi(a, \cdots)$.
2. Ersetze jede Teilformel der Form $\forall x\, \psi(x, \cdots)$ durch $\bigwedge_{a \in A} \psi(a, \cdots)$.
3. Ersetze jedes Konstantensymbol c durch die zugehörige Konstante $c^{\mathfrak{A}}$.
4. Ersetze jedes "Atom" der Form $X_i(\vec{a})$ (für $i \in \{1, \ldots, d\}$ und $\vec{a} \in A^{ar(X_i)}$) durch die aussagenlogische Variable $v_{X_i,\vec{a}}$.
5. Ersetze jedes "Atom" der Form $R(\vec{a})$ (für $R \in \sigma$ und $\vec{a} \in A^{ar(R)}$) durch den Wahrheitswert 1, falls $\vec{a} \in R^{\mathfrak{A}}$, und durch den Wahrheitswert 0, falls $\vec{a} \notin R^{\mathfrak{A}}$.
6. Ersetze jede Gleichung der Form $a = b$ (für $a, b \in A$) durch den Wahrheitswert 1, falls $a = b$ ist, und durch den Wahrheitswert 0, falls $a \neq b$ ist.

Gemäß dieser Konstruktion gilt für alle endlichen σ-Strukturen $\mathfrak{A}$:

- $\mathfrak{A} \models \Phi \iff \alpha_{\Phi,\mathfrak{A}}$ ist erfüllbar,
- Bei Eingabe der Struktur $\mathfrak{A}$ kann man in polynomieller Zeit (also Zeit $|A|^k$, für ein $k \in \mathbb{N}$) die Formel $\alpha_{\Phi,\mathfrak{A}}$ erzeugen.

Somit ist die Abbildung f, die jeder endlichen σ-Struktur $\mathfrak{A}$ die Formel $\alpha_{\Phi,\mathfrak{A}}$ zuordnet, eine Polynomialzeit-Reduktion von dem Problem L_{C} auf das aussagenlogische Erfüllbarkeitsproblem SAT.
Insgesamt haben wir also gezeigt, dass das Problem SAT NP-vollständig ist. □

2.1.3 Logische Charakterisierung der Polynomialzeit-Hierarchie

Die folgenden Fragmente der Logik zweiter Stufe stehen in einem wichtigen Zusammenhang zu den Stufen der Polynomialzeit-Hierarchie:

2.10 Definition (Σ^1_k und Π^1_k)**.**
Für jedes $k \in \mathbb{N}$ definieren wir induktiv die Formelklassen Σ^1_k und Π^1_k folgendermaßen:

- $\Sigma^1_0 := \Pi^1_0 :=$ FO.
- $\Sigma^1_{k+1} := \{\, \exists X_1 \cdots \exists X_\ell\, \varphi \;:\; \ell \geqslant 0,\ X_1, \ldots, X_\ell$ sind Relationsvariablen und $\varphi \in \Pi^1_k \,\}$
- $\Pi^1_{k+1} := \{\, \forall X_1 \cdots \forall X_\ell\, \varphi \;:\; \ell \geqslant 0,\ X_1, \ldots, X_\ell$ sind Relationsvariablen und $\varphi \in \Sigma^1_k \,\}$.

2.11 Bemerkungen.

(a) Man beachte, dass $\Sigma_1^1 = \text{ESO}$, und für alle $k \in \mathbb{N}$ gilt:

$$\text{FO} \subseteq \Sigma_k \subseteq \Pi_{k+1} \subseteq \Sigma_{k+2} \subseteq \text{SO}.$$

(b) Σ_k^1-Formeln sind von der Gestalt

$$\exists \vec{Y}_1 \, \forall \vec{Y}_2 \cdots Q \, \vec{Y}_k \, \varphi,$$

wobei φ eine FO-Formel, jedes $\vec{Y}_i$ ein Tupel von Relationsvariablen und

$$Q = \begin{cases} \forall, & \text{falls } k \text{ gerade} \\ \exists, & \text{falls } k \text{ ungerade.} \end{cases}$$

Die einzelnen Tupel $\vec{Y}_i$ heißen (existentielle bzw. universelle) *Quantorenblöcke*.

(c) Man kann leicht sehen, dass es für jede SO-Formel Φ eine Zahl k und eine Formel $\Psi \in \Sigma_k^1$ gibt, so dass Ψ äquivalent zu Φ ist.
Somit ist jede SO-Formel äquivalent zu einer Formel in $\bigcup_{k \in \mathbb{N}} \Sigma_k^1$.

Aus dem Satz von Fagin erhält man leicht die folgende Charakterisierung der Polynomialzeit-Hierarchie:

2.12 Satz.

(a) Für jedes $k \in \mathbb{N}_{\geqslant 1}$ gilt: Σ_k^1 beschreibt Σ_k^p auf Fin.

D.h. ein Problem gehört genau dann zur k-ten Stufe Σ_k^p der Polynomialzeit-Hierarchie, wenn es durch einen Σ_k^1-Satz beschrieben werden kann.

(b) Für jedes $k \in \mathbb{N}_{\geqslant 1}$ gilt: Π_k^1 beschreibt Π_k^p auf Fin.

(c) SO *beschreibt* PH *auf* Fin.

Beweis: *Übung.* □

2.2 Fixpunktlogiken zur Beschreibung von P und PSPACE

Satz von Fagin: Beschreibung von NP durch existentielle Logik zweiter Stufe (ESO).
Ziel dieses Abschnitts: Logiken zur Beschreibung von P und PSPACE
Möglichkeiten:

- P durch geeignete Fragmente von SO beschreiben
  etwa: SO-HORN, Σ_1^1-HORN (Grädel, 1991)
- Fixpunktlogiken: Erweiterungen von FO um die Möglichkeit, Relationen *induktiv* (bzw. rekursiv) zu definieren

Um Fixpunktlogiken einführen und verstehen zu können, benötigen wir zunächst einige grundlegende Begriffe.

2.2.1 Etwas Fixpunkttheorie

Wir schreiben $\mathrm{Pot}(A)$, um die Potenzmenge einer Menge A zu bezeichnen, d.h.

$$\mathrm{Pot}(A) \;=\; \{X \,:\, X \subseteq A\}.$$

Insbesondere gilt natürlich für jedes *endliche* A, dass $|\mathrm{Pot}(A)| \;=\; 2^{|A|}$.

2.13 Definition. Sei A eine Menge und $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ eine Abbildung.

(a) F heißt *monoton*, falls für alle Mengen $P, Q \in \mathrm{Pot}(A)$ gilt:

$$P \subseteq Q \;\implies\; F(P) \subseteq F(Q).$$

(b) F heißt *inflationär*, falls für alle $P \in \mathrm{Pot}(A)$ gilt: $P \subseteq F(P)$.

(c) Eine Menge $P \in \mathrm{Pot}(A)$ heißt *Fixpunkt* von F, falls $F(P) = P$.

(d) Eine Menge $P \in \mathrm{Pot}(A)$ heißt *kleinster Fixpunkt* von F, falls P ein Fixpunkt von F ist und für jeden Fixpunkt Q von F gilt: $P \subseteq Q$.

2.14 Proposition.

(a) Es gibt Abbildungen, die weder monoton noch inflationär sind.

(b) Es gibt Abbildungen, die monoton, aber nicht inflationär sind.

(c) Es gibt Abbildungen, die inflationär, aber nicht monoton sind.

(d) Es gibt Abbildungen, die keinen Fixpunkt besitzen.

(e) Für jede Abbildung $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ gilt:
Falls es einen kleinsten Fixpunkt von F gibt, so ist dieser eindeutig bestimmt.

Beweis: *Übung.* □

Der folgende Satz charakterisiert die kleinsten Fixpunkte *monotoner* Abbildungen:

2.15 Satz (Knaster und Tarski)**.**
Sei A eine Menge und $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ eine monotone Abbildung.
Dann hat F einen kleinsten Fixpunkt, der $\mathbf{lfp}(F)$ genannt wird, und es gilt:[3]

$$\mathbf{lfp}(F) \;=\; \bigcap\{X \subseteq A \,:\, F(X) = X\} \;=\; \bigcap\{X \subseteq A \,:\, F(X) \subseteq X\}.$$

[3] Für eine Menge M, deren Elemente selbst Mengen sind, schreiben wir $\bigcap M$, um die Schnittmenge $\bigcap_{X \in M} X$ zu bezeichnen.

Beweis: Sei

$$M := \{X \subseteq A : F(X) \subseteq X\} \qquad \text{und} \qquad P := \bigcap M.$$

Schritt 1: *Wir zeigen zunächst, dass P ein Fixpunkt von F ist:*

"$F(P) \subseteq P$": Da $P = \bigcap M$, gilt für jedes $X \in M$, dass $P \subseteq X$. Da F monoton ist, folgt $F(P) \subseteq F(X)$. Wegen $X \in M$ gilt $F(X) \subseteq X$. Somit gilt für alle $X \in M$, dass $F(P) \subseteq X$. Daher gilt also:

$$F(P) \subseteq \bigcap M \stackrel{\text{def}}{=} P.$$

"$P \subseteq F(P)$": Da (wie wir gerade gezeigt haben) $F(P) \subseteq P$ ist, folgt aus der Monotonie von F, dass auch $F\big(F(P)\big) \subseteq F(P)$. Somit gilt $F(P) \in M$. Wegen $P = \bigcap M$ folgt daher $P \subseteq F(P)$.

Schritt 2: *Für jeden Fixpunkt Q von F gilt $P \subseteq Q$, denn:*

Für jeden Fixpunkt Q von F gilt $F(Q) \subseteq Q$. Daher ist $Q \in M$, und gemäß Definition von P daher $P \subseteq Q$.

Schritt 3: *Abschluss des Beweises:*

Aus den Schritten 1 und 2 folgt, dass P der kleinste Fixpunkt von F ist.
Außerdem gilt für die Menge

$$M' := \{X \subseteq A : F(X) = X\},$$

dass $P \in M'$ (da $F(P) = P$) und $M' \subseteq M$. Somit folgt

$$P \underset{\text{(Def.)}}{=} \bigcap M \underset{(M \supseteq M')}{\subseteq} \bigcap M' \underset{(P \in M')}{\subseteq} P.$$

Daher gilt für $\mathbf{lfp}(F) := P$, dass

$$\mathbf{lfp}(F) = \bigcap\{X \subseteq A : F(X) = X\} = \bigcap\{X \subseteq A : F(X) \subseteq X\}$$

der kleinste Fixpunkt von F ist. □

Eine andere Charakterisierung der kleinsten Fixpunkte monotoner Funktionen, die auch gleich ein Verfahren zum *Ausrechnen* des kleinsten Fixpunkts liefert, ergibt sich durch die Betrachtung der einzelnen *Induktionsstufen*:

2.16 Definition (Induktionsstufen)**.**
Sei A eine *endliche* Menge und $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ eine Abbildung.

(a) Wir definieren induktiv eine Sequenz von Mengen $R^0, R^1, R^2, \ldots$ durch

$$R^0 := \varnothing \qquad \text{und} \qquad R^{i+1} := F\big(R^i\big) \qquad \text{(für alle } i \in \mathbb{N}\text{)}.$$

Die Menge R^i heißt *i-te Induktionsstufe* (oder auch: i-te Stufe).

(Es gilt also für alle $i \in \mathbb{N}$: $R^i = F^i(\varnothing)$.)

(b) F heißt *induktiv*, falls für alle $i \in \mathbb{N}$ gilt: $R^i \subseteq R^{i+1}$.

2.17 Proposition.

(a) Jede monotone oder inflationäre Abbildung ist induktiv.

(b) Es gibt Abbildungen, die induktiv, aber weder monoton noch inflationär sind.

Beweis: *Übung.* □

2.18 Proposition. *Für jede endliche Menge A und jede induktive Abbildung $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ wird die Sequenz der Induktionsstufen stationär, d.h. es gibt eine Zahl $i \leqslant |A|$, so dass $R^i = R^{i+1} \stackrel{def}{=} F(R^i)$, und somit $R^i = R^{i+n}$ für alle $n \geqslant 0$.*

Beweis: Da F induktiv ist, gilt

$$\varnothing \;=\; R^0 \;\subseteq\; R^1 \;\subseteq\; R^2 \;\subseteq\; \cdots \;\subseteq\; R^j \;\subseteq\; R^{j+1} \;\subseteq\; \cdots \;\subseteq\; A.$$

Da A nur $|A|$ viele Elemente besitzt, kann *nicht* für alle $j \in \{0, 1, \ldots, |A|\}$ gelten, dass $R^j \subsetneq R^{j+1}$ (denn sonst würde in jeder der Stufen $1, 2, \ldots, |A|, |A|+1$ mindestens ein neues Element hinzukommen, in A gibt es aber nur $|A|$ viele Elemente).
Somit muss es also ein $i \in \{0, 1, \ldots, |A|\}$ geben, so dass $R^i = R^{i+1}$. Gemäß der Definition der Stufen ($R^{j+1} := F(R^j)$) folgt daraus, dass

$$R^i \;=\; F(R^i) \;=\; F(F(R^i)) \;=\; F(F(F(R^i))) \;=\; \cdots,$$

also

$$R^i \;=\; R^{i+1} \;=\; R^{i+2} \;=\; R^{i+3} \;=\; \cdots.$$

□

2.19 Definition. Sei A eine endliche Menge und $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ induktiv.

(a) Die kleinste Zahl i, für die $R^i = R^{i+1}$ (d.h. $F^i(\varnothing) = F^{i+1}(\varnothing) = F^{i+n}(\varnothing)$, für alle $n \geqslant 0$), heißt das *Abschlussordinal* von F, oder kurz: $\mathrm{cl}(F)$.

(b) Die Menge $R^\infty := R^{\mathrm{cl}(F)}$ heißt *induktiver Fixpunkt* von F.

2.20 Satz. *Sei A eine endliche Menge.*
Für jede monotone Abbildung $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ gilt:

$$\mathbf{lfp}(F) \;=\; R^\infty.$$

D.h.: Der induktive Fixpunkt einer monotonen Abbildung ist gerade der kleinste Fixpunkt.

Beweis: "$\subseteq$": Gemäß der Definition des inflationären Fixpunkts gilt $R^\infty = F(R^\infty)$, R^∞ ist also ein Fixpunkt von F. Insbesondere gilt

$$\mathbf{lfp}(F) \subseteq R^\infty,$$

da der kleinste Fixpunkt gemäß Definition in jedem anderen Fixpunkt enthalten ist.

"$\supseteq$": Per Induktion zeigen wir, dass $R^i \subseteq \mathbf{lfp}(F)$ für alle $i \in \mathbb{N}$.
$i = 0$: Klar, da $R^0 = \varnothing \subseteq \mathbf{lfp}(F)$.
$i \to i{+}1$: Per Induktion gilt $R^i \subseteq \mathbf{lfp}(F)$. Da F monoton ist, folgt, dass

$$R^{i+1} \;\stackrel{\text{def}}{=}\; F(R^i) \;\subseteq\; F(\mathbf{lfp}(F)) \;=\; \mathbf{lfp}(F).$$

□

Den kleinsten Fixpunkt einer monotonen Funktion kann man leicht berechnen, indem man nach und nach die einzelnen Induktionsstufen berechnet und abbricht, sobald diese stationär werden:

2.21 Proposition. *Ist $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ monoton und in polynomieller Zeit berechenbar, so kann $\mathbf{lfp}(F)$ in polynomieller Zeit berechnet werden.*

Beweis: Es gibt höchstens $|A|$ Induktionsstufen, und jede davon kann in polynomieller Zeit aus der vorangegangenen berechnet werden. □

2.2.2 Die kleinste Fixpunktlogik

2.22 Definition (Operator $F_{\varphi,\mathfrak{A}}$)**.**
Sei σ eine Signatur, $k \in \mathbb{N}_{\geqslant 1}$, R eine k-stellige Relationsvariable und $\vec{x} = x_1, \ldots, x_k$ ein Tupel aus k verschiedenen Variablen erster Stufe.
Jede $\mathrm{FO}[\sigma \dot\cup \{R\}]$-Formel $\varphi(R, \vec{x})$ definert in jeder σ-Struktur $\mathfrak{A}$ eine Abbildung $F_{\varphi,\mathfrak{A}}$ wie folgt:

$$\begin{array}{rccl} F_{\varphi,\mathfrak{A}}: & \mathrm{Pot}(A^k) & \to & \mathrm{Pot}(A^k) \\ & P & \mapsto & \{\vec{a} \in A^k \,:\, \mathfrak{A} \models \varphi[P, \vec{a}]\}. \end{array}$$

Zum Beispiel kann man sich φ als Datenbankanfrage vorstellen und $F_{\varphi,\mathfrak{A}}$ als die Abbildung, die jeder Datenbank-Relation P das Ergebnis der Anfrage zuordnet.

Die im Folgenden definierte *monotone Fixpunktlogik* ist eine Erweiterung der Logik erster

Stufe durch einen Operator, mit dem man aus einer Formel $\varphi(R, \vec{x})$ eine neue Formel erhalten kann, die $[\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t})$ genannt wird, und die in jeder Struktur $\mathfrak{A}$ genau von den Tupeln $\vec{t} \in A^k$ erfüllt wird, die zu dem kleinsten Fixpunkt der Operation $F_{\varphi,\mathfrak{A}}$ gehören. Dafür sollte man natürlich wissen, ob der kleinste Fixpunkt auch wirklich existiert. Gemäß dem Satz von Knaster und Tarski (Satz 2.15) existiert der kleinste Fixpunkt von $F_{\varphi,\mathfrak{A}}$ auf jeden Fall dann, wenn $F_{\varphi,\mathfrak{A}}$ *monoton* ist.

2.23 Definition (Monotone Fixpunktlogik MFP)**.** Sei σ eine Signatur.
Die Formelmenge MFP$[\sigma]$ ist induktiv durch die Regeln (A1), (A2), (A3), (BC) und (Q1) der Logik erster Stufe, sowie die folgende Regel (MFP) definiert:

(MFP) Ist $\varphi(R, \vec{x})$ eine MFP$[\sigma]$-Formel, wobei

- R eine k-stellige Relationsvariable, für ein $k \in \mathbb{N}_{\geqslant 1}$,
- $\vec{x} = x_1, \ldots, x_k$ ein Tupel aus k verschiedenen Variablen erster Stufe, und
- φ außer R und $\vec{x}$ evtl. noch andere freie Variablen hat (etwa $\vec{u}, \vec{S}$),

ist $\vec{t} = t_1, \ldots, t_k$ ein k-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ, und ist $\boldsymbol{F_{\varphi,\mathfrak{A}}}$ ***monoton auf jeder endlichen*** $(\sigma \dot\cup \{\vec{u}, \vec{S}\})$***-Struktur*** $\mathfrak{A}$, so ist

$$[\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t})$$

eine MFP$[\sigma]$-Formel.

2.24 Bemerkungen.

(a) Die Menge der *freien Variablen* einer MFP$[\sigma]$-Formel ist induktiv definiert wie für FO bzw. SO, mit der zusätzlichen Regel

$$\mathit{frei}\big([\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t})\big) \;:=\; \big(\mathit{frei}(\varphi) \setminus \{R, \vec{x}\}\big) \cup \{t_i \,:\, t_i \in \mathrm{Var}_1\}.$$

(b) Gilt $\mathit{frei}(\varphi) = \{R, \vec{x}, \vec{u}, \vec{S}\}$, für ein Tupel $\vec{u}$ von Variablen erster Stufe und ein Tupel $\vec{S}$ von Relationsvariablen, so nennen wir die Variablen in $\vec{u}$ die *Parameter* der Formel $[\mathbf{lfp}_{R,\vec{x}}(\varphi)](\vec{t})$.

2.25 Definition (Semantik von MFP$[\sigma]$-Formeln)**.** Die Semantik von MFP$[\sigma]$-Formeln der Form $\psi := [\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t})$ ist folgendermaßen definiert:
Ist $\mathit{frei}(\psi) = \{\vec{u}, \vec{S}\}$ und ist $\mathfrak{A}$ eine endliche $(\sigma \dot\cup \{\vec{u}, \vec{S}\})$-Struktur, so gilt:

$$\mathfrak{A} \models [\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t}) \quad :\Longleftrightarrow \quad \vec{t}^{\,\mathfrak{A}} \in \mathbf{lfp}(F_{\varphi,\mathfrak{A}}).$$

Aus Satz 1.68 ist bekannt, dass Monotonie von FO-Formeln auf endlichen Strukturen unentscheidbar ist. Es gibt also keinen Algorithmus, der bei Eingabe einer Formel $\varphi(R, \vec{x})$ entscheidet, ob $F_{\varphi,\mathfrak{A}}$ für alle endlichen Strukturen $\mathfrak{A}$ monoton ist.

Daher ist die Syntax von MFP unentscheidbar, d.h. es gibt keinen Algorithmus, der bei Eingabe einer Formel ψ entscheidet, ob diese zu MFP gehört.

Wünschenswert wäre, eine Fixpunktlogik zu defineren, bei der man schon an der Syntax einer Formel erkennt, ob diese zur Logik gehört oder nicht. Ein rein *syntaktisches* Kriterium, das hinreichend für Monotonie ist, wird durch die folgende Definition gegeben:

2.26 Definition. Eine Formel $\varphi(R, \vec{x})$ heißt *positiv in* R (bzw. *negativ in* R), falls Atome der Form $R(\vec{y})$ in φ stets im Bereich einer *geraden* (bzw. *ungeraden*) Anzahl von Negationssymbolen vorkommen und in φ kein Implikationspfeil "$\rightarrow$" und kein Biimplikationspfeil "$\leftrightarrow$" vorkommt.

Man sieht leicht:

2.27 Proposition. *Für jede Formel $\varphi(R, \vec{x})$, die positiv in R ist, gilt:*
$F_{\varphi,\mathfrak{A}}$ *ist* monoton *für alle Strukturen* $\mathfrak{A}$.

Beweisidee: Per Induktion nach dem Formelaufbau zeigt man, dass für jede Formel

$$\varphi(\vec{x}, R_1, \ldots, R_k, S_1, \ldots, S_\ell),$$

die *positiv* in $R_1, \ldots, R_k$ und *negativ* in $S_1, \ldots, S_\ell$ ist, folgendes für alle Strukturen $\mathfrak{A}$, alle Tupel $\vec{a}$ und alle Relationen $R_1^{\mathfrak{A}} \subseteq {R'_1}^{\mathfrak{A}}, \ldots, R_k^{\mathfrak{A}} \subseteq {R'_k}^{\mathfrak{A}}$ und $S_1^{\mathfrak{A}} \supseteq {S'_1}^{\mathfrak{A}}, \ldots, S_\ell^{\mathfrak{A}} \supseteq {S'_\ell}^{\mathfrak{A}}$ über dem Universum von A gilt: Falls $\mathfrak{A} \models \varphi[\vec{a}, \vec{R}^{\mathfrak{A}}, \vec{S}^{\mathfrak{A}}]$, so auch $\mathfrak{A} \models \varphi[\vec{a}, \vec{R'}^{\mathfrak{A}}, \vec{S'}^{\mathfrak{A}}]$.
Details: *Übung*. □

Die *kleinste Fixpunktlogik* ist analog zur monotonen Fixpunktlogik definiert, mit dem Unterschied, dass man jetzt an Stelle der Monotonie fordert, dass Formeln $\varphi(R, \vec{x})$, auf die der Fixpunktoperator $\mathbf{lfp}(\cdot)$ angewandt werden soll, *positiv* in R sind.

2.28 Definition (Kleinste Fixpunktlogik LFP)**.** Sei σ eine Signatur.
Die Formelmenge LFP$[\sigma]$ ist induktiv durch die Regeln (A1),(A2),(A3),(BC) und (Q1) der Logik erster Stufe, sowie die folgende Regel (LFP) definiert:

(LFP) Ist $\varphi(R, \vec{x})$ eine LFP$[\sigma]$-Formel, wobei

- R eine k-stellige Relationsvariable, für ein $k \in \mathbb{N}_{\geqslant 1}$,
- $\vec{x} = x_1, \ldots, x_k$ ein Tupel aus k verschiedenen Variablen erster Stufe, und
- φ außer R und $\vec{x}$ evtl. noch andere freie Variablen hat,

ist $\vec{t} = t_1, \ldots, t_k$ ein k-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ, und ist $\boldsymbol{\varphi}$ ***positiv in*** $\boldsymbol{R}$, so ist

$$[\mathbf{lfp}_{R,\vec{x}}\, \varphi](\vec{t})$$

eine LFP$[\sigma]$-Formel.

Die *Semantik* der LFP-Formeln ist genauso definiert wie die Semantik von MFP.

2.29 Beispiele. (a) *Erreichbarkeit:*

Sei $G = (V, E, s, t)$ ein Graph mit zwei ausgezeichneten Knoten s, t. Dann gilt

$$G \models \big[\mathbf{lfp}_{R,x}\big(x = s \vee \exists z\,(R(z) \wedge E(z,x))\big)\big](t)$$

genau dann, wenn es in G einen Pfad von s nach t gibt.

Dies sieht man, indem man die einzelnen Stufen $R^0, R^1, R^2, \ldots$ des Operators $F_{\varphi,G}$ ausrechnet. Per Induktion nach i erhält man so für obige Beispielformel:

$$R^i = \{v \in V : \text{es gibt in } G \text{ einen Pfad der Länge} < i \text{ von } s \text{ nach } v\}.$$

Daher gilt für $\mathbf{lfp}(F_{\varphi,G}) = R^\infty$:

$$R^\infty = \{v \in V : \text{es gibt in } G \text{ einen Pfad von } s \text{ nach } v\}.$$

(b) *Graphzusammenhang:*

Sei $G = (V, E)$ ein Graph. Dann gilt

$$G \models \forall x\,\forall y\,\big[\mathbf{lfp}_{R,x,y}\, x = y \vee \exists z\,\big(R(x,z) \wedge E(z,y)\big)\big](x,y)$$

genau dann, wenn G zusammenhängend ist.
Für die einzelnen Stufen gilt hier in jedem Graphen G:

$$R^i = \{(u,v) \in V^2 : \text{es gibt in } G \text{ einen Pfad der Länge} < i \text{ von } u \text{ nach } v\}.$$

(c) *Eine definierbare Wortsprache:*

Sei $\sigma = \{<, P_a, P_b\}$ die Signatur, mit der Wörter $w \in \{a,b\}^*$ als σ-Strukturen $\mathfrak{A}_w$ kodiert werden (siehe Übungsblatt 1).
Wir definieren nun eine Formel $\varphi(R,x)$, die positiv in R ist, so dass für alle nicht-leeren Worte $w = w_0 \cdots w_{n-1} \in \{a,b\}^*$ gilt:

$\mathfrak{A}_w \models [\mathbf{lfp}_{R,x}\,\varphi](i) \iff$ an Position i steht in w der Buchstabe a und auf den Positionen $0, \ldots, i$ steht eine *ungerade* Anzahl von as.

Wenn wir φ so gewählt haben, gilt für jedes Wort w:

$$\mathfrak{A}_w \models \forall x\Big(\underbrace{\big(P_a(x) \wedge \neg\exists y\,(P_a(y) \wedge x < y)\big)}_{x \text{ ist die Position des letzten } a\text{s in } w} \rightarrow \neg[\mathbf{lfp}_{R,x}\,\varphi](x)\Big)$$

genau dann, wenn die Anzahl der as in w gerade ist.

Wahl der Formel $\varphi(R,x)$:

$$\begin{aligned}\varphi(R,x) \;:=\; & \Big(P_a(x) \wedge \neg\exists y\,(P_a(y) \wedge y < x)\Big) \vee \\ & \Big(\exists z\,\exists y\,\big(\,R(z) \wedge P_a(y) \wedge P_a(x) \wedge (z<y<x) \wedge \\ & \qquad \forall u\,\big(\big(\neg(z<u<y \vee y<u<x)\big) \vee P_b(u)\big)\,\big)\Big).\end{aligned}$$

2.30 Lemma.
Die Datenkomplexität des Auswertungsproblems für LFP *auf* Fin *liegt in* PTIME.

Beweis: Per Induktion über den Formelaufbau. Der einzige interessante Fall ist der Fixpunktoperator

$$\psi \;:=\; [\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t}).$$

Nach Induktionsannahme kann die Formel φ für jede Induktionsstufe R^i in polynomieller Zeit ausgewertet werden. Die Behauptung folgt damit aus Proposition 2.21. □

2.2.3 Inflationäre Fixpunktlogik

Bei der Definition der *kleinsten Fixpunktlogik* wurde durch ein syntaktisches Kriterium (nämlich dadurch, dass die Formel $\varphi(R,\vec{x})$ *positiv* in R sein muss), gewährleistet, dass die Sequenz $R^0, R^1, R^2, \ldots$ der Induktionsstufen *induktiv* ist, d.h.

$$R^0 \;\subseteq\; R^1 \;\subseteq\; R^2 \;\subseteq\; \cdots \;\subseteq\; R^i \subseteq\; R^{i+1} \;\subseteq\; \cdots,$$

und daher der induktive Fixpunkt $R^\infty = R^{\mathrm{cl}(F)}$ existiert.
In diesem Abschnitt werden wir die Bedingung "$\varphi(R,\vec{x})$ positiv in R" fallenlassen und stattdessen durch explizites Vereinigen erzwingen, dass jede Induktionsstufe ihre gesamte Vorgängerstufe enthält.

2.31 Definition (Inflationärer Fixpunkt $\mathbf{ifp}(F)$)**.** Zu jeder endlichen Menge A und jeder Abbildung $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ ist die Abbildung I_F wie folgt definiert:

$$\begin{array}{rccl} I_F \;:\; \mathrm{Pot}(A) & \to & \mathrm{Pot}(A) \\ X & \mapsto & X \cup F(X). \end{array}$$

Offensichtlich ist I_F *inflationär* und hat daher einen induktiven Fixpunkt $R^\infty = R^{\mathrm{cl}(I_F)}$.
R^∞ heißt der *inflationäre Fixpunkt von* F, geschrieben $\mathbf{ifp}(F)$.

2.32 Bemerkung. Falls F *monoton* ist, so haben F und I_F die gleichen Induktionsstufen, und es gilt $\mathbf{ifp}(F) = \mathbf{lfp}(F)$.

2.33 Definition (Inflationäre Fixpunktlogik IFP)**.** Sei σ eine Signatur.
Die Formelmenge $\mathrm{IFP}[\sigma]$ ist induktiv durch die Regeln (A1),(A2),(A3),(BC) und (Q1) der Logik erster Stufe, sowie die folgende Regel (IFP) definiert:

(IFP) Ist $\varphi(R, \vec{x})$ eine IFP$[\sigma]$-Formel, wobei

- R eine k-stellige Relationsvariable, für ein $k \in \mathbb{N}_{\geqslant 1}$,
- $\vec{x} = x_1, .., x_k$ ein Tupel aus k verschiedenen Variablen erster Stufe, und
- φ außer R und $\vec{x}$ evtl. noch andere freie Variablen hat,

und ist $\vec{t} = t_1, .., t_k$ ein k-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ, so ist

$$[\mathbf{ifp}_{R,\vec{x}}\, \varphi](\vec{t})$$

eine IFP$[\sigma]$-Formel.

2.34 Definition (Semantik von IFP$[\sigma]$-Formeln)**.** Die Semantik von IFP$[\sigma]$-Formeln der Form $\psi := [\mathbf{ifp}_{R,\vec{x}}\, \varphi](\vec{t})$ ist folgendermaßen definiert:
Ist *frei*$(\psi) = \{\vec{u}, \vec{S}\}$ und ist $\mathfrak{A}$ eine endliche $(\sigma \dot\cup \{\vec{u}, \vec{S}\})$-Struktur, so gilt:

$$\mathfrak{A} \models [\mathbf{ifp}_{R,\vec{x}}\, \varphi](\vec{t}) \quad :\Longleftrightarrow \quad \vec{t}^{\,\mathfrak{A}} \in \mathbf{ifp}(F_{\varphi,\mathfrak{A}}).$$

2.35 Beispiel. Sei $\sigma = \{<, P_a, P_b\}$ die Signatur, mit der Wörter $w \in \{a, b\}^*$ als σ-Strukturen $\mathfrak{A}_w$ kodiert werden (siehe Übungsblatt 1).
Wir definieren eine Formel $\varphi(R, x)$, so dass für alle nicht-leeren Worte $w = w_0 \cdots w_{n-1} \in \{a, b\}^*$ gilt:

$$\mathfrak{A}_w \models\ \forall u\, [\mathbf{ifp}_{R,x}\, \varphi](u) \quad \Longleftrightarrow \quad w \in \{a^n b^n\ :\ n \geqslant 1\}.$$

Wir wählen $\varphi(R, x)$:=

$$\Big(\text{“}P_a(0)\text{”} \wedge \text{“}P_b(\textit{max})\text{”} \wedge \big(\text{“}x = 0\text{”} \vee \text{“}x = \textit{max}\text{”}\big)\Big) \ \vee$$
$$\exists y\, \exists z \Big(\ y < z \ \wedge\ R(y) \ \wedge\ R(z) \ \wedge\ \forall v \big(\text{“}y < v < z\text{”} \rightarrow \neg R(v)\big) \wedge$$
$$\text{“}P_a(y{+}1)\text{”} \wedge \text{“}P_b(z{-}1)\text{”} \wedge \big(\text{“}x = y{+}1\text{”} \vee \text{“}x = z{-}1\text{”}\big)\Big).$$

Für die i-te Induktionsstufe R^i des *inflationären* Fixpunkts von F gilt dann:

$$R^i \ = \ \{0, .., j, n{-}1, .., n{-}1{-}j\ :\ j < i \text{ maximal, so dass } w \in a^j\{a, b\}^* b^j\}.$$

Man beachte, dass $\varphi(R, x)$ *nicht positiv* in R ist, dass “$\forall u\, [\mathbf{lfp}_{R,x}\, \varphi](u)$” also *keine* LFP-Formel ist.

2.36 Lemma.
Die Datenkomplexität des Auswertungsproblems für IFP *auf* Fin *liegt in* PTIME.

Beweis: Analog zum Beweis von Lemma 2.30 für LFP. □

2.37 Proposition. *Jede* LFP*-Formel ist äquivalent zu einer* IFP*-Formel (kurz:* LFP $\leqslant$ IFP*).*

Beweis: Per Induktion über den Formelaufbau. Der einzige interessante Fall ist der Fixpunktoperator

$$[\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t}),$$

wobei $\varphi(R,\vec{x})$ eine LFP-Formel ist, die *positiv* in R ist. Insbesondere ist die Abbildung $F_{\varphi,\mathfrak{A}}$ *monoton* für alle Strukturen $\mathfrak{A}$. Gemäß Induktionsannahme gibt es eine IFP-Formel $\varphi'(R,\vec{x})$, die äquivalent zu $\varphi(R,\vec{x})$ ist. Insbesondere gilt $F_{\varphi',\mathfrak{A}} = F_{\varphi,\mathfrak{A}}$ für alle Strukturen $\mathfrak{A}$. Aus Bemerkung 2.32 folgt direkt, dass die Formel

$$[\mathbf{ifp}_{R,\vec{x}}\,\varphi'](\vec{t})$$

äquivalent zur Formel $[\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t})$ ist. □

2.2.4 Partielle Fixpunktlogik

Von den Lemmas 2.30 und 2.36 wissen wir, dass jedes Problem, das durch eine LFP-Formel oder eine IFP-Formel beschrieben werden kann, zur Komplexitätsklasse PTIME gehört. Zur Beschreibung von Problemen, deren Komplexität jenseits von PTIME liegt, eignen sich die Logiken LFP und IFP also nicht.

Um eine Logik größerer Ausdrucksstärke zu erhalten, beschränken wir im Folgenden die Aufmerksamkeit nicht mehr nur auf *induktive* Abbildungen F bzw. I_F, sondern betrachten *beliebige* Abbildungen $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$. Für diese bildet die Sequenz $R^0, R^1, R^2, \ldots$ der Induktionsstufen nicht mehr unbedingt eine aufsteigende Kette, und sie wird auch nicht immer stationär, erreicht also nicht notwendigerweise einen Fixpunkt.

2.38 Definition (Partieller Fixpunkt $\mathbf{pfp}(F)$)**.** Sei A eine Menge und $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ eine beliebige Abbildung. Der *partielle Fixpunkt* $\mathbf{pfp}(F)$ ist definiert als

$$\mathbf{pfp}(F) := \begin{cases} R^i, & \text{falls es ein } i \in \mathbb{N} \text{ gibt, so dass } R^i = R^{i+1}, \\ \varnothing, & \text{sonst} \end{cases}$$

2.39 Bemerkung. Jede partielle Fixpunktinduktion über einer n-elementigen Menge A kann höchstens $2^n = |\mathrm{Pot}(A)|$ viele Induktionschritte durchlaufen, bevor entweder ein Fixpunkt erreicht oder die Induktion zyklisch wird. Es gilt:

(1) Falls $R^{2^n-1} = R^{2^n}$, so $\mathbf{pfp}(F) = R^{2^n}$.

(2) Falls $R^{2^n-1} \neq R^{2^n}$, so $\mathbf{pfp}(F) = \varnothing$.

Beweis: Es gibt nur 2^n verschiedene Teilmengen von A. Somit gibt es $i < j$ mit $i, j \in \{0, \ldots, 2^n\}$, so dass $R^i = R^j$.
Falls $R^i = R^{i+1}$, so ist $\mathbf{pfp}(F) = R^i = R^{i+1} = R^{2^n-1} = R^{2^n}$.
Falls $R^i \neq R^{i+1}$, so ist die Folge $R^0, R^1, R^2, \ldots$ der Iterationsstufen von der Form

$$R^0, R^1, \ldots, R^{i-1}, \Big(\overbrace{\underbrace{R^i}_{\neq}, R^{i+1}}^{=R^j}, \ldots, R^{j-1}\Big)^*.$$

Daher existiert kein $k \in \mathbb{N}$ mit $R^k = R^{k+1}$. Insbesondere gilt: $R^{2^n-1} \neq R^{2^n}$ und $\mathbf{pfp}(F) = \varnothing$. □

2.40 Bemerkung. Für *induktive* Abbildungen $F : \mathrm{Pot}(A) \to \mathrm{Pot}(A)$ existiert der partielle Fixpunkt und es gilt: $\mathbf{pfp}(F) = \mathbf{ifp}(F)$.

2.41 Beispiel. Im Folgenden konstruieren wir eine Formel $\varphi(R, x)$, so dass für jede linear geordnete endliche Menge $\mathfrak{A} = (A, <^{\mathfrak{A}})$ gilt:

(1) Für jede Teilmenge $X \subseteq A$ gibt es ein $i < 2^{|A|}$, so dass die i-te Induktionsstufe R^i von $F_{\varphi,\mathfrak{A}}$ genau die Menge X ist, und

(2) $\mathbf{pfp}(F_{\varphi,\mathfrak{A}}) = R^{2^{|A|}} = A$.

Die Induktionsstufen $R^0, R^1, R^2, \ldots$ durchlaufen also sämtliche Teilmengen von A und enden schließlich mit der Menge A als partiellem Fixpunkt.

Idee zur Konstruktion von $\varphi(R, x)$:
Sei $A = \{a_0 <^{\mathfrak{A}} \cdots <^{\mathfrak{A}} a_{n-1}\}$. Eine Menge $X \subseteq A$ kodiert ein 0-1-Wort $w_X = w_{n-1} \cdots w_0$ — bzw. die Zahl $z_X := \sum_{i<n} w_i \cdot 2^i \in \{0, \ldots, 2^n-1\}$ — via

$$w_i = 1 \iff a_i \in X.$$

Die Formel $\varphi(R, x)$ zählt mit ihren Induktionsstufen alle (Binärdarstellungen von) Zahlen aus $\{0, \ldots, 2^n-1\}$ der Reihe nach auf. Wir wählen dazu $\varphi(R, x) :=$

$$\big(\forall z\, R(z)\big) \vee \exists y \Big(\neg R(y) \wedge \forall z \big(z < y \to R(z)\big) \wedge \big(x = y \vee (x > y \wedge R(x))\big)\Big).$$

Durch Betrachten der Binärdarstellungen sieht man leicht, dass

- $R^0 = \varnothing$,
- für alle $i < 2^n-1$ gilt: $z_{R^{i+1}} = z_{R^i} + 1$, und
- für $i = 2^n-1$ gilt: $R^i = A = R^{i+1}$.

2.42 Definition (Partielle Fixpunktlogik PFP)**.** Sei σ eine Signatur.
Die Formelmenge PFP$[\sigma]$ ist induktiv durch die Regeln (A1),(A2),(A3),(BC) und (Q1) der Logik erster Stufe, sowie die folgende Regel (PFP) definiert:

(PFP) Ist $\varphi(R, \vec{x})$ eine PFP$[\sigma]$-Formel, wobei

- R eine k-stellige Relationsvariable, für ein $k \in \mathbb{N}_{\geqslant 1}$,
- $\vec{x} = x_1, \ldots, x_k$ ein Tupel aus k verschiedenen Variablen erster Stufe, und
- φ außer R und $\vec{x}$ evtl. noch andere freie Variablen hat,

und ist $\vec{t} = t_1, .., t_k$ ein k-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ, so ist

$$[\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{t})$$

eine PFP$[\sigma]$-Formel.

2.43 Definition (Semantik von PFP$[\sigma]$-Formeln)**.** Die Semantik von PFP$[\sigma]$-Formeln der Form $\psi := [\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{t})$ ist folgendermaßen definiert:
Ist *frei*$(\psi) = \{\vec{u}, \vec{S}\}$ und ist $\mathfrak{A}$ eine endliche $(\sigma \dot\cup \{\vec{u}, \vec{S}\})$-Struktur, so gilt:

$$\mathfrak{A} \models [\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{t}) \quad :\Longleftrightarrow \quad \vec{t}^{\,\mathfrak{A}} \in \mathbf{pfp}(F_{\varphi,\mathfrak{A}}).$$

2.44 Lemma.
Die Datenkomplexität des Auswertungsproblems für PFP *auf* Fin *liegt in* PSPACE*.*

Beweis: Per Induktion über den Formelaufbau. Der einzige interessante Fall ist der Fixpunktoperator

$$[\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{t}).$$

Gemäß Induktionsannahme kann für jede Induktionsstufe R^i die Formel $\varphi(R, \vec{x})$ in einer endlichen Struktur $\mathfrak{A}$ auf Platz polynomiell in $|A|$ ausgewertet werden.
Jede Induktionsstufe R^i ist eine Teilmenge von A^k (mit $k := ar(R)$) und kann somit auf Platz polynomiell in $|A|$ gespeichert werden.
Um R^{i+1} aus R^i zu berechnen, braucht man *nur 2 Stufen* zu speichern (nämlich R^i und R^{i+1}). Sind R^i und R^{i+1} berechnet und gespeichert, so kann man ohne zusätzlichen Platzaufwand testen, ob $R^i = R^{i+1}$, der partielle Fixpunkt also erreicht ist. Wenn bei $i = 2^{(|A|^k)}$ immer noch kein partieller Fixpunkt erreicht wurde, so muss ein Zyklus eingetreten sein, und man weiß, dass $\mathbf{pfp}(F_{\varphi,\mathfrak{A}}) = \varnothing$ ist.
Insgesamt kann man die Formel $[\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{t})$ also auf polynomiellem Platz auswerten. □

2.45 Proposition. *Jede* IFP*-Formel ist äquivalent zu einer* PFP*-Formel (kurz:* IFP $\leqslant$ PFP*).*

Beweis: Per Induktion über den Formelaufbau. Der einzige interessante Fall ist der Fixpunktoperator

$$[\mathbf{ifp}_{R,\vec{x}}\,\varphi](\vec{t}).$$

Für inflationäre Abbildungen existiert der partielle Fixpunkt immer und ist identisch mit dem inflationären Fixpunkt. Es gilt daher:

$$[\mathbf{ifp}_{R,\vec{x}}\,\varphi](\vec{t}) \quad \text{ist äquivalent zu} \quad [\mathbf{pfp}_{R,\vec{x}}\,\big(R(\vec{x}) \vee \varphi\big)](\vec{t}).$$

□

Zusammenfassung:

Für die drei Fixpunktlogiken LFP (kleinste Fixpunktlogik), IFP (inflationäre Fixpunktlogik) und PFP (partielle Fixpunktlogik) gilt:

$$\text{LFP} \leqslant \text{IFP} \leqslant \text{PFP}$$

(d.h.: PFP ist mindestens so ausdrucksstark wie IFP, und IFP ist mindestens so ausdrucksstark wie LFP).

2.2.5 Fixpunktlogiken und Komplexitätsklassen

Ähnlich zum Beweis des Satzes von Fagin kann man zeigen, dass die Logiken LFP und IFP die Komplexitätsklasse PTIME auf der Klasse aller endlichen geordneten Strukturen bescheiben, und dass die Logik PFP die Komplexitätsklasse PSPACE auf der Klasse aller endlichen geordneten Strukturen beschreibt:

2.46 Theorem (Satz von Immerman und Vardi, 1982)**.**

(a) LFP *beschreibt* PTIME *auf* FinOrd.

(b) IFP *beschreibt* PTIME *auf* FinOrd.

Beweis: Wegen Lemma 2.36 und Proposition 2.37 folgt *(b)* direkt aus *(a)*. Von Lemma 2.30 wissen wir, dass die Datenkomplexität des Auswertungsproblems für LFP auf FinOrd in PTIME liegt. Im Folgenden brauchen wir also nur noch zu beweisen, dass jedes Problem aus PTIME durch eine LFP-Formel beschrieben werden kann. Sei dazu σ eine Signatur, die u.a. ein 2-stelliges Relationssymbol $<$ enthält, und sei $\mathsf{C} \subseteq \mathsf{FinOrd}$ eine Klasse endlicher geordneter σ-Strukturen, so dass

$$L_{\mathsf{C}} := \{enc(\mathfrak{A}) : \mathfrak{A} \in \mathsf{C}\} \in \text{PTIME}.$$

D.h. es gibt eine deterministische Turing-Maschine $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ und eine Konstante $k \in \mathbb{N}$, so dass M bei Eingabe (der Kodierung) einer endlichen geordneten σ-Struktur $\mathfrak{A}$ entscheidet, ob $\mathfrak{A} \in \mathsf{C}$ und dabei weniger als n^k Schritte macht. Wie beim Beweis des Satzes von Fagin bezeichnen wir mit n immer die Größe des Universums der Eingabe-Struktur $\mathfrak{A}$ und nehmen o.B.d.A. an, dass

- F_{akz} aus genau einem akzeptierenden Zustand q_{akz} besteht,
- jeder Lauf von M bei jeder Eingabe-Struktur $\mathfrak{A}$ mit $|A| \geqslant 2$ nach *genau* $n^k - 1$ Schritten in einem (akzeptierenden oder verwerfenden) Endzustand endet,
- $n^k \geqslant |enc(\mathfrak{A})|$,

- $Q = \{0, 1, .., s_Q\}$ für ein $s_Q \in \mathbb{N}$, Endzustand $q_{\text{akz}} = 0$, Anfangszustand $q_0 = 1$,
- $\Gamma = \{0, 1, 2, .., s_\Gamma\}$ für ein $s_\Gamma \in \mathbb{N}$, Blank-Symbol $\square = 2$,
- $s := \max\{s_Q, s_\Gamma\}$.

Da M *deterministisch* ist, können wir die Überführungsrelation Δ als Abbildung

$$\delta \;:\; (Q \setminus F) \times \Gamma \;\to\; Q \times \Gamma \times \{-1, 0, 1\}$$

darstellen.

Unser Ziel ist, einen LFP$[\sigma]$-Satz ψ zu finden, so dass $\mathsf{C} = \text{Mod}_{\mathsf{FinOrd}}(\psi)$, d.h. für jede endliche geordnete σ-Struktur $\mathfrak{A}$ gilt:

$$\begin{aligned} \mathfrak{A} \models \psi \quad &\iff \quad \mathfrak{A} \in \mathsf{C} \\ &\iff \quad M \text{ akzeptiert } \mathfrak{A} \\ &\iff \quad \text{der Lauf von } M \text{ bei Eingabe } \mathit{enc}(\mathfrak{A}), \text{ endet nach } n^k{-}1 \\ &\qquad\quad \text{Schritten in Zustand } q_{\text{akz}}. \end{aligned}$$

Idee zur Konstruktion von ψ: Ähnlich wie im Beweis des Satzes von Fagin werden Zeitpunkte t und Bandpositionen p in $\{0, 1, .., n^k{-}1\}$ durch k-Tupel $\vec{t} = (t_1, .., t_k)$ bzw. $\vec{p} = (p_1, .., p_k)$ über $\{0, .., n{-}1\}$ kodiert. Unter Verwendung der linearen Ordnung $<^{\mathfrak{A}}$ kann man das Universum A mit der Menge $\{0, .., n{-}1\}$ identifizieren und k-Tupel $\vec{a} \in A^k$ mit Zahlen aus $\{0, 1, .., n^k{-}1\}$.
Jeden Zustand $q \in Q = \{0, 1, .., s_Q\}$ und jedes Symbol $\gamma \in \Gamma \in \{0, 1, .., s_\Gamma\}$ werden wir in einer Struktur $\mathfrak{A}$ durch das q-größte bzw. das γ-größte Element im Universum A repräsentieren.[4]
Um die Notation etwas übersichtlicher zu machen, nehmen wir o.B.d.A. an, dass σ zwei Konstantensymbole 0 und *max* enthält, die in geordneten σ-Strukturen stets mit dem kleinsten bzw. dem größten Element der linearen Ordnung interpretiert werden.

Der Lauf der DTM M bei Eingabe $\mathit{enc}(\mathfrak{A})$ wird durch folgende $(2k + 2)$-stellige Relation $R^{\mathfrak{A}} \subseteq A^{2k+2}$ repräsentiert:

- $(0, \vec{t}, \vec{p}, 0) \in R^{\mathfrak{A}} \iff$ der Schreib-/Lesekopf steht zum Zeitpunkt $\vec{t}$ auf Bandposition $\vec{p}$.
- $(1, \vec{t}, \vec{p}, \gamma) \in R^{\mathfrak{A}} \iff$ auf Bandposition $\vec{p}$ steht zum Zeitpunkt $\vec{t}$ das Symbol γ.
- $(\mathit{max}, \vec{t}, \vec{max}, q) \in R^{\mathfrak{A}} \iff M$ ist zum Zeitpunkt $\vec{t}$ in Zustand q.

[4] Der LFP-Satz ψ, den wir im Folgenden konstruieren, wird daher nur für solche Strukturen $\mathfrak{A}$ korrekt sein, die mehr als $s = \max\{s_Q, s_\Gamma\}$ Elemente besitzen. Das ist aber nicht weiter schlimm, denn es gibt nur *endlich* viele verschiedene σ-Strukturen mit $\leqslant s$ Elementen, und diese kann man explizit durch eine weitere FO-Formel abhandeln.

Wir konstruieren im Folgenden eine $\text{FO}[\sigma\dot\cup\{R\}]$-Formel $\varphi(R,u,\vec t,\vec p,z)$, die schrittweise die Relation $R^{\mathfrak{A}}$ so aufbaut, dass $R^{\mathfrak{A}}$ durch "$[\mathbf{lfp}_{R,u,\vec t,\vec p,z}\,\varphi]$" beschrieben wird. Die einzelnen Induktionsstufen entsprechen dabei den Zeitpunkten während der Berechnung von M, d.h.:

$$\begin{aligned}
R^0 &= \varnothing,\\
R^1 &= \text{"Startkonfiguration bei Eingabe } \mathit{enc}(\mathfrak{A})\text{"}\\
&= \{(u,\vec t,\vec p,z)\in R^{\mathfrak{A}} \,:\, \mathit{rg}_{<^{\mathfrak{A}}_{lex}}(\vec t)=0\},\\
&\vdots\\
R^{i+1} &= \{(u,\vec t,\vec p,z)\in R^{\mathfrak{A}} \,:\, \mathit{rg}_{<^{\mathfrak{A}}_{lex}}(\vec t)\leqslant i\}.
\end{aligned}$$

Wenn φ so konstruiert ist, dann gilt (beachte dazu, dass $q_{\text{akz}}=0$):

$$M \text{ akzeptiert } \mathfrak{A} \iff \mathfrak{A} \models [\mathbf{lfp}_{R,u,\vec t,\vec p,z}\,\varphi](\mathit{max},\vec{\mathit{max}},\vec{\mathit{max}},0).$$

Die Formel φ ist von der Form

$$\begin{aligned}
&\varphi(R,u,\vec t,\vec p,z) :=\\
&\exists z_0\cdots\exists z_s\,\Big(\varphi_{\mathit{Zahlen}}(z_0,..,z_s)\ \wedge\ \Big(\big(\text{"}\vec t=\vec 0\text{"}\ \wedge\ \varphi_{\mathit{Start}}(u,\vec p,z)\big)\ \vee\ \varphi_{\mathit{Schritt}}(R,u,\vec t,\vec p,z)\Big)\Big),
\end{aligned}$$

wobei $\varphi_{\mathit{Zahlen}}(z_0,..,z_s)$ eine $\text{FO}[<]$-Formel ist, die besagt, dass die Variablen $z_0,..,z_s$ mit den $s{+}1$ kleinsten Elementen aus dem Universum der jeweiligen Struktur belegt werden.

Um die Startkonfiguration von M bei Eingabe $\mathit{enc}(\mathfrak{A})$ durch die Formel φ_{Start} zu beschreiben, benutzen wir die Formeln $\zeta_0(\vec p)$ und $\zeta_1(\vec p)$ aus dem Beweis des Satzes von Fagin (Theorem 2.8), die besagen, dass in dem 0-1-Wort $\mathit{enc}(\mathfrak{A})$ an Position $\mathit{rg}^{\mathfrak{A}}_{<_{lex}}(\vec p)$ das Symbol 0 bzw. das Symbol 1 steht. Wir wählen

$$\begin{aligned}
&\varphi_{\mathit{Start}}(u,\vec p,z) :=\\
&\quad\big(\ u=0\ \wedge\ \text{"}\vec p=\vec 0\text{"}\ \wedge\ z=0\big) && \text{(Kopf auf Position 0)}\\
&\vee\big(u=\mathit{max}\ \wedge\ \text{"}\vec p=\vec{\mathit{max}}\text{"}\ \wedge\ z=z_1\big) && (M \text{ ist im Startzustand } q_0=1)\\
&\vee\big(\ u=z_1\ \wedge\ \Big(\big(z=0\ \wedge\zeta_0(\vec p)\big)\vee && \text{(Bandbeschriftung: } \mathit{enc}(\mathfrak{A}))\\
&\qquad\qquad\big(z=z_1\wedge\zeta_1(\vec p)\big)\vee\\
&\qquad\qquad\big(z=z_2\ \wedge\neg(\zeta_0(\vec p)\vee\zeta_1(\vec p))\big)\Big)\ \big) && \text{(Beachte: Blank-Symbol } \square=2)
\end{aligned}$$

Für die Formel $\varphi_{\mathit{Schritt}}$ sei $\chi_m(\vec p\,',\vec p)$ für $m\in\{-1,0,1\}$ eine Formel, die besagt "$\vec p=\vec p\,'{+}m$", d.h.:

$$\chi_1(\vec p\,',\vec p) := \mathit{succ}_{<_{lex}}(\vec p\,',\vec p),\quad \chi_{-1}(\vec p\,',\vec p) := \mathit{succ}_{<_{lex}}(\vec p,\vec p\,'),\quad \chi_0(\vec p\,',\vec p) := \text{"}\vec p\,'=\vec p\text{"}.$$

Wir wählen

$$\varphi_{\mathit{Schritt}}(R,u,\vec{t},\vec{p},z) \;:=$$

$$\exists\vec{t}\,'\,\exists\vec{p}\,'\,\Big(\mathit{succ}_{<_{\mathit{lex}}}(\vec{t}\,',\vec{t}) \;\wedge\; R(0,\vec{t}\,',\vec{p}\,',0)\wedge \qquad \text{(Kopfpos. } \vec{p}\,' \text{ zum Zeitpkt. } \vec{t}\,' := \vec{t}-1)$$

$$\bigvee_{\substack{q'\in Q\setminus F,\gamma'\in\Gamma,\\ (q,\gamma,m):=\delta(q',\gamma')}} \Big(R(\mathit{max},\vec{t}\,',\vec{\mathit{max}},z_{q'}) \;\wedge\; R(z_1,\vec{t}\,',\vec{p}\,',z_{\gamma'}) \;\wedge \qquad \text{(zum Zeitpkt. } \vec{t}\,'\text{: in Zustand } q', \text{ liest Symbol } \gamma')$$

$$\Big(\quad \big(u=\mathit{max}\;\wedge\;\vec{p}=\vec{\mathit{max}}\;\wedge\;z=z_q\big) \qquad \text{(zum Zeitpkt. } \vec{t} \text{ in Zustand } q)$$

$$\vee\;\big(u=0\;\wedge\;\chi_m(\vec{p}\,',\vec{p})\;\wedge\;z=0\big) \qquad \text{(zum Zeitpkt. } \vec{t} \text{ Kopfpos. } \vec{p}\,'+m)$$

$$\vee\;\big(u=z_1\;\wedge\;\text{“}\vec{p}\,'=\vec{p}\text{”}\;\wedge\;z=z_\gamma\big) \qquad \text{(zum Zeitpkt. } \vec{t} \text{ Symbol } \gamma \text{ auf Pos. } \vec{p}\,')$$

$$\vee\;\big(u=z_1\;\wedge\;\text{“}\vec{p}\,'\neq\vec{p}\text{”}\;\wedge\;R(z_1,\vec{t}\,',\vec{p},z)\big)\Big)\Big)\Big) \qquad \text{(auf Pos. } \vec{p}\neq\vec{p}\,' \text{ zum Zeitpkt. } \vec{t} \text{ gleiches Symbol wie zum Zeitpkt. } \vec{t}\,')$$

Man beachte, dass diese Formel *positiv* in R ist. Daher ist die Formel

$$\psi \;:=\; [\mathbf{lfp}_{R,u,\vec{t},\vec{p},z}\,\varphi](\mathit{max},\vec{\mathit{max}},\vec{\mathit{max}},0)$$

eine LFP$[\sigma]$-Formel.
Per Induktion nach i kann man leicht für alle endlichen geordneten σ-Strukturen $\mathfrak{A}$ und alle $i\in\mathbb{N}$ zeigen, dass die $(i{+}1)$-te Induktionsstufe R^{i+1} des Operators $F_{\varphi,\mathfrak{A}}$ aus genau den Tupeln $(u,\vec{t},\vec{p},z)\in A^{2k+2}$ besteht, für die gilt: $j := \mathit{rg}_{<^{\mathfrak{A}}_{\mathit{lex}}}(\vec{t}) \leqslant i$ und

$$\{(u',\vec{t}\,',\vec{p}\,',z')\in R^{i+1} \;:\; \vec{t}\,'=\vec{t}\}$$

kodiert die Konfiguration von M bei Eingabe $\mathit{enc}(\mathfrak{A})$ zum Zeitpunkt j.
Daraus folgt dann direkt:

$$M \text{ akzeptiert } \mathit{enc}(\mathfrak{A}) \iff (\mathit{max}^{\mathfrak{A}},\vec{\mathit{max}}^{\mathfrak{A}},\vec{\mathit{max}}^{\mathfrak{A}},0^{\mathfrak{A}})\in R^{n^k} \iff \mathfrak{A}\models\psi.$$

Somit sind wir fertig mit dem Beweis von Theorem 2.46 □

Auf ähnliche Art kann man auch folgendes beweisen:

2.47 Theorem. PFP *beschreibt* PSPACE *auf* FinOrd.

Beweis: Ähnlich zum Beweis von Theorem 2.46. Beachte: Eine PSPACE-Berechnung kann evtl. aus exponentiell vielen Schritten bestehen. Daher kann nicht mehr jeder Berechnungszeitpunkt t durch ein k-Tupel $\vec{t}=(t_1,..,t_k)\in A^k$ kodiert werden. Insgesamt konstruiert man für jedes Problem $\mathsf{C}\subseteq\mathsf{FinOrd}$ in PSPACE eine FO$[\sigma]$-Formel $\varphi(R,u,\vec{p},z)$, so dass für alle endlichen geordneten σ-Strukturen $\mathfrak{A}$ gilt:

$$\mathfrak{A}\in\mathsf{C} \iff \mathfrak{A} \models [\mathbf{pfp}_{R,u,\vec{p},z}\,\varphi](\mathit{max},\vec{\mathit{max}},0).$$

Rest: *Übung*. □

2.48 Bemerkung. In den Theoremen 2.46 und 2.47 ist es wichtig, dass die Strukturen in FinOrd liegen, d.h. dass die Strukturen *geordnet* sind, also eine 2-stellige Relation enthalten, von der man weiß, dass sie eine lineare Ordnung des Universums der Struktur darstellt. Beim Satz von Fagin, der besagt

$$\text{ESO beschreibt NP auf Fin},$$

war dies nicht nötig, da man sich in einer ESO-Formel eine lineare Ordnung "raten" bzw. definieren kann. In Kapitel 3 (Ehrenfeucht-Fraïssé Spiele) werden wir beweisen, dass dies *nicht* durch eine Fixpunktformel geleistet werden kann und dass daher gilt:

$$\text{LFP bzw. IFP beschreibt } \textit{nicht} \text{ PTIME auf Fin},$$

$$\text{PFP beschreibt } \textit{nicht} \text{ PSPACE auf Fin}.$$

Aus den Theoremen 2.46 und 2.47 ergibt sich direkt:

2.49 Korollar. $\text{PSPACE} = \text{PTIME} \iff \text{PFP} = \text{LFP}$ *auf* FinOrd.

Dabei heißt "PFP $=$ LFP auf FinOrd", dass es zu jedem PFP-Satz ψ einen LFP-Satz ψ' gibt, so dass für jede endliche geordnete Struktur $\mathfrak{A}$ gilt:

$$\mathfrak{A} \models \psi \iff \mathfrak{A} \models \psi'.$$

(Die Umkehrung gilt sowieso, da LFP $\leqslant$ PFP; vgl. Proposition 2.45.)

2.50 Bemerkung. Es gilt sogar die folgende Verschärfung von Korollar 2.49, die als *Satz von Abiteboul und Vianu* bekannt ist und die wir in Kapitel 5 (Fixpunktlogiken) auch beweisen werden:

$$\text{PSPACE} = \text{PTIME} \iff \text{PFP} = \text{LFP auf FinOrd} \iff \text{PFP} = \text{LFP auf Fin}.$$

Ähnlich wie beim Beweis des Satzes von Cook, bei dem wir die logische Charakterisierung von NP genutzt haben, um die NP-Vollständigkeit des aussagenlogischen Erfüllbarkeitsproblems nachzuweisen, können wir die logische Charakterisierung von PSPACE nutzen, um die PSPACE-Vollständigkeit des Auswertungsproblems für FO zu beweisen (dies wurde in Kapitel 1, Satz 1.61 schon erwähnt, aber noch nicht bewiesen):

2.51 Satz. *Die kombinierte Komplexität des Auswertungsproblems für* FO *auf* FinOrd *ist* PSPACE*-vollständig.*

Beweis: Mit Satz 1.60 wurde bereits gezeigt, dass die kombinierte Komplexität des Auswertungsproblems für FO in PSPACE liegt. Im Folgenden zeigen wir, dass das Problem auch PSPACE-hart ist, d.h. dass sich jedes Problem $L \in \text{PSPACE}$ auf das Auswertungsproblem für FO reduzieren lässt. Jedes Problem $L \in \text{PSPACE}$ kann man, für eine geeignete Signatur σ, mit einer Klasse $\mathsf{C} \subseteq \text{FinOrd}$ (bzw. der zugehörigen Menge L_{C}) identifizieren. Da nach Voraussetzung $L_{\mathsf{C}} \in \text{PSPACE}$ ist, liefert Theorem 2.47, dass es einen $\text{PFP}[\sigma]$-Satz Φ gibt, so

dass für jede endliche geordnete σ-Struktur $\mathfrak{A}$ gilt: $\mathfrak{A} \in \mathsf{C} \iff \mathfrak{A} \models \Phi$.
Aus dem Beweis von Theorem 2.47 folgt, dass Φ o.B.d.A. von der Form

$$[\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{max}, 0)$$

ist, wobei $\varphi(R, \vec{x}) \in \text{FO}[\sigma \dot\cup \{R\}]$. Sei $r := ar(R)$ die Stelligkeit von R und sei $\vec{x} = x_1, \ldots, x_r$.

Aus Bemerkung 2.39 folgt, dass für jede endliche σ-Struktur $\mathfrak{A}$, für $n := |A|$, für $N := n^r = |A^r|$ und für die Induktionsstufen R^i des Operators $F_{\varphi,\mathfrak{A}}$ gilt:

(1) Falls $R^{2^N} = R^{2^N+1}$, so $\mathbf{pfp}(F_{\varphi,\mathfrak{A}}) = R^{2^N}$.

(2) Falls $R^{2^N} \neq R^{2^N+1}$, so $\mathbf{pfp}(F_{\varphi,\mathfrak{A}}) = \varnothing$.

Unser Ziel ist, eine Polynomialzeit-Reduktion f von $\mathsf{C} \subseteq \mathsf{FinOrd}$ auf das Auswertungsproblem für FO zu finden.

Ansatz: f bildet jedes $\mathfrak{A} \in \mathsf{FinOrd}$ (von dem man wissen will, ob es zur Klasse C gehört) auf ein Tupel $(\mathfrak{A}, \psi_{\Phi,\mathfrak{A}})$ ab, wobei $\psi_{\Phi,\mathfrak{A}}$ ein FO$[\sigma]$-Satz ist, für den gilt:

(3) $\mathfrak{A} \models \psi_{\Phi,\mathfrak{A}} \iff \mathfrak{A} \in \mathsf{C} \overset{\text{def}}{\iff} \mathfrak{A} \models \Phi \overset{\text{def}}{\iff} \mathfrak{A} \models [\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{max}, 0)$.

(4) $\psi_{\Phi,\mathfrak{A}}$ ist in Zeit, die polynomiell in $|A|$ ist, berechenbar.
Insbesondere ist die Länge $|\psi_{\Phi,\mathfrak{A}}|$ der Formel $\psi_{\Phi,\mathfrak{A}}$ polynomiell in der Größe von $\mathfrak{A}$.

Idee: Sei $\mathfrak{A}$ gegeben und sei $n := |A|$ und $N := n^r$. O.B.d.A. ist $n \geqslant 2$, insbesondere werden also die Konstantensymbole 0 und *max* durch zwei verschiedene Elemente in A interpretiert.
Seien $v_1, \ldots, v_N$ Variablen erster Stufe und seien $\vec{x}$, $\vec{y}_1$, …, $\vec{y}_N$ jeweils r-Tupel von Variablen erster Stufe. Für jedes $i \in \{0, \ldots, N\}$ definieren wir induktiv eine FO$[\sigma]$-Formel

$$\varphi_i\big(\vec{x},\, \vec{y}_1, v_1,\, \vec{y}_2, v_2,\, \ldots, \vec{y}_N, v_N\,\big),$$

so dass für alle $\vec{b}_1, \ldots, \vec{b}_N \in A^r$ und alle $d_1, \ldots, d_N \in \{0^{\mathfrak{A}}, \vec{max}^{\mathfrak{A}}\}$ gilt:

$$(*)_i: \quad \{\, \vec{a} \in A^r \;:\; \mathfrak{A} \models \varphi_i[\vec{a}, \vec{b}_1, d_1, \ldots, \vec{b}_N, d_N] \,\} \;=\; \underbrace{F_{\varphi,\mathfrak{A}}{}^{2^i}}_{\substack{2^i\text{-fache Anwendung} \\ \text{des Operators } F_{\varphi,\mathfrak{A}}}} \Big(\{\vec{b}_j \;:\; 1 \leqslant j \leqslant N,\; d_j \neq 0^{\mathfrak{A}}\}\Big)$$

Bevor wir die genaue Konstruktion der Formeln φ_i (für $i \in \{0, \ldots, N\}$) angeben, zeigen wir zunächst, wie wir die gewünschte Formel $\psi_{\Phi,\mathfrak{A}}$ aus φ_N erhalten. Es gilt:

$$\begin{aligned}
\mathfrak{A} \models \Phi \quad &\iff \quad \mathfrak{A} \models [\mathbf{pfp}_{R,\vec{x}}\,\varphi](\vec{max}, 0) \\
&\iff \quad (\vec{max}, 0) \in R^{2^N} \quad \text{und} \quad R^{2^N} = R^{2^N+1} \\
&\iff \quad \mathfrak{A} \models \psi_{\Phi,\mathfrak{A}},
\end{aligned}$$

wobei

$\psi_{\Phi,\mathfrak{A}} :=$

$$\exists \vec{y}_1 \, \exists v_1 \, \cdots \, \exists \vec{y}_N \, \exists v_N \, \Big(\quad \bigwedge_{j=1}^{N} \Big(v_j = 0 \ \vee \ v_j = \mathit{max} \Big) \qquad \text{(Zeile 1)}$$

$$\wedge \ \forall \vec{x} \, \Big(\varphi_N(\vec{x}, \vec{y}_1, 0, .., \vec{y}_N, 0) \ \leftrightarrow \ \bigvee_{j=1}^{N} (v_j \neq 0 \ \wedge \ \vec{x} = \vec{y}_j) \Big) \qquad \text{(Zeile 2)}$$

$$\wedge \ \Big(\bigvee_{j=1}^{N} \Big(v_j \neq 0 \ \wedge \ \vec{y}_j = (\vec{\mathit{max}}, 0) \Big) \Big) \qquad \text{(Zeile 3)}$$

$$\wedge \ \forall \vec{x} \, \Big(\big(\bigvee_{j=1}^{N} (v_j \neq 0 \ \wedge \ \vec{x} = \vec{y}_j) \big) \ \leftrightarrow \ \varphi\big(\vec{x}, \tfrac{R(\vec{u})}{\bigvee_{j=1}^{N}(v_j \neq 0 \wedge \vec{u} = \vec{y}_j)}\big) \Big) \Big) \qquad \text{(Zeile 4)}$$

Dabei besagt

- Zeile 2, dass $R^{2^N} \stackrel{\text{def}}{=} F_{\varphi,\mathfrak{A}}{}^{2^N}(\varnothing) \ = \ \{\vec{y}_j \, : \, v_j \neq 0\}$,
- Zeile 3, dass $(\vec{\mathit{max}}, 0) \ \in \ R^{2^N}$,
- Zeile 4, dass $R^{2^N} \ = \ R^{2^N+1} \stackrel{\text{def}}{=} F_{\varphi,\mathfrak{A}}(R^{2^N})$.

  Hierbei bezeichnet $\varphi\big(\vec{x}, \frac{R(\vec{u})}{\bigvee_{j=1}^{N}(v_j \neq 0 \wedge \vec{u} = \vec{y}_j)}\big)$ die Formel, die aus $\varphi(R, \vec{x})$ entsteht, indem jedes Vorkommen eines Atoms der Form $R(\vec{u})$ durch die Formel $\bigvee_{j=1}^{N}(v_j \neq 0 \wedge \vec{u} = \vec{y}_j)$ ersetzt wird.

Klar ist: Wenn die Formel φ_N die Eigenschaft $(*)_N$ hat und in Zeit poly(n) konstruierbar ist, so ist auch $\psi_{\Phi,\mathfrak{A}}$ in Zeit poly(n) konstruierbar und es gilt:

$$\mathfrak{A} \in \mathsf{C} \quad \Longleftrightarrow \quad \mathfrak{A} \models [\mathbf{pfp}_{R,\vec{x}} \, \varphi](\vec{\mathit{max}}, 0) \quad \Longleftrightarrow \quad \mathfrak{A} \models \psi_{\Phi,\mathfrak{A}}.$$

Somit ist die Abbildung, die jeder σ-Struktur $\mathfrak{A} \in$ FinOrd das Tupel $(\mathfrak{A}, \psi_{\Phi,\mathfrak{A}})$ zuordnet, eine Polynomialzeit-Reduktion von C auf das Auswertungsproblem für FO auf FinOrd.
Um den Beweis von Satz 2.51 zu beenden, müssen wir also nur noch die gewünschte Formel φ_N konstruieren. Dazu konstruieren wir induktiv für $i \in \{0, .., N\}$ Formeln φ_i mit Eigenschaft $(*)_i$, so dass

$$|\varphi_{i+1}| \ \leqslant \ n^{\text{Konstante}} \ + \ |\varphi_i|.$$

Insgesamt ist dann

$$|\varphi_N| \ \leqslant \ N \cdot n^{\text{Konstante}} \ = \ n^r \cdot n^{\text{Konstante}} \ = \ \text{poly}(n).$$

Induktionsanfang $i = 0$: $2^i = 2^0 = 1$. Wir setzen

$$\varphi_0(\vec{x}, \vec{y}_1, v_1, .., \vec{y}_N, v_N) \ := \ \varphi\Big(\vec{x}, \, \frac{R(\vec{u})}{\bigvee_{j=1}^{N}(v_j \neq 0 \ \wedge \ \vec{u} = \vec{y}_j)}\Big).$$

Offensichtlich hat φ_0 die Eigenschaft $(*)_0$, und es gilt $|\varphi_0| = \mathcal{O}(|\varphi| \cdot r \cdot N) = \text{poly}(n)$.

Induktionsschritt $i \to i{+}1$: Um eine Formel $\varphi_{i+1}(\vec{x}, \vec{y}_1, v_1, .., \vec{y}_N, v_N)$ mit der Eigenschaft $(*)_{i+1}$ zu konstruieren, nutzen wir, dass für $F := F_{\varphi,\mathfrak{A}}$ gilt:

$$F^{2^{i+1}}\big(\{\vec{y}_j \,:\, v_j \neq 0\}\big) \;=\; F^{2^i}\Big(F^{2^i}\Big(\{\vec{y}_j \,:\, v_j \neq 0\}\Big)\Big),$$

setzen

$$\begin{array}{rcl} \{\vec{y}\,'_j \,:\, v'_j \neq 0\} & := & F^{2^i}\Big(\{\vec{y}_j \,:\, v_j \neq 0\}\Big) \quad \text{und} \\ \{\vec{y}\,''_j \,:\, v''_j \neq 0\} & := & F^{2^i}\Big(\{\vec{y}\,'_j \,:\, v'_j \neq 0\}\Big) \;=\; F^{2^{i+1}}\Big(\{\vec{y}_j \,:\, v_j \neq 0\}\Big) \end{array}$$

und nutzen die Formel φ_i, um die Mengen $\{\vec{y}\,'_j \,:\, v'_j \neq 0\}$ und $\{\vec{y}\,''_j \,:\, v''_j \neq 0\}$ zu bestimmen. Ein technisches Problem dabei ist, dass die Formel φ_{i+1} nicht *zweimal* die Formel φ_i "aufrufen" kann (einmal mit $\vec{y}_1, v_1, .., \vec{y}_N, v_N$ und dann noch mal mit $\vec{y}\,'_1, v'_1, .., \vec{y}\,'_N, v'_N$, um zuerst $\vec{y}\,'_1, v'_1, .., \vec{y}\,'_N, v'_N$ und danach $\vec{y}\,''_1, v''_1, .., \vec{y}\,''_N, v''_N$ zu ermitteln). Dann wäre nämlich φ_{i+1} *doppelt* so lang wie φ_i, und am Ende wäre

$$|\varphi_N| \;\geqslant\; 2^N \cdot |\varphi|,$$

und das ist viel zu lang als dass man φ_N noch in Zeit poly(n) berechnen könnte.
Um dies zu vermeiden, wählen wir die folgende Formel, die Allquantoren benutzt, um mit einem einzigen "Aufruf" der Formel φ_i sowohl die Werte $\vec{y}\,'_1, v'_1, .., \vec{y}\,'_N, v'_N$ als auch die Werte $\vec{y}\,''_1, v''_1, .., \vec{y}\,''_N, v''_N$ festzulegen.

$$\varphi_{i+1}(\vec{x}, \vec{y}_1, v_1, .., \vec{y}_N, v_N) \;:=$$

$$\exists \vec{y}\,'_1\, \exists v'_1 \,\cdots\, \exists \vec{y}\,'_N\, \exists v'_N \;\; \exists \vec{y}\,''_1\, \exists v''_1 \,\cdots\, \exists \vec{y}\,''_N\, \exists v''_N \,\Big($$

$$\bigvee_{j=1}^{N} (v''_j \neq 0 \,\wedge\, \vec{x} = \vec{y}\,''_j) \;\wedge$$

$$\bigwedge_{j=1}^{N} \big(\, v'_j = 0 \,\vee\, v'_j = \mathit{max}\big) \;\wedge\; \bigwedge_{j=1}^{N} \big(\, v''_j = 0 \,\vee\, v''_j = \mathit{max}\big) \;\wedge$$

$$\forall \vec{w}_1 \forall u_1 \cdots \forall \vec{w}_N \forall u_N \,\Big(\big(\, \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}, v)} \;\vee\; \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}\,', v')} \,\big) \;\rightarrow$$

$$\Big(\, \forall \vec{x} \;\Big(\; \varphi_i(\vec{x}, \vec{w}_1, u_1, .., \vec{w}_N, u_N) \;\leftrightarrow$$

$$\Big(\big(\, \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}, v)} \;\wedge\; \bigvee_{j=1}^{N} (\, v'_j \neq 0 \,\wedge\, \vec{x} = \vec{y}\,'_j \,)\,\big) \vee$$

$$\big(\, \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}\,', v')} \;\wedge\; \bigvee_{j=1}^{N} (\, v''_j \neq 0 \,\wedge\, \vec{x} = \vec{y}\,''_j \,)\,\big)\big)\,\big)\,\Big)\Big),$$

wobei "$\overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}, v)}$" als Abkürzung für die Formel $\bigwedge_{j=1}^{N} \big(\text{"}\vec{w}_j = \vec{y}_j\text{"} \wedge u_j = v_j\big)$ benutzt wird.

Gemäß dieser Konstruktion hat φ_{i+1} die Eigenschaft $(*)_{i+1}$ und es gilt

$$|\varphi_{i+1}| \;\leqslant\; n^{\text{Konstante}} + |\varphi_i|.$$

Speziell für $i = N$ gilt: Die Formel φ_N hat die Eigenschaft $(*)_N$ und kann in Zeit poly(n) konstruiert werden. Somit sind wir fertig mit dem Beweis von Satz 2.51. □

2.3 TC-Logiken zur Beschreibung von LOGSPACE und NLOGSPACE

2.3.1 Die Logik TC

2.52 Definition. Sei A eine Menge, sei $k \in \mathbb{N}_{\geqslant 1}$ und sei $R \subseteq A^{2k}$.
Die *transitive Hülle* (engl.: transitive closure) $\mathbf{tc}(R)$ von R ist folgendermaßen definiert:

$$\mathbf{tc}(R) \;:=\; \left\{ (\vec{x},\vec{y}) \in A^{2k} \;\middle|\; \begin{array}{l} \text{es gibt im Graphen } G_R := (V,E) \text{ mit } V_R := A^k \\ \text{und } E_R := \{(\vec{u},\vec{v}) \in A^k \times A^k \,:\, (\vec{u},\vec{v}) \in R\} \text{ einen} \\ \text{Pfad}^5 \text{der Länge} \geqslant 1 \text{ von Knoten } \vec{x} \text{ zu Knoten } \vec{y} \end{array} \right\}.$$

2.53 Definition (Transitive Hüllen-Logik TC)**.** Sei σ eine Signatur.
Die Formelmenge TC$[\sigma]$ ist induktiv durch die Regeln (A1),(A2),(BC) und (Q1) der Logik erster Stufe, sowie die folgende Regel (TC) definiert:

(TC) Ist $\varphi(\vec{x},\vec{y})$ eine TC$[\sigma]$-Formel, wobei

- $\vec{x}$ und $\vec{y}$ zwei k-Tupel aus paarweise verschiedenen Variablen erster Stufe sind, für ein $k \in \mathbb{N}_{\geqslant 1}$,
- φ außer $\vec{x}$ und $\vec{y}$ evtl. noch andere freie Variablen erster Stufe hat,

und sind $\vec{s}$ und $\vec{t}$ zwei k-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ, so ist

$$[\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$$

eine TC$[\sigma]$-Formel.

2.54 Bemerkungen.

(a) Man beachte, dass es in der Logik TC *keine* Relationsvariablen gibt.

(b) Jede Formel $\varphi(\vec{x},\vec{y})$ mit $2k$ freien Variablen $\vec{x},\vec{y}$ definiert in jeder Struktur $\mathfrak{A}$ (der passenden Signatur) eine $2k$-stellige Relation

$$\varphi(\mathfrak{A}) \;:=\; \{\,(\vec{u},\vec{v}) \in A^{2k} \;:\; \mathfrak{A} \models \varphi[\vec{u},\vec{v}]\,\}$$

und einen Graph $G_{\varphi,\mathfrak{A}} := (V_{\varphi,\mathfrak{A}}, E_{\varphi,\mathfrak{A}})$ mit $V_{\varphi,\mathfrak{A}} = A^k$ und $E_{\varphi,\mathfrak{A}} = \varphi(\mathfrak{A})$.

[5] Ein Pfad der Länge $\ell \in \mathbb{N}$ ist dabei eine Folge $\vec{v}_0,\ldots,\vec{v}_\ell \in V_R$ von Knoten, so dass für alle $i < \ell$ gilt: $(\vec{v}_i,\vec{v}_{i+1}) \in E_R$.

(c) Die *freien Variablen* einer TC-Formel der Form $[\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$ sind

$$\Big(\mathit{frei}(\varphi) \setminus \{\vec{x},\vec{y}\}\Big) \;\cup\; \{s_j \,:\, s_j \in \text{Var}_1\} \;\cup\; \{t_j \,:\, t_j \in \text{Var}_1\}.$$

2.55 Definition (Semantik von TC$[\sigma]$-Formeln)**.** Die Semantik von TC$[\sigma]$-Formeln der Form $\psi := [\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$ ist folgendermaßen definiert:
Ist $\mathit{frei}(\psi) = \{\vec{z}\}$ und ist $\mathfrak{A}$ eine $(\sigma \dot\cup \{\vec{z}\})$-Struktur, so gilt:

$$\mathfrak{A} \;\models\; [\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t}) \quad :\Longleftrightarrow \quad \big(\vec{s}^{\,\mathfrak{A}},\vec{t}^{\,\mathfrak{A}}\big) \;\in\; \mathbf{tc}\,\big(\varphi(\mathfrak{A})\big).$$

2.56 Beispiele.

(a) *Graphzusammenhang:*

Für jeden Graphen $G = (V,E)$ gilt

$$G \;\models\; \forall x\,\forall y\,\big[\;\mathbf{tc}_{x,y}\,\big(x = y \;\vee E(x,y)\big)\;\big](x,y)$$

genau dann, wenn G *stark zusammenhängend* ist, d.h. es gibt von jedem Knoten x zu jedem Knoten y einen Weg.

(b) *Lineare Ordnung zu einer Nachfolger-Relation:*

Die TC$[\{\mathit{succ}\}]$-Formel

$$\psi(u,v) \;:=\; [\mathbf{tc}_{x,y}\,\mathit{succ}(x,y)](u,v)$$

definiert für jedes $n \in \mathbb{N}$ in der Struktur $\mathfrak{A}_n := (\{0,\ldots,n\}, \mathit{succ}^n)$ mit $\mathit{succ}^n := \{(i,i+1) \,:\, 0 \leqslant i < n\}$ die natürliche lineare Ordnung auf $\{0,\ldots,n\}$, d.h. es gilt für alle $n \in \mathbb{N}$ und alle $i,j \in \{0,\ldots,n\}$, dass

$$\mathfrak{A}_n \models \psi[i,j] \;\iff\; i < j.$$

(c) *Addition:*

Die TC$[\{\mathit{succ},0\}]$-Formel

$$\varphi_+(u,v,w) \;:=\; \big[\,\mathbf{tc}_{x_1,x_2,y_1,y_2}\,\mathit{succ}(x_1,y_1) \wedge \mathit{succ}(x_2,y_2)\big]\,(0,u,v,w)$$

definiert für jedes $n \in \mathbb{N}$ in der Struktur $\mathfrak{B}_n := (\{0,\ldots,n\}, \mathit{succ}^n, 0)$ die Addition auf $\{0,\ldots,n\}$. D.h. für alle $n \in \mathbb{N}$ und alle $a,b,c \in \{0,\ldots,n\}$ gilt: $\mathfrak{B}_n \models \varphi_+[a,b,c] \iff a + b = c$. (Idee dabei: Die TC-Formel beschreibt den Pfad $(0,u) \to (1,u{+}1) \to (2,u{+}2) \to \cdots \to (v,u{+}v) \to \cdots$.)

(d) *Kardinalität modulo 2:*

Für alle endlichen linearen Ordnungen $\mathfrak{A} = (A, <^{\mathfrak{A}})$ gilt:

$$\begin{aligned}\mathfrak{A} \;\models\; \exists z_0\,\exists z_{\mathit{max}}\,\Big(\;&\forall x\,\big(\,\neg\, x < z_0 \;\wedge\; \neg\, z_{\mathit{max}} < x\,\big) \;\wedge\\ &\big[\,\mathbf{tc}_{x,y}\;\exists z\big(x < z < y \;\wedge \forall z'(x < z' < y \to z' = z)\big)\big](z_0, z_{\mathit{max}})\;\Big)\end{aligned}$$

genau dann, wenn $|A|$ ungerade ist.

2.3.2 Varianten von TC: Die Logiken DTC, posTC und posDTC

2.57 Definition. Sei A eine Menge, sei $k \in \mathbb{N}_{\geqslant 1}$ und sei $R \subseteq A^{2k}$.
Die *deterministische transitive Hülle* $\mathbf{dtc}(R)$ von R ist folgendermaßen definiert:

$$\mathbf{dtc}(R) \;:=\; \left\{ (\vec{x},\vec{y}) \in A^{2k} \;\middle|\; \begin{array}{l} \text{es gibt im Graphen } G_R := (V_R, E_R) \text{ mit } V_R := A^k \\ \text{und } E_R := \{(\vec{u},\vec{v}) \in A^k \times A^k \,:\, (\vec{u},\vec{v}) \in R\} \text{ einen} \\ \textit{deterministischen} \text{ Pfad von } \vec{x} \text{ zu } \vec{y}, \text{ d.h. einen Pfad} \\ \text{der Länge } \geqslant 1, \text{ so dass es für jeden Knoten } \vec{u} \text{ auf} \\ \text{diesem Pfad (außer evtl. } \vec{y}) \textit{ genau einen} \text{ Knoten } \vec{v} \\ \text{mit } (\vec{u},\vec{v}) \in E_R \text{ gibt} \end{array} \right\}.$$

2.58 Definition (Deterministische Transitive Hüllen-Logik DTC)**.** Sei σ eine Signatur.
Die Formelmenge DTC$[\sigma]$ ist induktiv durch die Regeln (A1),(A2),(BC) und (Q1) der Logik erster Stufe, sowie die folgende Regel (DTC) definiert:

(DTC) Ist $\varphi(\vec{x},\vec{y})$ eine DTC$[\sigma]$-Formel, wobei

- $\vec{x}$ und $\vec{y}$ zwei k-Tupel aus paarweise verschiedenen Variablen erster Stufe sind, für ein $k \in \mathbb{N}_{\geqslant 1}$,
- φ außer $\vec{x}$ und $\vec{y}$ evtl. noch andere freie Variablen erster Stufe hat,

und sind $\vec{s}$ und $\vec{t}$ zwei k-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ, so ist

$$[\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$$

eine DTC$[\sigma]$-Formel.

2.59 Definition (Semantik von DTC$[\sigma]$-Formeln)**.** Die Semantik von DTC$[\sigma]$-Formeln der Form $\psi := [\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$ ist folgendermaßen definiert:
Ist *frei*$(\psi) = \{\vec{z}\}$ und ist $\mathfrak{A}$ eine $(\sigma \dot\cup \{\vec{z}\})$-Struktur, so gilt:

$$\mathfrak{A} \;\models\; [\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t}) \quad :\Longleftrightarrow \quad \big(\vec{s}^{\,\mathfrak{A}}, \vec{t}^{\,\mathfrak{A}}\big) \;\in\; \mathbf{dtc}\,\big(\varphi(\mathfrak{A})\big).$$

2.60 Proposition. *Jede* DTC*-Formel ist äquivalent zu einer* TC*-Formel (kurz:* DTC $\leqslant$ TC*).*

Beweis: Per Induktion über den Formelaufbau. Der einzige interessante Fall ist der $\mathbf{dtc}$-Operator

$$\big[\,\mathbf{dtc}_{\vec{x},\vec{y}}\;\varphi(\vec{x},\vec{y})\big]\,(\vec{s},\vec{t}).$$

Diese Formel ist äquivalent zu der TC-Formel

$$\Big[\,\mathbf{tc}_{\vec{x},\vec{y}}\;\Big(\varphi(\vec{x},\vec{y}) \;\wedge\; \forall\vec{z}\big(\varphi(\vec{x},\vec{z}) \rightarrow \vec{z} = \vec{y}\big)\Big)\;\Big]\,(\vec{s},\vec{t}).$$

□

2.61 Definition (Positive Transitive Hüllen-Logiken posTC und posDTC)**.**
Die Logiken posTC und posDTC sind definiert als die Fragmente von TC und DTC, in denen weder Implikationspfeile "$\rightarrow$" noch Biimplikationspfeile "$\leftrightarrow$" vorkommen und die $\mathbf{tc}$- bzw. $\mathbf{dtc}$-Operatoren nur *positiv* verwendet werden, d.h. im Einflussbereich einer *geraden* Anzahl von Negationssymbolen.

2.62 Proposition. *Jede* DTC*-Formel ist äquivalent zu einer* posDTC*-Formel (kurz:* posDTC $=$ DTC*).*

Beweis: Per Induktion nach dem Formelaufbau. Der Haupschritt besteht darin, eine DTC-Formel der Form

$$\psi \;:=\; \neg\,[\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi]\,(\vec{s},\vec{t})$$

in eine äquivalente posDTC-Formel zu übersetzen. Gemäß Induktionsannahme können wir dabei annehmen, dass sowohl φ als auch $\neg\varphi$ äquivalent zu posDTC-Formeln sind.
Ist σ die Signatur, über der ψ gebildet ist, so gilt für jede $\big(\sigma\dot{\cup}\mathit{frei}(\psi)\big)$-Struktur $\mathfrak{A}$:

$\mathfrak{A} \models \psi$

$\iff$ $\mathfrak{A} \not\models [\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi]\,(\vec{s},\vec{t})$

$\iff$ der (eindeutig bestimmte) deterministische Pfad π in $G_{\varphi,\mathfrak{A}} = (V_{\varphi,\mathfrak{A}}, E_{\varphi,\mathfrak{A}})$ mit $V_{\varphi,\mathfrak{A}} = A^k$ und $E_{\varphi,\mathfrak{A}} = \varphi(\mathfrak{A})$, der bei Knoten $\vec{s}^{\,\mathfrak{A}}$ beginnt, erreicht *nicht* den Knoten $\vec{t}^{\,\mathfrak{A}}$

$\iff$ π erreicht, ohne durch den Knoten $\vec{t}^{\,\mathfrak{A}}$ zu führen, einen Knoten $\vec{z} \in A^k$, der (1) keinen deterministischen Nachfolger in $G_{\varphi,\mathfrak{A}}$ hat oder der (2) auf einem deterministischen Kreis in $G_{\varphi,\mathfrak{A}}$ liegt, der nicht durch Knoten $\vec{t}^{\,\mathfrak{A}}$ führt

$\iff$ $\mathfrak{A} \models \psi'$,

wobei

$$\begin{aligned}\psi' \;:=\; \exists\vec{z}\,\Big(\; & \big(\,\vec{s}=\vec{z} \;\vee\; \big[\;\mathbf{dtc}_{\vec{x},\vec{y}}\,\big(\varphi(\vec{x},\vec{y}) \;\wedge\; \neg\,\vec{y}=\vec{t}\big)\;\big]\,(\vec{s},\vec{z})\,\big) \\ & \wedge\; \Big(\big(\forall\vec{u}\,\neg\varphi(\vec{z},\vec{u}) \;\;\vee\;\; \exists\vec{u}\,\exists\vec{v}\,(\varphi(\vec{z},\vec{u}) \wedge \varphi(\vec{z},\vec{v}) \wedge \neg\,\vec{u}=\vec{v})\big) \qquad (1) \\ & \quad \vee\,\big[\,\mathbf{dtc}_{\vec{x},\vec{y}}\,\big(\varphi(\vec{x},\vec{y}) \;\wedge\; \neg\,\vec{y}=\vec{t}\big)\,\big]\,(\vec{z},\vec{z})\,\Big)\Big). \qquad (2)\end{aligned}$$

□

2.63 Bemerkung. Es ist bekannt, dass obige Aussage *nicht* für die Logik TC gilt, d.h. es gibt eine TC-Formel ψ, zu der es keine posTC-Formel gibt, die *auf allen endlichen Strukturen* äquivalent zu ψ ist. (Das werden wir in dieser Vorlesung allerdings nicht beweisen.)
Es gilt aber die folgende schwächere Äquivalenz: Auf endlichen *geordneten* Strukturen ist TC äquivalent zu posTC (kurz: posTC $=$ TC auf FinOrd). Dies werden wir später auch noch beweisen (Theorem 2.66).

2.3.3 TC-Logiken und Komplexitätsklassen

2.64 Theorem (Immerman, 1986)**.**

(a) posDTC *beschreibt* LOGSPACE *auf* FinOrd.

(b) posTC *beschreibt* NLOGSPACE *auf* FinOrd.

Beweis: Sei σ eine Signatur, die u.a. ein 2-stelliges Relationssymbol $<$ enthält und die o.B.d.A. auch zwei Konstantensymbole 0 und *max* enthält, die in geordneten σ-Strukturen stets mit dem kleinsten bzw. dem größten Element der linearen Ordnung $<$ interpretiert werden.

"$\subseteq$:" Zunächst zeigen wir, dass die Datenkomplexität von posDTC in LOGSPACE liegt.
Jede posDTC-Formel kann man leicht in eine äquivalente posDTC-Formel umformen, bei der kein $\mathbf{dtc}$-Operator im Einflussbereich eines Negationszeichens vorkommt. Per Induktion über den Aufbau von solchen posDTC$[\sigma]$-Formeln zeigen wir, dass jede solche Formel durch eine deterministische Logspace-beschränkte Turing-Maschine ausgewertet werden kann. Die Fälle für die Quantoren und Verknüfungen der Logik erster Stufe werden wie im Algorithmus EVAL aus dem Beweis von Lemma 1.59 behandelt. Für den $\mathbf{dtc}$-Operator, d.h. für eine Formel der Form

$$\psi \;:=\; [\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$$

wird bei Eingabe einer Struktur $\mathfrak{A}$ der in Knoten $\vec{s}^{\mathfrak{A}}$ startende deterministische Pfad berechnet. Dazu geht man nach folgendem Algorithmus vor:

1. $\vec{v} := \vec{s}^{\mathfrak{A}}$
2. FOR $i := 0$ TO $|A|^k$ DO
3. $\vec{w} :=$ NIL
4. FOR ALL $\vec{u} \in A^k$ DO
5. IF $\mathfrak{A} \models \varphi[\vec{v},\vec{u}]$ THEN
6. IF $\vec{w} =$ NIL THEN $\vec{w} := \vec{u}$ ELSE STOP WITH OUTPUT "nein" ENDIF ENDIF
7. ENDFOR
8. IF $\vec{w} =$ NIL THEN STOP WITH OUTPUT "nein" ELSE $\vec{v} := \vec{w}$ ENDIF
9. IF $\vec{v} = \vec{t}^{\mathfrak{A}}$ THEN STOP WITH OUTPUT "ja" ENDIF
10. ENDFOR
11. STOP WITH OUTPUT "nein"

Man sieht leicht, dass dieser Algorithmus genau dann "ja" ausgibt, wenn der Knoten $\vec{t}^{\mathfrak{A}}$ auf dem in $\vec{s}^{\mathfrak{A}}$ startenden durch φ definierten deterministischen Pfad liegt, d.h., wenn $\mathfrak{A} \models [\mathbf{dtc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$.
Gemäß Induktionsannahme wird für Zeile 5 nur Platz $\mathcal{O}(\log|A|)$ benötigt. Abgesehen davon muss der Algorithmus zu jedem Zeitpunkt nur die drei Knoten $\vec{v}$, $\vec{w}$ und $\vec{u}$ speichern, und dazu reicht Platz $\mathcal{O}(\log|A|)$.
Insgesamt erhalten wir somit, dass die Datenkomplexität von posDTC in LOGSPACE liegt.

Auf ähnliche Weise kann man auch zeigen, dass die Datenkomplexität von posTC in NLOGSPACE liegt — zum Auswerten einer Formel der Form $[\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t})$ "rät" eine nichtdeterministische Logspace-beschränkte Turing-Maschine an Stelle der Schleife in den Zeilen 4–7 dabei einfach einen Knoten $\vec{u}$, für den sie dann testet, ob $\mathfrak{A} \models \varphi[\vec{v},\vec{u}]$ gilt.

"$\supseteq$:" Sei nun C eine Klasse endlicher geordneter σ-Strukturen, so dass $L_{\mathsf{C}} \in$ LOGSPACE. D.h. es gibt eine deterministische Turing-Maschine $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ (die ein separates Eingabeband besitzt) und eine Konstante k, so dass M bei Eingabe der Kodierung einer endlichen geordneten σ-Struktur $\mathfrak{A}$ mit $n := |A|$ auf Platz $k \cdot \lfloor \log n \rfloor$ entscheidet, ob $\mathfrak{A} \in \mathsf{C}$. Wir können o.B.d.A. annehmen, dass $|\mathit{enc}(\mathfrak{A})| \leqslant n^k$.

Idee: Jede Konfiguration von M bei Eingabe $\mathit{enc}(\mathfrak{A})$ ist durch ein Tupel $(q, p_{\text{Eingabe}}, p, w)$ eindeutig bestimmt, wobei $q \in Q$ den aktuellen Zustand bezeichnet, $p_{\text{Eingabe}} \in \{0,\ldots,n^k\}$ die Kopfposition auf dem Eingabeband, $p \in \{0,\ldots,k\cdot\lfloor\log n\rfloor\}$ die Kopfposition auf dem Arbeitsband und $w \in \Gamma^{k\cdot\lfloor\log n\rfloor}$ die komplette Beschriftung des Arbeitsbandes. Daher gibt es eine Konstante $d \in \mathbb{N}$, so dass M bei Eingabe einer Struktur $\mathfrak{A}$ mit $n := |A|$ höchstens $2^{d\cdot\log n} = n^d$ Konfigurationen durchlaufen kann. Jede einzelne solche Konfiguration kann man durch ein d-Tupel $\vec{v} \in A^d$ repräsentieren.
Das Problem C wird dann durch eine posDTC$[\sigma]$-Formel der Form

$$\psi \;:=\; \exists\vec{s}\;\exists\vec{t}\;\Big(\varphi_{\mathit{Start}}(\vec{s})\;\wedge\;\varphi_{\mathit{Akzeptiere}}(\vec{t})\;\wedge\;\big[\;\mathbf{dtc}_{\vec{x},\vec{y}}\;\varphi_{\mathit{Schritt}}(\vec{x},\vec{y})\;\big]\;(\vec{s},\vec{t})\Big)$$

beschrieben, wobei

- $\varphi_{\mathit{Start}}(\vec{s})$ besagt, dass $\vec{s}$ die Startkonfiguration von M bei Eingabe $\mathit{enc}(\mathfrak{A})$ ist,
- $\varphi_{\mathit{Akzeptiere}}(\vec{t})$ besagt, dass $\vec{t}$ eine akzeptierende Endkonfiguration ist,
- $\varphi_{\mathit{Schritt}}(\vec{x},\vec{y})$ besagt, dass $\vec{y}$ eine Nachfolgekonfiguration von $\vec{x}$ ist.

Genauer: Jede Konfiguration $(q, p_{\text{Eingabe}}, p, w)$ wird durch ein Tupel der Länge

$$d \;:=\; 1 + k + k + |\Gamma| \cdot k$$

der Form

$$\Big(q,\ \vec{p}_{\text{Eingabe}},\ \vec{p},\ \big(\vec{x}_\gamma\big)_{\gamma\in\Gamma}\Big)$$

repräsentiert, wobei q, $\vec{p}_{\text{Eingabe}}$ und $\vec{p}$ Zustand sowie Bandpositionen analog zum Beweis von Theorem 2.46 kodieren. Die Tupel $\vec{x}_\gamma = (x_{\gamma,0},\ldots,x_{\gamma,k-1})$, für $\gamma \in \Gamma$, beschreiben die Beschriftung des Arbeitsbandes folgendermaßen. Für alle $i, j \in \{0,\ldots,n-1\}$ gilt:

an Position $j \cdot \log n + i$ steht das Symbol γ $\iff$ das i-te Bit der Zahl $x_{\gamma,j}$ ist 1, d.h. $\left\lfloor \frac{x_{\gamma,j}}{2^i} \right\rfloor$ ist ungerade.

Man kann zeigen (*Übung*), dass es posDTC[$<$]-Formeln $\varphi_{\mathit{Bit}=1}(x,y)$ und $\varphi_{\mathit{Bit}=0}(x,y)$ gibt, die besagen, dass das y-te Bit der Zahl x 1 bzw. 0 ist. Diese Formeln kann man nutzen, um posDTC[σ]-Formeln φ_{Start}, $\varphi_{\mathit{Akzeptiere}}$ und $\varphi_{\mathit{Schritt}}$ mit den gewünschten Eigenschaften zu konstruieren. Rest: *Übung*.

Mit derselben Vorgehensweise kann man auch für jede Klasse C mit $L_{\mathsf{C}} \in$ NLOGSPACE eine posTC-Formel konstruieren, die genau von genau den Strukturen erfüllt wird, die in C liegen. □

Aus Theorem 2.64 (a) und Proposition 2.62 folgt direkt:

2.65 Theorem. DTC *beschreibt* LOGSPACE *auf* FinOrd.

Um die analoge Aussage auch für TC und NLOGSPACE zu beweisen, muss man zeigen, dass auch *Negationen* von TC-Formeln durch nichtdeterministische Logspace-beschränkte Turing-Maschinen ausgewertet werden können. Dies wird durch folgenden Satz gewährleistet:

2.66 Theorem (Immerman, 1987)**.** *Auf endlichen geordneten Strukturen ist* TC *äquivalent zu* posTC *(kurz:* posTC $=$ TC *auf* FinOrd*).*

Beweis: Per Induktion nach dem Formelaufbau. Der Hauptschritt besteht darin, eine TC[σ]-Formel der Form

$$\psi \;:=\; \neg\big[\ \mathbf{tc}_{\vec{x},\vec{y}}\ \varphi(\vec{x},\vec{y})\ \big]\,(\vec{s},\vec{t}\,)$$

in eine posTC[σ]-Formel zu übersetzen, die auf allen *geordneten* endlichen σ-Strukturen äquivalent zu ψ ist. Gemäß Induktionsannahme können wir davon ausgehen, dass sowohl φ als auch $\neg\varphi$ auf geordneten endlichen Strukturen äquivalent zu einer posTC-Formel sind.
Sei $k \geqslant 1$ die Länge der Tupel $\vec{x}$ und $\vec{y}$, d.h. $\vec{x} = x_1,..,x_k$ und $\vec{y} = y_1,..,y_k$. Sei $\sigma' := \sigma \dot\cup \mathit{frei}(\psi)$.

Zu einer endlichen geordneten σ'-Struktur $\mathfrak{A}$ und k-Tupeln $\vec{a},\vec{b} \in A^k$ und jeder posTC[σ']-Formel $\chi(\vec{x},\vec{y})$ definieren wir

$$\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{a},\vec{b}\,) \;:=\; \begin{cases} \infty\,, & \text{falls es keinen Pfad der Länge } \geqslant 1 \text{ in } G_{\chi,\mathfrak{A}} \text{ von } \vec{a} \text{ nach } \vec{b} \text{ gibt,} \\ \min\left\{ \ell \geqslant 1 \;\middle|\; \begin{array}{l}\text{es gibt in } G_{\chi,\mathfrak{A}} \text{ einen Pfad der} \\ \text{Länge } \ell \text{ von } \vec{a} \text{ nach } \vec{b}\end{array} \right\}, & \text{sonst.} \end{cases}$$

Es gilt:

$$\begin{aligned} & \mathfrak{A} \models \neg[\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t}\,) \\ \iff\quad & \big|\,\{\vec{b} \in A^k \;:\; \mathrm{Dist}^{\mathfrak{A}}_{\varphi}(\vec{s},\vec{b}) < \infty\}\,\big| \;=\; \big|\,\{\vec{b} \in A^k \;:\; \mathrm{Dist}^{\mathfrak{A}}_{\varphi(\vec{x},\vec{y})\,\wedge\,\neg\vec{y}=\vec{t}}(\vec{s},\vec{b}) < \infty\}\,\big| \qquad (*) \end{aligned}$$

d.h. die Anzahl der Elemente, die in $G_{\varphi,\mathfrak{A}}$ von Knoten $\vec{s}$ aus erreichbar sind, ist gleich der Anzahl der Elemente, die von $\vec{s}$ aus über einen Pfad erreichbar sind, der den Knoten $\vec{t}$ *nicht*

durchläuft.

Ziel für den Rest des Beweises: Drücke die Gleichung $(*)$ durch eine posTC-Formel aus.

Schritt 1: Es gilt für alle Formeln $\chi(\vec{x},\vec{y})$ und alle Tupel $\vec{s},\vec{b} \in A^k$ und für $n := |A|$:

$$\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{b}) < \infty \iff \mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{b}) \leqslant n^k,$$

denn der Graph $G_{\chi,\mathfrak{A}}$ hat nur $|A|^k$ Knoten.

Schritt 2: Wie beim Beweis des Satzes von Immerman und Vardi (Theorem 2.46) können wir für $n := |A|$ mittels der linearen Ordnung $<^{\mathfrak{A}}$ das Universum A mit der Menge $\{0,\ldots,n-1\}$ identifizieren. Unter Verwendung der lexikographischen Ordnung $<^{\mathfrak{A}}_{lex}$ identifizieren wir $k{+}1$-Tupel $\vec{c} \in A^{k+1}$ mit Zahlen in $\{0,\ldots,n^{k+1}-1\}$, indem wir dem Tupel $\vec{c}$ die Zahl

$$\big[\vec{c}\big]^{\mathfrak{A}} \;:=\; \mathit{rg}_{<^{\mathfrak{A}}_{lex}}(\vec{c}) \;\in\; \{0,\ldots,n^{k+1}-1\}$$

zuordnen. Insbesondere gilt

$$|A|^k \;=\; n^k \;=\; \big[\,(1,\underbrace{0,0,\cdots,0}_{k})\,\big]^{\mathfrak{A}},$$

wobei 0 das kleinste und 1 das zweitkleinste Element in A bzgl. $<^{\mathfrak{A}}$ bezeichnen.
Wir nehmen im Folgenden o.B.d.A. an, dass die Signatur σ zwei Konstantensymbole 0 und 1 enthält, die in geordneten σ-Strukturen stets mit dem kleinsten bzw. dem zweitkleinsten Element interpretiert werden.

Schritt 3: Für jede posTC$[\sigma']$-Formel $\chi(\vec{x},\vec{y})$ konstruieren wir eine posTC-Formel $\mathit{dist}_{\chi}(\vec{x},\vec{y},\vec{u})$ (wobei $\vec{x}$ und $\vec{y}$ jeweils k-Tupel und $\vec{u}$ ein $k{+}1$-Tupel von Variablen erster Stufe sind), so dass für alle endlichen geordneten σ'-Strukturen $\mathfrak{A}$ und alle $\vec{a},\vec{b} \in A^k$ und $\vec{d} \in A^{k+1}$ gilt:

$$\mathfrak{A} \models \mathit{dist}_{\chi}[\vec{a},\vec{b},\vec{d}] \quad\Longleftrightarrow\quad \mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{a},\vec{b}) \;\leqslant\; \big[\vec{d}\,\big]^{\mathfrak{A}}.$$

Dazu wählen wir

$$\mathit{dist}_{\chi}(\vec{x},\vec{y},\vec{u}) \;:=\; \big[\;\mathbf{tc}_{\vec{x},\vec{u},\vec{x}',\vec{u}'}\;\chi(\vec{x},\vec{x}\,') \wedge \mathit{succ}_{<_{lex}}(\vec{u},\vec{u}\,')\,\big]\,(\vec{x},\vec{0},\vec{y},\vec{u}).$$

(Idee dabei: In dem Tupel $\vec{u}$ wird die Länge eines Pfads von $\vec{x}$ nach $\vec{y}$ gezählt.)

Schritt 4: Für jede TC$[\sigma']$-Formel $\chi(\vec{x},\vec{y})$, so dass sowohl χ als auch $\neg\chi$ äquivalent zu einer posTC-Formel sind, konstruieren wir eine posTC-Formel $\mathit{anzahl}_{\chi}(\vec{x},\vec{z})$, so dass für alle endlichen geordneten σ'-Strukturen $\mathfrak{A}$, alle $\vec{s} \in A^k$ und alle $\vec{c} \in A^{k+1}$ gilt:

$$\mathfrak{A} \models \mathit{anzahl}_{\chi}\,[\vec{s},\vec{c}] \quad\Longleftrightarrow\quad \big[\vec{c}\big]^{\mathfrak{A}} \;=\; \big|\,\{\vec{b}\;:\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{b}) < \infty\,\}\,\big|\,.$$

Dazu betrachten wir zunächst die Funktion

$$\mathrm{Anz}^{\mathfrak{A}}_{\chi} \;:\; A^k \times \{0,\ldots,n^k\} \;\to\; \{0,\ldots,n^k\}$$

mit

$$\text{Anz}^{\mathfrak{A}}_{\chi}(\vec{s}, d) \;:=\; \big|\, \{\vec{b} \in A^k \;:\; \text{Dist}^{\mathfrak{A}}_{\chi}(\vec{s}, \vec{b}) \leqslant d\} \,\big| \,.$$

Für jedes feste $\vec{s} \in A^k$ können wir $\text{Anz}^{\mathfrak{A}}_{\chi}(\vec{s}, d)$ induktiv für $d = 0, 1, 2, \ldots$ folgendermaßen bestimmen:

Induktionsanfang $d = 0$:

$\text{Anz}^{\mathfrak{A}}_{\chi}(\vec{s}, 0) = 0$, denn für alle $\vec{b} \in A^k$ gilt gemäß Definition, dass $\text{Dist}^{\mathfrak{A}}_{\chi}(\vec{s}, \vec{b}) \geqslant 1$.

Induktionssschritt $d \to d{+}1$:

Sei $c := \text{Anz}^{\mathfrak{A}}_{\chi}(\vec{s}, d)$. Gesucht ist die Zahl c', so dass $\text{Anz}^{\mathfrak{A}}_{\chi}(\vec{s}, d{+}1) = c'$. Algorithmisch können wir dies ermitteln, indem wir mit einem Variablentupel $\vec{y}\,'$ nach und nach ganz A^k gemäß der lexikographischen Ordnung durchlaufen und dabei den Wert einer Variablen v jedesmal dann um 1 hochzählen, wenn für das gerade betrachtete Tupel $\vec{y}\,'$ gilt: $\text{Dist}^{\mathfrak{A}}_{\chi}(\vec{s}, \vec{y}) \leqslant d{+}1$.

Unser nächstes Ziel ist nun, eine posTC-Formel ζ_χ zu konstruieren, die auf dieser Methode basiert. Sie benutzt $k{+}1$-Tupel $\vec{v}$, $\vec{u}$ und $\vec{u}\,'$, um die Anzahl der gefundenen k-Tupel bzw. die Distanzen d und $d{+}1$ zu repräsentieren. Die Formel

$$\zeta_\chi(\vec{u}, \vec{z}, \vec{u}\,', \vec{z}\,', \vec{s}\,)$$

soll für $\vec{u}, \vec{u}\,'$ mit $\big[\vec{u}\,'\big]^{\mathfrak{A}} = \big[\vec{u}\big]^{\mathfrak{A}}{+}1$ besagen:

$$\text{Falls} \quad \big[\vec{z}\big]^{\mathfrak{A}} = \text{Anz}^{\mathfrak{A}}_{\chi}\big(\vec{s}, \big[\vec{u}\big]^{\mathfrak{A}}\big), \quad \text{so} \quad \big[\vec{z}\,'\big]^{\mathfrak{A}} = \text{Anz}^{\mathfrak{A}}_{\chi}\big(\vec{s}, \big[\vec{u}\,'\big]^{\mathfrak{A}}\big)\big).$$

Wir wählen dazu

$$\zeta_\chi(\vec{u}, \vec{z}, \vec{u}\,', \vec{z}\,', \vec{s}\,) \;:=\; \big[\; \mathbf{tc}_{\vec{y},\vec{v},\vec{y}\,',\vec{v}\,'}\; \delta_\chi \,\big]\, (\vec{0}, \vec{0}, \vec{max}, \vec{z}\,'),$$

wobei

$$\delta_\chi(\vec{y}, \vec{v}, \vec{y}\,', \vec{v}\,', \vec{s}, \vec{u}, \vec{u}\,') \;:=\; \Big(\;\; \big(\, \mathit{succ}_{<_{lex}}(\vec{y}, \vec{y}\,') \;\wedge\; \mathit{dist}_\chi(\vec{s}, \vec{y}\,', \vec{u}\,') \;\wedge\; \mathit{succ}_{<_{lex}}(\vec{v}, \vec{v}\,') \,\big)$$
$$\vee \,\big(\, \mathit{succ}_{<_{lex}}(\vec{y}, \vec{y}\,') \;\wedge\; \text{“}\neg\, \mathit{dist}_\chi(\vec{s}, \vec{y}\,', \vec{u}\,')\text{”} \;\wedge\; \vec{v} = \vec{v}\,' \,\big) \Big).$$

Um sicherzustellen, dass die Formel ζ_χ in posTC liegt, müssen wir noch eine posTC-Formel für “$\neg\, \mathit{dist}_\chi(\vec{s}, \vec{y}\,', \vec{u}\,')$” finden, d.h. wir müssen eine posTC-Formel finden, die besagt

$$\text{Dist}^{\mathfrak{A}}_{\chi}\big(\vec{s}, \vec{y}\,'\big) \;>\; \big[\vec{u}\,'\big]^{\mathfrak{A}}.$$

Falls $\big[\vec{u}\,'\big]^{\mathfrak{A}} = \big[\vec{u}\big]^{\mathfrak{A}}{+}1$ und $\big[\vec{z}\big]^{\mathfrak{A}} = \text{Anz}^{\mathfrak{A}}_{\chi}(\vec{s}, \big[\vec{u}\big]^{\mathfrak{A}})$, so gilt:

- Ist $\big[\vec{u}\big]^{\mathfrak{A}} = 0$, so $\text{Dist}^{\mathfrak{A}}_{\chi}(\vec{s}, \vec{y}\,') > \big[\vec{u}\,'\big]^{\mathfrak{A}} \iff \mathfrak{A} \not\models \chi(\vec{s}, \vec{y}\,')$.
- Ist $\big[\vec{u}\big]^{\mathfrak{A}} > 0$, so $\text{Dist}^{\mathfrak{A}}_{\chi}(\vec{s}, \vec{y}\,') > \big[\vec{u}\,'\big]^{\mathfrak{A}} \iff$ es gibt $\vec{z}$ Elemente $\vec{w}\,' \neq \vec{y}\,'$, so dass $\text{Dist}^{\mathfrak{A}}_{\chi}(\vec{s}, \vec{w}\,') \leqslant \big[\vec{u}\big]^{\mathfrak{A}}$ und $\mathfrak{A} \not\models \chi(\vec{w}\,', \vec{y}\,')$.

Genau dies wird durch die folgende posTC-Formel bewerkstelligt, die in der Formel δ_χ an die Stelle von "$\neg \mathit{dist}_\chi(\vec{s},\vec{y}\,',\vec{u}\,')$" gesetzt werden muss:

$$\Big(\vec{u}=\vec{0}\ \wedge\ \neg\,\chi(\vec{s},\vec{y}\,')\Big)\ \vee$$
$$\Big(\vec{u}\neq\vec{0}\ \wedge\ \Big[\,\mathbf{tc}_{\vec{w},\vec{q},\vec{w}\,',\vec{q}\,'}\ \begin{pmatrix} (\ \mathit{succ}_{<_{lex}}(\vec{w},\vec{w}\,')\ \wedge\ \vec{q}=\vec{q}\,')\ \vee \\ (\ \mathit{succ}_{<_{lex}}(\vec{w},\vec{w}\,')\ \wedge\ \mathit{succ}_{<_{lex}}(\vec{q},\vec{q}\,')\ \wedge \\ \ \vec{w}\,'\neq\vec{y}\,'\ \wedge\ \mathit{dist}_\chi(\vec{s},\vec{w}\,',\vec{u})\ \wedge\ \neg\,\chi(\vec{w}\,',\vec{y}\,') \end{pmatrix}\,\Big]\ (\vec{0},\vec{0},\mathit{m\vec{a}x},\vec{z})\ \Big).$$

(Idee dabei: Durchlaufe A^k mit Variablen $\vec{w}\,'$ gemäß der lexikographischen Ordnung und zähle den Wert der Variablen $\vec{q}$ immer dann um 1 hoch, wenn $\vec{w}\,'\neq\vec{y}\,'$ und $\mathrm{Dist}^{\mathfrak{A}}_\chi(\vec{s},\vec{w}\,')\leqslant \big[\vec{u}\big]^{\mathfrak{A}}$ und $\mathfrak{A}\not\models\chi(\vec{w},\vec{y}\,')$.)

Schließlich wählen wir

$$\mathit{anzahl}_\chi(\vec{s},\vec{z})\ :=\ \big[\ \mathbf{tc}_{\vec{u},\vec{z},\vec{u}\,',\vec{z}\,'}\ \mathit{succ}_{<_{lex}}(\vec{u},\vec{u}\,')\ \wedge\ \zeta_\chi(\vec{u},\vec{z},\vec{u}\,',\vec{z}\,',\vec{s})\ \big]\ (\underbrace{0\vec{0}}_{\hat{=}0},\vec{0},\underbrace{1\vec{0}}_{\hat{=}n^k},\vec{z})\,.$$

Dies ist eine posTC-Formel, die besagt, dass $\big[\vec{z}\big]^{\mathfrak{A}}$ die Anzahl aller Knoten von $G_{\chi,\mathfrak{A}}$ ist, die über einen Pfad der Länge $\leqslant n^k$ von Knoten $\vec{s}$ aus erreichbar sind. Gemäß Schritt 1 besagt die Formel $\mathit{anzahl}_\chi(\vec{s},\vec{z})$, dass $\big[\vec{z}\big]^{\mathfrak{A}}$ die Anzahl der von $\vec{s}$ aus erreichbaren Knoten in $G_{\chi,\mathfrak{A}}$ ist. Somit sind wir fertig mit Schritt 4.

Schritt 5: Unter Benutzung der Gleichung $(*)$ und der posTC-Formel aus Schritt 4 erhalten wir, dass die Formel

$$\neg\ \big[\ \mathbf{tc}_{\vec{x},\vec{y}}\ \varphi(\vec{x},\vec{y})\ \big]\ (\vec{s},\vec{t})$$

äquivalent zur posTC-Formel

$$\exists\vec{z}\ \Big(\mathit{anzahl}_\varphi(\vec{s},\vec{z})\ \wedge\ \mathit{anzahl}_{\varphi(\vec{x},\vec{y})\wedge\neg\vec{y}=\vec{t}}(\vec{s},\vec{z})\Big)$$

ist. Dies beendet den Beweis von Theorem 2.66. □

Aus Theorem 2.64 (b) und Theorem 2.66 folgt direkt:

2.67 Theorem. TC *beschreibt* NLOGSPACE *auf* FinOrd.

Mit Hilfe dieser logischen Beschreibung von NLOGSPACE kann man leicht beweisen, dass NLOGSPACE unter Komplementbildung abgeschlossen ist:

2.68 Korollar (Satz von Immerman und Szelepcsényi)**.** NLOGSPACE = co-NLOGSPACE.

Beweis: "$\supseteq$:" Wir müssen zeigen, dass für jede Sprache $L\in$ NLOGSPACE gilt, dass auch das Komplement $\overline{L}$ in NLOGSPACE liegt. Sei also L ein Problem in NLOGSPACE. Gemäß Theorem 2.67 gibt es einen TC-Satz ψ, der L beschreibt, d.h. für alle Strukturen $\mathfrak{A}$ gilt: $\mathfrak{A}\in L\iff\mathfrak{A}\models\psi$. Da die Logik TC abgeschlossen ist unter Negation, ist $\psi':=\neg\psi$ ein TC-Satz, und es gilt für alle Strukturen $\mathfrak{A}$: $\mathfrak{A}\models\psi'\iff\mathfrak{A}\notin L\iff\mathfrak{A}\in\overline{L}$. Somit ist ψ' ein TC-Satz, der das Problem $\overline{L}$ beschreibt. Gemäß Theorem 2.67 gilt $\overline{L}\in$ NLOGSPACE.
"$\subseteq$:" analog. □

Zusammenfassung der logischen Charakterisierungen:

In Kapitel 2 wurden die in Abbildung 2.1 dargestellten logischen Charakterisierungen von Komplexitätsklassen behandelt.

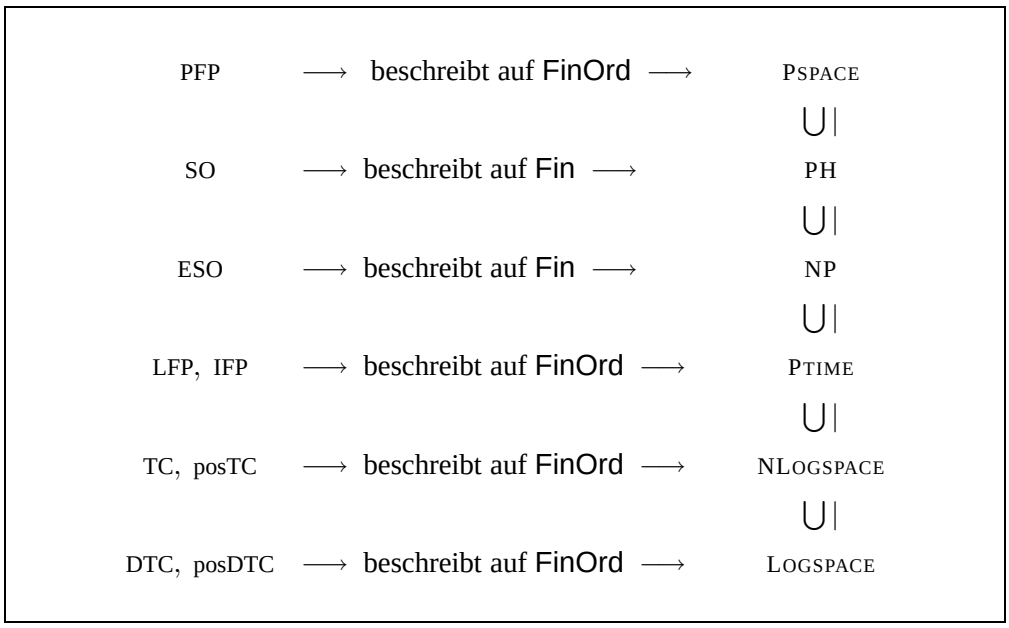


Abbildung 2.1: Logische Beschreibungen von Komplexitätsklassen.

3 Ehrenfeucht-Fraïssé Spiele

Ehrenfeucht-Fraïssé Spiele (kurz: EF-Spiele): eine Methode, zum Beweis, dass bestimmte Probleme *nicht* durch Formeln einer bestimmten Logik (z.B. FO) definierbar sind.

Vereinbarung:
Ab jetzt werden wieder sowohl endliche als auch unendliche Strukturen betrachtet.

3.1 Definition. Sei $\mathscr{L}$ eine Logik, S eine Klasse von Strukturen, σ eine Signatur und $\mathsf{C} \subseteq \mathsf{S}$ eine Klasse von σ-Strukturen.
C heißt *$\mathscr{L}$-definierbar in* S (auch: *$\mathscr{L}$-axiomatisierbar*), falls es einen $\mathscr{L}[\sigma]$-Satz φ gibt, so dass $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\varphi)$ (d.h. für alle σ-Strukturen $\mathfrak{A} \in \mathsf{S}$ gilt: $\mathfrak{A} \in \mathsf{C} \iff \mathfrak{A} \models \varphi$.)

3.1 Das EF-Spiel für die Logik erster Stufe

3.1.1 Vorbemerkungen

Bevor wir die Spielregeln des EF-Spiels einführen, halten wir zunächst einige nützliche Beobachtungen und Begriffe fest.

3.2 Proposition. *Jede endliche Struktur ist bis auf Isomorphie in der Logik erster Stufe definierbar, d.h. für jede Signatur σ und jede endliche σ-Struktur $\mathfrak{A}$ gibt es einen* FO$[\sigma]$*-Satz $\varphi_{\mathfrak{A}}$, so dass für alle endlichen σ-Strukturen $\mathfrak{B}$ gilt:* $\mathfrak{B} \models \varphi_{\mathfrak{A}} \iff \mathfrak{B} \cong \mathfrak{A}$.

Beweis: *Übung.* □

Zur Konstruktion der Formel $\varphi_{\mathfrak{A}}$ im Beweis von Proposition 3.2 benötigt man in etwa so viele Quantoren wie es Elemente im Universum von $\mathfrak{A}$ gibt.

3.3 Definition (Quantorenrang)**.** Der *Quantorenrang* (auch: Quantorentiefe) $qr(\varphi)$ einer FO-Formel φ ist die maximale Anzahl von ineinander geschachtelten Quantoren, die in der Formel φ vorkommen. Per Induktion nach dem Formelaufbau ist der Quantorenrang also folgendermaßen definiert:

- $qr(\varphi) = 0$ falls φ eine atomare Formel ist
- $qr(\varphi) = qr(\psi)$ falls φ eine Formel der Form $\neg\psi$ ist
- $qr(\varphi) = qr(\psi) + 1$ falls φ eine Formel der Form $\exists x\psi$ oder $\forall x\psi$ ist
- $qr(\varphi) = \max\{qr(\varphi_1), qr(\varphi_2)\}$ falls φ eine Formel der Form $(\varphi_1 * \varphi_2)$ für ein $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ ist.

3.4 Beispiele.

- $qr\big(\exists x\, \forall y\, (\, x = y \;\vee\; R(x,y,z)\,)\big) \;=\; 2,$
- $qr\big(\exists x\, \big(T(x) \;\vee\; \forall y\, R(x,y,z)\big)\big) \;=\; 2,$
- $qr\big(\big(\,\exists x\, T(x)\,\big) \;\vee\; \forall y\, (\, R(y,y,z) \to y = z\,)\big) \;=\; 1.$

3.5 Definition (m-Äquivalenz)**.** Sei σ eine Signatur und sei $m \in \mathbb{N}$. Zwei σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ heißen *m-äquivalent* (kurz: $\mathfrak{A} \equiv_m \mathfrak{B}$), falls sie die gleichen FO$[\sigma]$-Sätze der Quantorentiefe m erfüllen.

3.6 Bemerkungen. (a) Für jedes $m \in \mathbb{N}$ ist $\equiv_m$ eine *Äquivalenzrelation* auf der Klasse aller σ-Strukturen.

(b) Für alle natürlichen Zahlen $m \leqslant n$ gilt: Falls $\mathfrak{A} \equiv_n \mathfrak{B}$, so auch $\mathfrak{A} \equiv_m \mathfrak{B}$.
Umgekehrt heißt das: Falls $\mathfrak{A} \not\equiv_m \mathfrak{B}$, so $\mathfrak{A} \not\equiv_n \mathfrak{B}$.

Zum Nachweis, dass bestimmte Probleme *nicht* FO-definierbar sind, kann man sich folgende Beobachtung zu Nutze machen:

3.7 Proposition. *Sei σ eine Signatur und S eine Klasse von σ-Strukturen. Sei $\mathsf{C} \subseteq \mathsf{S}$ eine Teilklasse von S, so dass es für jedes $m \in \mathbb{N}$ ein $\mathfrak{A}_m \in \mathsf{C}$ und ein $\mathfrak{B}_m \in \mathsf{S} \setminus \mathsf{C}$ mit $\mathfrak{A}_m \equiv_m \mathfrak{B}_m$ gibt. Dann ist C nicht* FO*-definierbar in S.*

Beweis: Angenommen C ist FO-definierbar in S, d.h. es gibt einen FO$[\sigma]$-Satz φ mit $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\varphi)$. Sei $m := qr(\varphi)$ der Quantorenrang von φ. Laut Voraussetzung gibt es Strukturen $\mathfrak{A}_m \in \mathsf{C}$ und $\mathfrak{B}_m \in \mathsf{S} \setminus \mathsf{C}$, so dass $\mathfrak{A}_m \equiv_m \mathfrak{B}_m$. Wegen $\mathfrak{A}_m \in \mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\varphi)$, gilt $\mathfrak{A}_m \models \varphi$ und $\mathfrak{B}_m \not\models \varphi$. Da $qr(\varphi) = m$ ist, widerspricht dies aber der Aussage $\mathfrak{A}_m \equiv_m \mathfrak{B}_m$. □

3.8 Definition (Partieller Isomorphismus)**.** Sei σ eine Signatur und $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen.

(a) Eine Abbildung

$$\pi \;:\; \mathrm{def}(\pi) \;\to\; \mathrm{bild}(\pi)$$

mit $\mathrm{def}(\pi) \subseteq A$ und $\mathrm{bild}(\pi) \subseteq B$ heißt *partieller Isomorphismus* von $\mathfrak{A}$ nach $\mathfrak{B}$, falls gilt:

(a) π ist *bijektiv*,

(b) für jedes Konstantensymbol $c \in \sigma$ ist $c^{\mathfrak{A}} \in \mathrm{def}(\pi)$ und $\pi(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$, und

(c) für jedes Relationssymbol $R \in \sigma$ der Stelligkeit $k := ar(R)$ und alle k-Tupel $\vec{a} \in \mathrm{def}(\pi)^k$ gilt[1]

$$\vec{a} \;\in\; R^{\mathfrak{A}} \quad\Longleftrightarrow\quad \pi(\vec{a}) \;\in\; R^{\mathfrak{B}}\,.$$

(b) $\mathrm{Part}(\mathfrak{A}, \mathfrak{B})$ bezeichnet die Menge aller partiellen Isomorphismen von $\mathfrak{A}$ nach $\mathfrak{B}$.

[1] Zur Erinnerung: Für $\vec{a} = (a_1, \ldots, a_k)$ ist $\pi(\vec{a}) := \big(\pi(a_1), \ldots, \pi(a_k)\big)$.

3.9 Bemerkungen.

(a) Sei $\sigma = \{R_1, .., R_k, c_1, .., c_\ell\}$ eine Signatur und sei $r_i := ar(R_i)$ für jedes $i \in \{1, .., k\}$. Sei $\mathfrak{A}$ eine σ-Struktur und sei $U \subseteq A$ eine Teilmenge von A, die jede Konstante von $\mathfrak{A}$ enthält (d.h. $c_1^{\mathfrak{A}}, \ldots, c_\ell^{\mathfrak{A}} \in U$). Die *von U induzierte Substruktur von $\mathfrak{A}$* ist definiert als die σ-Struktur

$$\mathfrak{A}_{|U} := \big(U, (R_1^{\mathfrak{A}} \cap U^{r_1}), \ldots, (R_k^{\mathfrak{A}} \cap U^{r_k}),\ c_1^{\mathfrak{A}}, \cdots, c_\ell^{\mathfrak{A}}\big).$$

(b) Offensichtlich ist ein partieller Isomorphismus π von $\mathfrak{A}$ nach $\mathfrak{B}$ ein Isomorphismus von $\mathfrak{A}_{|\text{def}(\pi)}$ auf $\mathfrak{B}_{|\text{bild}(\pi)}$.

(c) Oft schreiben wir

$$\pi : a_1, .., a_m \mapsto b_1, .., b_m \qquad \text{bzw.} \qquad \pi : \big(a_i \mapsto b_i\big)_{i=1,..,m}$$

um die Abbildung π mit $\text{def}(\pi) = \{a_1, .., a_m\}$ und $\pi(a_i) = b_i$, für alle $i \in \{1, .., m\}$, zu bezeichnen.

(d) Oft identifizieren wir eine Abbildung π mit ihrem Graph

$$\{\, \big(a, \pi(a)\big) \;:\; a \in \text{def}(\pi) \,\}.$$

Insbesondere bedeutet dann

$$\pi \subseteq \pi',$$

dass π' eine *Erweiterung* von π ist (d.h. $\text{def}(\pi) \subseteq \text{def}(\pi')$ und für alle $a \in \text{def}(\pi)$ ist $\pi'(a) = \pi(a)$).

3.10 Proposition. *Sei σ eine Signatur. $\mathfrak{A}$ und $\mathfrak{B}$ seien σ-Strukturen.*

(a) $\pi := \varnothing$ (d.h. die Abbildung mit leerem Definitionsbereich) ist genau dann ein partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$, wenn σ keine Konstantensymbole enthält.

(b) Sei $\vec{a} = (a_1, .., a_m) \in A^m$ und $\vec{b} = (b_1, .., b_m) \in B^m$. Dann sind äquivalent:

(1) $\pi : a_1, .., a_m, (c^{\mathfrak{A}})_{c\in\sigma} \mapsto b_1, .., b_m, (c^{\mathfrak{B}})_{c\in\sigma}$ ist ein partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

(2) Für alle atomaren *σ-Formeln $\alpha(x_1, .., x_m)$ mit frei$(\alpha) \subseteq \{x_1, .., x_m\}$ gilt: $\mathfrak{A} \models \alpha[\vec{a}] \iff \mathfrak{B} \models \alpha[\vec{b}]$.*

(3) Für alle quantorenfreien FO$[\sigma]$*-Formeln $\varphi(x_1, .., x_m)$ mit frei$(\varphi) \subseteq \{x_1, .., x_m\}$ gilt: $\mathfrak{A} \models \varphi[\vec{a}] \iff \mathfrak{B} \models \varphi[\vec{b}]$.*

Beweis: Übung. □

Man beachte: Die Existenz eines partiellen Isomorphismus $\pi : a_1, .., a_m \mapsto b_1, .., b_m$ bedeutet *nicht* unbedingt, dass auch Formeln mit Quantoren erhalten werden. Ein partieller Isomorphismus sagt also etwas über $\equiv_0$ aus, aber nicht über $\equiv_m$ für $m > 0$.

3.1.2 Das m-Runden EF-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$

Sei σ eine Signatur und seien $m, k \in \mathbb{N}$. Seien $\mathfrak{A}$ und $\mathfrak{B}$ zwei σ-Strukturen und seien $\vec{a}' = a'_1, \ldots, a'_k \in A$ und $\vec{b}' = b'_1, \ldots, b'_k \in B$.

Spielregeln und Gewinnbedingung:

Das m-Runden EF-Spiel[2] $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$ wird von zwei Spielern, *Spoiler* und *Duplicator* (die manchmal auch Spieler I und Spieler II genannt werden) auf den beiden Strukturen $(\mathfrak{A}, \vec{a}')$ und $(\mathfrak{B}, \vec{b}')$ gespielt.

Spoilers Ziel: Zeige, dass die beiden Strukturen verschieden sind.
Genauer: Zeige, dass $(\mathfrak{A}, \vec{a}') \not\equiv_m (\mathfrak{B}, \vec{b}')$.

Duplicators Ziel: Vertusche einen (etwaigen) Unterschied zwischen den beiden Strukturen.
Genauer: Zeige, dass $(\mathfrak{A}, \vec{a}') \equiv_m (\mathfrak{B}, \vec{b}')$.

Eine *Partie* des Spiels besteht aus m *Runden*.
In jeder Runde $i \in \{1, \ldots, m\}$ geschieht folgendes:
Zunächst wählt Spoiler entweder ein Element aus A, das im Folgenden mit a_i bezeichnet wird, oder er wählt ein Element aus B, das im Folgenden mit b_i bezeichnet wird. Danach antwortet Duplicator mit einem Element aus dem Universum der anderen Struktur, d.h. er wählt ein Element $b_i \in B$, falls Spoiler ein $a_i \in A$ gewählt hat, bzw. ein Element $a_i \in A$, falls Spoiler ein $b_i \in B$ gewählt hat.

Nach der m-ten Runde wird der Gewinner ermittelt:
Duplicator hat gewonnen, falls die Abbildung

$$\pi \;:\; \left\{ \begin{array}{lcll} a_i & \mapsto & b_i & \text{für alle } i \in \{1, \ldots, m\} \\ c^{\mathfrak{A}} & \mapsto & c^{\mathfrak{B}} & \text{für alle Konstantensymbole } c \in \sigma \\ a'_j & \mapsto & b'_j & \text{für alle } j \in \{1, \ldots, k\} \end{array} \right\}$$

ein *partieller Isomorphismus* von $\mathfrak{A}$ nach $\mathfrak{B}$ ist.
Ansonsten hat *Spoiler* gewonnen.

Gewinnstrategien:

Eine *Strategie* für einen der beiden Spieler ist eine Vorschrift, die ihm sagt, welchen Zug er als nächstes machen soll. Formal:

Eine Strategie für Spoiler ist eine Abbildung

$$f_S \;:\; \bigcup_{i=0}^{m-1} \left(A^i \times B^i\right) \;\to\; A \dot{\cup} B.$$

[2] Im Fall $k = 0$ schreiben wir oft $\mathcal{G}_m(\mathfrak{A}, \mathfrak{B})$ an Stelle von $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$.

(Sind $\vec{a} := a_1, \ldots, a_i$ und $\vec{b} := b_1, \ldots, b_i$ die in den ersten i Runden gewählten Elemente, so gibt $f_S(\vec{a}, \vec{b})$ an, welches Element Spoiler in der $i{+}1$-ten Runde wählen soll.)

Eine Strategie für Duplicator ist eine Abbildung

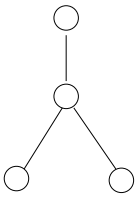
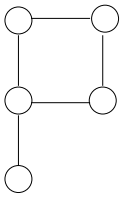
$$f_D \;:\; \bigcup_{i=0}^{m-1} \big((A^i \times B^i) \times (A \dot{\cup} B)\big) \;\to\; A \dot{\cup} B,$$

so dass für alle $\vec{a}, \vec{b}, c$ gilt: $f_D(\vec{a}, \vec{b}, c) \in B \iff c \in A$.
(Sind $\vec{a} := a_1, \ldots, a_i$ und $\vec{b} := b_1, \ldots, b_i$ die in den ersten i Runden gewählten Elemente und ist c das von Spoiler in der $i{+}1$-ten Runde gewählte Element, so gibt $f_D(\vec{a}, \vec{b}, c)$ an, mit welchem Element Duplicator in der $i{+}1$-ten Runde antworten soll.)

Eine *Gewinnstrategie* ist eine Strategie für einen der beiden Spieler, mit der er alle Partien des Spiels $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$ gewinnt.
Wir sagen: Spoiler (bzw. Duplicator) *gewinnt* $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$, falls er eine Gewinnstrategie im m-Runden EF-Spiel auf $(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$ hat.

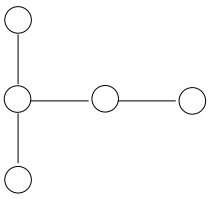
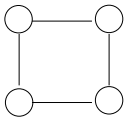
3.11 Beispiele.

(a) *Spoiler* gewinnt das 2-Runden EF-Spiel auf den folgenden Graphen

$\mathfrak{A}$:  $\mathfrak{B}$: 

indem er in Runde 1 denjenigen Knoten a_1 von Graph $\mathfrak{A}$ wählt, der mit allen anderen Knoten verbunden ist. In Runde 2 wählt der dann einen Knoten b_2 in $\mathfrak{B}$, der nicht zu Knoten b_1 benachbart ist.

(b) *Duplicator* gewinnt das 2-Runden EF-Spiel auf den folgenden Graphen

$\mathfrak{A}$:  $\mathfrak{B}$: 

denn in beiden Graphen gibt es zu jedem Knoten sowohl einen Nachbarn als auch einen Nicht-Nachbarn.

(c) *Spoiler* gewinnt das 3-Runden EF-Spiel auf den Graphen $\mathfrak{A}$ und $\mathfrak{B}$ aus (b), indem er in den ersten drei Runden drei verschiedene nicht benachbarte Knoten in $\mathfrak{A}$ wählt.

(d) Für $\mathfrak{A} := (\{0,..,7\}, <, 0, 7)$ und $\mathfrak{B} := (\{0,..,8\}, <, 0, 8)$ gilt: Duplicator gewinnt $\mathcal{G}_2(\mathfrak{A}, \mathfrak{B})$ und Spoiler gewinnt $\mathcal{G}_3(\mathfrak{A}, \mathfrak{B})$.

3.12 Satz.
Für jedes $m \geqslant 1$ und alle geordneten endlichen Strukturen $\mathfrak{A} = (A, <^{\mathfrak{A}}, 0^{\mathfrak{A}}, max^{\mathfrak{A}})$ und $\mathfrak{B} = (B, <^{\mathfrak{B}}, 0^{\mathfrak{B}}, max^{\mathfrak{B}})$ gilt:

$$\textit{Duplicator gewinnt das Spiel } \mathcal{G}_m(\mathfrak{A}, \mathfrak{B}) \iff |A| = |B| \text{ oder } |A|, |B| > 2^m.$$

Beweis: "$\Longleftarrow$": Falls $|A| = |B| := n$ ist, so können wir sowohl $\mathfrak{A}$ als auch $\mathfrak{B}$ mit der Struktur $(\{0,..,n-1\}, <, 0, n-1)$ identifizieren. Duplicator gewinnt das m-Runden Spiel, indem er in jeder Runde Spoilers Zug einfach "kopiert".
Im Folgenden betrachten wir den Fall, dass sowohl $|A|$ als auch $|B|$ größer als 2^m sind.
Für $\mathfrak{C} := \mathfrak{A}$ oder $\mathfrak{C} := \mathfrak{B}$ betrachten wir die folgende auf $C \times C$ definierte *Distanzfunktion*

$$\text{Dist}(c, c') \;:=\; |\{\, d \in C \;:\; (c <^{\mathfrak{C}} d \leqslant^{\mathfrak{C}} c') \text{ oder } (c' <^{\mathfrak{C}} d \leqslant^{\mathfrak{C}} c)\}|\,.$$

Wir zeigen nun, dass Duplicator so spielen kann, dass für jedes $i \in \{0,..,m\}$ die folgende Invariante $(*)_i$ erfüllt ist:

$(*)_i$: Sind $a_1,..,a_i$ und $b_1,..,b_i$ die in den Runden $1,..,i$ gewählten Elemente in A und B und ist $a_0 := 0^{\mathfrak{A}}$, $a_{max} := max^{\mathfrak{A}}$, $b_0 := 0^{\mathfrak{B}}$ und $b_{max} := max^{\mathfrak{B}}$, so gilt für alle $j, j' \in \{0, max, 1,..,i\}$:

(a) $a_j <^{\mathfrak{A}} a_{j'} \iff b_j <^{\mathfrak{B}} b_{j'}$ und

(b) $\text{Dist}(a_j, a_{j'}) = \text{Dist}(b_j, b_{j'})$ oder $\text{Dist}(a_j, a_{j'}),\ \text{Dist}(b_j, b_{j'}) \geqslant 2^{m-i}$.

Der Beweis folgt per Induktion nach i.

$\underline{i = 0:}$ Die Bedingung $(*)_0$ ist erfüllt, da $\text{Dist}(a_0, a_{max}) = |A|-1 \geqslant 2^m$ und $\text{Dist}(b_0, b_{max}) = |B| - 1 \geqslant 2^m$.

$\underline{i \to i+1:}$ Gemäß Induktionsannahme sind bereits i Runden gespielt und die Bedingung $(*)_i$ ist nach der i-ten Runde erfüllt.

Fall 1: Spoiler wählt in der $i+1$-ten Runde ein Element a_{i+1} in A.
Falls $a_{i+1} = a_j$ für ein $j \in \{0, max, 1,..,i\}$, so antwortet Duplicator mit $b_{i+1} := b_j$ und hat damit bewirkt, dass die Bedingung $(*)_{i+1}$ gilt.
Ansonsten gibt es Indices $j, j' \in \{0, max, 1,..,i\}$ so, dass $a_j <^{\mathfrak{A}} a_{i+1} <^{\mathfrak{A}} a_{j'}$ und für alle $j'' \in \{0, max, 1,..,i\}$ gilt: $a_{j''} \leqslant^{\mathfrak{A}} a_j$ oder $a_{j'} \leqslant^{\mathfrak{A}} a_{j''}$.
Da die Bedingung $(*)_i$ gemäß Induktionsannahme erfüllt ist, gilt

(1.) $\text{Dist}(a_j, a_{j'}) = \text{Dist}(b_j, b_{j'})$ oder

(2.) $\text{Dist}(a_j, a_{j'}),\ \text{Dist}(b_j, b_{j'}) \geqslant 2^{m-i}$.

In Fall (1.) gibt es ein Element b_{i+1} in B, so dass $b_j <^{\mathfrak{B}} b_{i+1} <^{\mathfrak{B}} b_{j'}$ und $\text{Dist}(b_j, b_{i+1}) = \text{Dist}(a_j, a_{i+1})$ und $\text{Dist}(b_{i+1}, b_{j'}) = \text{Dist}(a_{i+1}, a_{j'})$. Offensichtlich ist die Bedingung $(*)_{i+1}$ erfüllt, wenn Duplicator in der $i+1$-ten Runde dieses b_{i+1} wählt.

In Fall (2.) muss es mindestens ein Element c in B geben, so dass $b_j <^{\mathfrak{B}} c <^{\mathfrak{B}} b_{j'}$ und $\text{Dist}(b_j, c) \geqslant \frac{2^{m-i}}{2} = 2^{m-(i+1)}$ und $\text{Dist}(c, b_{j'}) \geqslant \frac{2^{m-i}}{2} = 2^{m-(i+1)}$.

- Falls $\text{Dist}(a_j, a_{i+1}) \geqslant 2^{m-(i+1)}$ und $\text{Dist}(a_{i+1}, a_{j'}) \geqslant 2^{m-(i+1)}$, so wählt Duplicator in der $i+1$-ten Runde $b_{i+1} := c$.

- Falls $\text{Dist}(a_j, a_{i+1}) < 2^{m-(i+1)}$, so wählt Duplicator das $b_{i+1} >^{\mathfrak{B}} b_j$ mit $\text{Dist}(b_j, b_{i+1}) = \text{Dist}(a_j, a_{i+1})$.

- Falls $\text{Dist}(a_{i+1}, a_{j'}) < 2^{m-(i+1)}$, so wählt Duplicator das $b_{i+1} <^{\mathfrak{B}} b_{j'}$ mit $\text{Dist}(b_{i+1}, b_{j'}) = \text{Dist}(a_{i+1}, a_{j'})$.

Man kann leicht nachprüfen, dass in jedem der 3 Fälle die Bedingung $(*)_{i+1}$ erfüllt ist.

Fall 2: Spoiler wählt in der $i+1$-ten Runde ein Element b_{i+1} in B.
Duplicators Antwort a_{i+1} in A wird analog zu Fall 1 ermittelt.

Damit sind wir fertig mit dem Induktionsschritt. Wir haben also bewiesen, dass Duplicator so spielen kann, dass nach jeder Runde $i \in \{0, \ldots, m\}$ die Bedingung $(*)_i$ erfüllt ist. Insbesondere ist nach m Runden die Bedingung $(*)_m$ erfüllt und Duplicator hat daher die Partie gewonnen.

"$\Longrightarrow$": Offensichtlich genügt es, folgendes zu zeigen: Falls $|A| < |B|$ und $|A| \leqslant 2^m$, so hat *Spoiler* eine Gewinnstrategie im m-Runden EF-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$. Dies kann man beweisen, indem man zeigt, dass Spoiler so spielen kann, dass für jedes $i \in \{0, 1, \ldots, m\}$ die folgende Invariante $(**)_i$ erfüllt ist:

$(**)_i$: Sind $a_1, \ldots, a_i$ und $b_1, \ldots, b_i$ die in den Runden $1, \ldots, i$ gewählten Elemente in A und B und ist $a_0 := 0^{\mathfrak{A}}$, $a_{max} := max^{\mathfrak{A}}$, $b_0 := 0^{\mathfrak{B}}$ und $b_{max} := max^{\mathfrak{B}}$, so gibt es $j, j' \in \{0, max, 1, \ldots, i\}$, so dass

(a) $\big(a_j <^{\mathfrak{A}} a_{j'}$ und $b_j \geqslant^{\mathfrak{B}} b_{j'} \big)$ oder $\big(a_j \geqslant^{\mathfrak{A}} a_{j'}$ und $b_j <^{\mathfrak{B}} b_{j'} \big)$ oder

(b) $\text{Dist}(a_j, a_{j'}) < 2^{m-i}$ und $\text{Dist}(a_j, a_{j'}) < \text{Dist}(b_j, b_{j'})$.

Details: *Übung*. □

3.13 Bemerkung. Per Induktion nach m kann man leicht zeigen, dass für alle $m \in \mathbb{N}$, alle Signaturen σ und alle σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ gilt: genau einer der beiden Spieler (Spoiler oder Duplicator) hat eine Gewinnstrategie im Spiel $\mathcal{G}_m(\mathfrak{A}, \mathfrak{B})$ (Details: *Übung*).

3.1.3 Der Satz von Ehrenfeucht

Ziel dieses Abschnitts ist es, zu zeigen, dass

$$\mathfrak{A} \equiv_m \mathfrak{B} \iff \text{Duplicator gewinnt das } m\text{-Runden EF-Spiel auf } \mathfrak{A} \text{ und } \mathfrak{B}.$$

Wegen Proposition 3.7 kann man also EF-Spiele verwenden, um zu zeigen, dass bestimmte Probleme nicht FO-definierbar sind.

3.14 Definition. Sei σ eine Signatur, $\mathfrak{A}$ eine σ-Struktur, $k \in \mathbb{N}$, $\vec{a} = a_1, \ldots, a_k$ eine Sequenz von Elementen aus A und $\vec{x} = x_1, \ldots, x_k$ eine Sequenz von k verschiedenen Variablen erster Stufe. Wir definieren induktiv für jedes $m \in \mathbb{N}$ eine FO-Formel $\varphi^m_{\mathfrak{A},\vec{a}}(\vec{x})$ der Quantorentiefe m wie folgt:[3]

$$\varphi^0_{\mathfrak{A},\vec{a}}(\vec{x}) \;:=\; \bigwedge \left\{ \psi(\vec{x}) \;\middle|\; \begin{array}{l} \psi \text{ ist atomare oder negierte atomare } \sigma\text{-Formel mit} \\ \mathit{frei}(\psi) \subseteq \{x_1, \ldots, x_k\} \text{ und } \mathfrak{A} \models \psi[\vec{a}] \end{array} \right\}$$

und für $m > 0$

$$\varphi^m_{\mathfrak{A},\vec{a}}(\vec{x}) \;:=\; \bigwedge_{a \in A} \exists x_{k+1}\, \varphi^{m-1}_{\mathfrak{A},\vec{a}a}(\vec{x}, x_{k+1}) \;\wedge\; \forall x_{k+1} \bigvee_{a \in A} \varphi^{m-1}_{\mathfrak{A},\vec{a}a}(\vec{x}, x_{k+1})\,.$$

Die Formel $\varphi^m_{\mathfrak{A},\vec{a}}(\vec{x})$ heißt *m-Isomorphietyp* (oder *m-Hintikka-Formel*) von $\vec{a}$ in $\mathfrak{A}$.

3.15 Bemerkungen.

(a) In der obigen Definition ist $k{=}0$ erlaubt. Der m-Isomorphietyp ist dann ein *Satz* $\varphi^m_{\mathfrak{A}}$.

(b) Für alle $k, m \in \mathbb{N}$ ist die Menge

$$m\text{-Typen}_k \;:=\; \{\, \varphi^m_{\mathfrak{A},\vec{a}}(\vec{x}) \;:\; \mathfrak{A} \text{ ist eine } \sigma\text{-Struktur und } \vec{a} \in A^k \,\}$$

endlich. (Für $m{=}0$ gilt das, da es nur endlich viele verschiedene *atomare* σ-Formeln über den Variablen $x_1, \ldots, x_k$ gibt. Für $m > 0$ folgt die Endlichkeit dann per Induktion.)

(c) $\mathfrak{A} \models \varphi^m_{\mathfrak{A},\vec{a}}[\vec{a}]$. (Das folgt leicht per Induktion nach m.)

3.16 Theorem (Satz von Ehrenfeucht, 1961)**.** *Sei σ eine Signatur. $\mathfrak{A}$ und $\mathfrak{B}$ seien σ-Strukturen, $k, m \in \mathbb{N}$ und $\vec{a}' = a'_1, \ldots, a'_k \in A$ und $\vec{b}' = b'_1, \ldots, b'_k \in B$. Dann sind äquivalent:*

(a) Duplicator gewinnt $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$.

(b) $\mathfrak{B} \;\models\; \varphi^m_{\mathfrak{A},\vec{a}'}[\vec{b}']$.

(c) $(\mathfrak{A}, \vec{a}') \equiv_m (\mathfrak{B}, \vec{b}')$,
d.h. für alle FO$[\sigma]$*-Formeln $\psi(x_1, \ldots, x_k)$ mit $\mathit{frei}(\psi) \subseteq \{x_1, \ldots, x_k\}$ und $\mathit{qr}(\psi) \leqslant m$ gilt:* $\mathfrak{A} \models \psi[\vec{a}'] \iff \mathfrak{B} \models \psi[\vec{b}']$.

[3] Für eine endliche Menge M von Formeln schreiben wir $\bigwedge M$, um die Formel $\bigwedge_{\psi \in M} \psi$ zu bezeichnen.

Beweis: "(c) $\Longrightarrow$ (b)": Es gilt $qr(\varphi^m_{\mathfrak{A},\vec{a}'}) = m$ und $\mathfrak{A} \models \varphi^m_{\mathfrak{A},\vec{a}'}[\vec{a}]$. Aus (c) folgt daher $\mathfrak{B} \models \varphi^m_{\mathfrak{A},\vec{a}'}[\vec{b}']$.

"(b) $\Longleftrightarrow$ (a)": Beweis per Induktion nach m.

$\underline{m{=}0}$:

Duplicator gewinnt $\mathcal{G}_0(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$

$\overset{\text{Gewinnbed.}}{\Longleftrightarrow}$ $\pi : \vec{a}', (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}', (c^{\mathfrak{B}})_{c\in\sigma} \in \text{Part}(\mathfrak{A}, \mathfrak{B})$

$\overset{\text{Wahl von } \varphi^0_{\mathfrak{A},\vec{a}'}}{\Longleftrightarrow}$ $\mathfrak{B} \models \varphi^0_{\mathfrak{A},\vec{a}'}[\vec{b}']$.

$\underline{m \mapsto m{+}1}$:

Duplicator gewinnt $\mathcal{G}_{m+1}(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$

$\Longleftrightarrow$ für alle $a \in A$ ex. $b \in B$, so dass Duplicator $\mathcal{G}_m(\mathfrak{A}, \vec{a}'a, \mathfrak{B}, \vec{b}'b)$ gewinnt, und
für alle $b \in B$ ex. $a \in A$, so dass Duplicator $\mathcal{G}_m(\mathfrak{A}, \vec{a}'a, \mathfrak{B}, \vec{b}'b)$ gewinnt

$\overset{\text{Ind.ann.}}{\Longleftrightarrow}$ für alle $a \in A$ gibt es ein $b \in B$, so dass $\mathfrak{B} \models \varphi^m_{\mathfrak{A},\vec{a}'a}[\vec{b}', b]$, und
für alle $b \in B$ gibt es ein $a \in A$, so dass $\mathfrak{B} \models \varphi^m_{\mathfrak{A},\vec{a}'a}[\vec{b}', b]$.

$\Longleftrightarrow$ $\mathfrak{B} \models \Big(\bigwedge\limits_{a\in A} \exists x_{k+1}\, \varphi^m_{\mathfrak{A},\vec{a}'a}(\vec{x}, x_{k+1}) \;\wedge\; \forall x_{k+1} \bigvee\limits_{a\in A} \varphi^m_{\mathfrak{A},\vec{a}'a}(\vec{x}, x_{k+1}) \Big)[\vec{b}']$

$\overset{\text{Def. } \varphi^{m+1}_{\mathfrak{A},\vec{a}'}}{\Longleftrightarrow}$ $\mathfrak{B} \models \varphi^{m+1}_{\mathfrak{A},\vec{a}'}[\vec{b}']$.

Somit sind wir fertig mit dem Beweis von "(b) $\Longleftrightarrow$ (a)".

"(a) $\Longrightarrow$ (c)": Per Induktion nach m.

$\underline{m{=}0}$: Wie bei "(b) $\Longleftrightarrow$ (a)".

$\underline{m \mapsto m{+}1}$: Gemäß Voraussetzung gewinnt Duplicator $\mathcal{G}_{m+1}(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$. Sei $\psi(\vec{x})$ eine FO-Formel mit $\mathit{frei}(\psi) \subseteq \{x_1, .., x_k\}$ und $qr(\psi) \leqslant m{+}1$. Wir müssen zeigen, dass

$$(*) \qquad \mathfrak{A} \models \psi[\vec{a}'] \iff \mathfrak{B} \models \psi[\vec{b}'].$$

Klar: Die Menge aller Formeln ψ, die die Bedingung $(*)$ erfüllen, ist abgeschlossen unter Booleschen Kombinationen $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ und enthält gemäß Induktionsannahme alle Formeln der Quantorentiefe $\leqslant m$.
Wir müssen daher o.B.d.A. nur noch den Fall betrachten, dass ψ von der Form

$$\psi \;=\; \exists x_{k+1}\, \chi(\vec{x}, x_{k+1})$$

ist, wobei χ eine Formel der Quantorentiefe m ist.

Wir betrachten nur den Fall, dass $\mathfrak{A} \models \psi[\vec{a}']$ und zeigen, dass dann auch $\mathfrak{B} \models \psi[\vec{b}']$. (Der andere Fall kann analog behandelt werden.)

Wegen $\mathfrak{A} \models \psi[\vec{a}']$ gibt es ein $a \in A$ mit $\mathfrak{A} \models \chi[\vec{a}', a]$. Da Duplicator $\mathcal{G}_{m+1}(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$ gewinnt, muss es ein $b \in B$ geben, so dass Duplicator das Spiel $\mathcal{G}_m(\mathfrak{A}, \vec{a}'a, \mathfrak{B}, \vec{b}'b)$ gewinnt. Gemäß Induktionsannahme gilt dann $\mathfrak{A} \models \chi[\vec{a}', a] \iff \mathfrak{B} \models \chi[\vec{b}', b]$. Es gilt also $\mathfrak{B} \models \chi[\vec{b}', b]$ und daher $\mathfrak{B} \models \exists x_{k+1} \chi\,[\vec{b}']$, d.h. $\mathfrak{B} \models \psi[\vec{b}'\,]$. □

3.17 Bemerkung. Sind $\mathfrak{A}$ und $\mathfrak{B}$ zwei Strukturen, die sich durch einen FO-Satz φ der Quantorentiefe m unterscheiden lassen (also $\mathfrak{A} \models \varphi$ und $\mathfrak{B} \not\models \varphi$), so gibt es gemäß Theorem 3.16 und Bemerkung 3.13 eine Gewinnstrategie für Spoiler im m-Runden EF-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$. Der Satz φ gibt sogar direkt eine solche Gewinnstrategie an: Spoiler gewinnt das Spiel, indem er Elemente in $\mathfrak{A}$ wählt, die den $\exists$-Quantoren in φ entsprechen, und Elemente in $\mathfrak{B}$, die den $\forall$-Quantoren in φ entsprechen.
Sind beispielsweise $\mathfrak{A}$ und $\mathfrak{B}$ die Graphen aus Beispiel 3.11 (a), so ist

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

ein Satz mit $\mathfrak{A} \models \varphi$ und $\mathfrak{B} \not\models \varphi$, d.h. es gilt

$$\mathfrak{A} \models \exists x\, \forall y\, \big(x{=}y \vee E(x,y)\big) \quad \text{und} \quad \mathfrak{B} \models \forall x\, \exists y\, \big(x{\neq}y \wedge \neg E(x,y)\big)\,.$$

Spoiler wählt in Runde 1 ein $a_1 \in A$, so dass

$$\mathfrak{A} \models \Big(\forall y\, \big(x{=}y \vee E(x,y)\big)\Big)[a_1]$$

(ein solches Element gibt es, weil $\mathfrak{A} \models \varphi$).
Da $\mathfrak{B} \not\models \varphi$, muss für *jede* beliebige Antwort $b_1 \in B$ von Duplicator gelten:

$$\mathfrak{B} \models \Big(\exists y\, \big(x{\neq}y \wedge \neg E(x,y)\big)\Big)[b_1].$$

In der zweiten Runde kann Spoiler daher ein Element $b_2 \in B$ auswählen, für das gilt:

$$\mathfrak{B} \models \Big(x{\neq}y \wedge \neg E(x,y)\Big)[b_1, b_2].$$

Für jede mögliche Antwort $a_2 \in A$, die Duplicator geben kann, gilt

$$\mathfrak{A} \models \Big(x{=}y \vee E(x,y)\Big)[a_1, a_2].$$

Daher kann die Abbildung $a_1, a_2 \mapsto b_1, b_2$ kein partieller Isomorphismus sein, Duplicator hat die Partie also verloren.

3.18 Bemerkung. Aus Theorem 3.16 und Bemerkung 3.15 (b) folgt direkt, dass für jedes $m \in \mathbb{N}$ und jede Signatur σ die Relation $\equiv_m$ nur *endlich* viele Äquivalenzklassen auf der Klasse aller σ-Strukturen hat. Die Äquivalenzklasse, zu der eine gegebene Struktur $\mathfrak{A}$ gehört, wird dabei durch den Satz $\varphi^m_{\mathfrak{A}}$ definiert.

Die Klasse aller σ-Strukturen ist also eine Vereinigung von *endlich* vielen Äquivalenzklassen von $\equiv_m$.

Für jeden FO-Satz ψ der Quantorentiefe m ist $\mathrm{Mod}_{\mathsf{All}}(\psi)$ natürlich eine Vereinigung von Äquivalenzklassen von $\equiv_m$. Insbesondere folgt daraus direkt, dass es für jedes m nur *endlich* viele nicht-äquivalente FO-Formeln der Quantorentiefe m gibt.

Die nächste Folgerung aus dem Satz von Ehrenfeucht wird oft benutzt um zu zeigen, dass bestimmte Probleme *nicht* FO-definierbar sind.

3.19 Korollar. *Ist σ eine Signatur, S eine Klasse von σ-Strukturen und $\mathsf{C} \subseteq \mathsf{S}$, so sind äquivalent:*

(a) C *ist nicht* FO*-definierbar in* S*.*

(b) Für alle $m \in \mathbb{N}$ gibt es $\mathfrak{A}_m \in \mathsf{C}$ und $\mathfrak{B}_m \in \mathsf{S} \setminus \mathsf{C}$, so dass $\mathfrak{A}_m \equiv_m \mathfrak{B}_m$.

(c) Für alle $m \in \mathbb{N}$ gibt es $\mathfrak{A}_m \in \mathsf{C}$ und $\mathfrak{B}_m \in \mathsf{S} \setminus \mathsf{C}$, so dass Duplicator das m-Runden EF-Spiel auf $\mathfrak{A}_m$ und $\mathfrak{B}_m$ gewinnt.

Beweis: "(b) $\Longleftrightarrow$ (c)": Folgt direkt aus Theorem 3.16.

"(b) $\Longrightarrow$ (a)": Das ist gerade die Aussage von Proposition 3.7.

"(a) $\Longrightarrow$ (b)": Angenommen (b) gilt nicht. Dann gibt es ein $m \in \mathbb{N}$, so dass für alle Strukturen $\mathfrak{A}, \mathfrak{B} \in \mathsf{S}$ gilt: Falls $\mathfrak{A} \in \mathsf{C}$ und $\mathfrak{A} \equiv_m \mathfrak{B}$, so $\mathfrak{B} \in \mathsf{C}$. Die Klasse C ist also eine Vereinigung von Äquivalenzklassen von $\equiv_m$ und wird daher durch den FO-Satz

$$\psi \;:=\; \bigvee \{\, \varphi^m_{\mathfrak{A}} \;:\; \mathfrak{A} \text{ ist eine } \sigma\text{-Struktur mit } \mathfrak{A} \in \mathsf{C} \,\}$$

definiert. Es gilt also: $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\psi)$. □

Als Anwendung erhalten wir, dass die folgenden Probleme nicht FO-definierbar sind:

3.20 Satz. *Für die Strukturklassen*

$\mathsf{S}_<$	:	*Klasse aller endlichen linearen Ordnungen $\mathfrak{A} = (A, <^{\mathfrak{A}})$,*
$\mathsf{Even}_<$	:	*Klasse aller endlichen linearen Ordnungen gerader Länge, d.h. aller linearen Ordnungen $(A, <^{\mathfrak{A}})$, bei denen $\lvert A\rvert$ gerade ist*
UGraphs	:	*Klasse aller endlichen ungerichteten Graphen,*[4]
Conn	:	*Klasse aller endlichen ungerichteten zusammenhängenden Graphen,*
RGraphs	:	*Klasse aller endlichen Graphen $G = (V, E^G, s^G, t^G)$ mit $s^G, t^G \in V$,*
Reach	:	*Klasse aller $G \in$ RGraphs, in denen es einen Pfad vom Knoten s^G zum Knoten t^G gibt*

gilt:

(a) $\mathsf{Even}_<$ *ist nicht* FO*-definierbar in* $\mathsf{S}_<$*.*

(b) Conn *ist nicht* FO*-definierbar in* UGraphs*.*

[4] Ein Graph $G = (V, E)$ heißt *ungerichtet*, falls für alle $v, w \in V$ gilt: $(v, v) \notin E$ und falls $(v, w) \in E$, so auch $(w, v) \in E$.

(c) Reach *ist nicht* FO*-definierbar in* RGraphs.

Beweis: (a): Für jedes $m \in \mathbb{N}$ sei $\mathfrak{A}_m$ eine lineare Ordnung auf 2^m+2 Elementen und $\mathfrak{B}_m$ eine lineare Ordnung auf 2^m+1 Elementen. Somit ist $\mathfrak{A}_m \in \mathsf{Even}_<$ und $\mathfrak{B}_m \in \mathsf{S}_< \setminus \mathsf{Even}_<$. Von Satz 3.12 wissen wir, dass Duplicator das m-Runden EF-Spiel auf $\mathfrak{A}_m$ und $\mathfrak{B}_m$ gewinnt. Gemäß Korollar 3.19 ist daher $\mathsf{Even}_<$ nicht FO-definierbar in $\mathsf{S}_<$.

(b): Durch Widerspruch. Angenommen, φ ist ein FO$[E]$-Satz, so dass für jeden ungerichteten Graphen $G = (V^G, E^G)$ gilt: $G \models \varphi \iff G$ ist zusammenhängend.

Idee: Nutze diesen Satz φ, um einen FO$[<]$-Satz ψ zu konstruieren, so dass für jede lineare Ordnung $\mathfrak{A} = (A, <^{\mathfrak{A}})$ gilt: $\mathfrak{A} \models \psi \iff |A|$ ist gerade.
Aus (a) wissen wir schon, dass es einen solchen Satz ψ nicht gibt.

Um den Satz ψ zu konstruieren, ordnen wir jeder linearen Ordnung $\mathfrak{A}$ mit $A = \{a_1 <^{\mathfrak{A}} \cdots <^{\mathfrak{A}} a_n\}$ (für beliebiges n) den Graphen $G_{\mathfrak{A}}$ zu, dessen Knotenmenge die Menge A ist, und dessen Kantenmenge aus genau den Kanten zwischen a_i und a_{i+2}, für alle $i \leqslant n-2$, sowie einer zusätzlichen Kante zwischen a_1 und a_n besteht. Man sieht leicht:

$$G_{\mathfrak{A}} \text{ ist zusammenhängend} \iff |A| \text{ ist gerade.} \tag{3.1}$$

Sei nun $\chi_E(x, y)$ eine FO$[<]$-Formel, die besagt

$$\text{“}y = x + 2\text{”} \text{ oder } \text{“}x = y + 2\text{”} \text{ oder}$$
$$\big(\text{“}x = \textit{min}\text{” und “}y = \textit{max}\text{”}\big) \text{ oder } \big(\text{“}y = \textit{min}\text{” und “}x = \textit{max}\text{”}\big)$$

(klar: eine solche FO$[<]$-Formel $\chi_E(x, y)$ lässt sich leicht konstruieren).

Ausgewertet in einer linearen Ordnung $\mathfrak{A}$ “simuliert” die Formel $\chi_E(x, y)$ gewissermaßen die Kantenrelation des Graphen $G_{\mathfrak{A}}$.

Sei nun ψ der FO$[<]$-Satz, der aus dem FO$[E]$-Satz φ entsteht, indem jedes Atom der Form $E(x, y)$ durch die FO$[<]$-Formel $\chi_E(x, y)$ ersetzt wird. Der Satz ψ ist also gerade so konstruiert, dass beim Auswerten von ψ in einer linearen Ordnung $\mathfrak{A}$ die Auswertung des Satzes φ in dem Graphen $G_{\mathfrak{A}}$ simuliert wird. Es gilt also: $\mathfrak{A} \models \psi \iff G_{\mathfrak{A}}$ ist zusammenhängend. Mit (3.1) folgt daher für jede lineare Ordnung $\mathfrak{A}$, dass

$$\mathfrak{A} \models \psi \iff G_{\mathfrak{A}} \text{ ist zusammenhängend} \iff |A| \text{ ist gerade.}$$

Dies ist aber ein Widerspruch zu (a).

(c): Folgt ähnlich wie (b) aus (a). Details: *Übung*. □

3.21 Bemerkung (logische Reduktionen)**.** Die im Beweis von (b) benutzte Vorgehensweise ist unter dem Begriff *logische Reduktion* bekannt. Im obigen Beweis wurde beispielsweise das Problem, einen FO-Satz zu finden, der ausdrückt, dass eine lineare Ordnung *gerade* viele Elemente besitzt, auf das Problem reduziert, einen FO-Satz zu finden, der Graph-Zusammenhang ausdrückt — d.h. es wurde gezeigt: falls Graph-Zusammenhang

FO-definierbar ist, so ist auch die Aussage "eine lineare Ordnung hat gerade Länge" FO-definierbar. Dies wurde dadurch erreicht, dass man innerhalb einer linearen Ordnung einen geeigneten Graphen "simuliert" (oder *interpretiert*), indem man die Kantenrelation des Graphen durch eine FO-Formel beschreibt.

Generell ist diese Methode der *logischen Reduktionen* (oder *logischen Interpretationen* oft nützlich, um bereits bekannte Nicht-Definierbarkeits-Resultate (beispielsweise das Resultat aus Satz 3.20 (a)) auf neue Nicht-Definierbarkeits-Resultate zu übertragen (beispielsweise die Resultate aus Satz 3.20 (b) und (c)).

Details zum Thema "Interpretationen und logische Reduktionen" werden in Kapitel 3.4 behandelt (das in der Vorlesung im Sommersemester 2007 allerdings nicht besprochen wird).

3.1.4 Der Satz von Hanf

Der Satz von Hanf liefert ein hinreichendes Kriterium für die m-Äquivalenz zweier Strukturen, so dass man durch eine einfache Anwendung dieses Satzes oft ganz leicht zeigen kann, dass $\mathfrak{A} \equiv_m \mathfrak{B}$, ohne dabei explizit eine Gewinnstrategie für das m-Runden EF-Spiel konstruieren zu müssen. Bevor wir die exakte Formulierung des Satzes von Hanf angeben können, benötigen wir allerdings ein paar Notationen:

3.22 Definition (Gaifman-Graph, Distanzfunktion, Nachbarschaft)**.**
Sei σ eine Signatur und $\mathfrak{A}$ eine σ-Struktur.

(a) Der *Gaifman-Graph* $\mathcal{G}(\mathfrak{A})$ von $\mathfrak{A}$ ist der ungerichtete Graph mit Knotenmenge $V^{\mathcal{G}(\mathfrak{A})} := A$ und Kantenmenge

$$E^{\mathcal{G}(\mathfrak{A})} := \left\{ (u,v) \;\middle|\; \begin{array}{l} u \neq v \text{ und es gibt ein } R \in \sigma \text{ und ein Tupel } (a_1,..,a_r) \in R^{\mathfrak{A}}, \\ \text{für } r := \mathit{ar}(R), \text{ so dass } u,v \in \{a_1,..,a_r\} \end{array} \right\}.$$

(b) Die *Distanzfunktion* $\mathrm{Dist}^{\mathfrak{A}}(\cdot,\cdot) : A \times A \to \mathbb{N}$ ist definiert durch

$$\mathrm{Dist}^{\mathfrak{A}}(u,v) := \begin{cases} 0\,, & \text{falls } u = v, \\ \infty\,, & \text{falls } u \neq v \text{ und es in } \mathcal{G}(\mathfrak{A}) \text{ keinen Pfad von } u \text{ nach } v \text{ gibt}, \\ \min\{\ell \in \mathbb{N} : \text{es gibt in } \mathcal{G}(\mathfrak{A}) \text{ einen Pfad der Länge } \ell \text{ von } u \text{ nach } v\}, & \text{sonst.} \end{cases}$$

(c) Für ein Element $a \in A$ und eine Zahl $r \in \mathbb{N}$ ist die r-*Umgebung* (oder: r-*Nachbarschaft*) von a die Menge

$$N_r^{\mathfrak{A}}(a) := \{b \in A : \mathrm{Dist}^{\mathfrak{A}}(a,b) \leqslant r\}.$$

Wir schreiben $\mathcal{N}_r^{\mathfrak{A}}(a)$ für die durch die Menge $N_r^{\mathfrak{A}}(a)$ induzierte Substruktur von $\mathfrak{A}$, wobei ggf. in σ vorkommende Konstantensymbole ignoriert werden. D.h.

$$\mathcal{N}_r^{\mathfrak{A}}(a) := \left(N_r^{\mathfrak{A}}(a),\ \left(R^{\mathfrak{A}} \cap N_r^{\mathfrak{A}}(a)^{\mathit{ar}(R)}\right)_{R \in \sigma} \right).$$

Insbesondere gilt: Ist $\sigma = \{R_1, .., R_k, c_1, .., c_\ell\}$, so ist $\mathcal{N}_r^{\mathfrak{A}}(a)$ eine $\{R_1, .., R_k\}$-Struktur.

3.23 Bemerkung. Für zwei beliebige Elemente $b, b' \in N_r^{\mathfrak{A}}(a)$ ist $\text{Dist}^{\mathfrak{A}}(b, b') \leqslant 2{\cdot}r$.

3.24 Definition (r-Umgebungstypen)**.**
Sei σ eine Signatur, $\mathfrak{A}$ eine σ-Struktur, $r \in \mathbb{N}$ und $a \in A$.

(a) Der *r-Umgebungstyp* $\text{Typ}(r, a, \mathfrak{A})$ *von a in $\mathfrak{A}$* ist folgendermaßen definiert:

$$\text{Typ}(r, a, \mathfrak{A}) \;:=\; \Big(\mathcal{N}_r^{\mathfrak{A}}(a),\; a,\; C\Big)$$

wobei $C := \{\, (c, c^{\mathfrak{A}}) \,:\, c \in \sigma \text{ mit } c^{\mathfrak{A}} \in N_r^{\mathfrak{A}}(a) \,\}$.

Für einen r-Umbegungstyp $\rho := \text{Typ}(r, a, \mathfrak{A})$ schreiben wir oft $C(\rho)$ um die zu ρ gehörende Menge C zu bezeichnen.

(b) Sei $\mathfrak{B}$ eine σ-Struktur und sei $b \in B$. Sei $\rho_{\mathfrak{A}} := \text{Typ}(r, a, \mathfrak{A})$ und $\rho_{\mathfrak{B}} := \text{Typ}(r, b, \mathfrak{B})$. Die beiden Typen $\rho_{\mathfrak{A}}$ und $\rho_{\mathfrak{B}}$ heißen *isomorph* (kurz: $\rho_{\mathfrak{A}} \cong \rho_{\mathfrak{B}}$), falls die beiden folgenden Bedingungen erfüllt sind:

(1) Für jedes $c \in \sigma$ gilt: Es gibt in $C(\rho_{\mathfrak{A}})$ genau dann ein Tupel, dessen erste Komponente c ist, wenn es in $C(\rho_{\mathfrak{B}})$ ein Tupel gibt, dessen erste Komponente c ist.

Falls dies der Fall ist, seien $c_1, .., c_s$ (für ein geeignetes $s \geqslant 0$) gerade diejenigen Konstantensymbole aus σ, die als erste Komponente eines Tupels in $C(\rho_{\mathfrak{A}})$ bzw. $C(\rho_{\mathfrak{B}})$ vorkommen.

(2) $\Big(\mathcal{N}_r^{\mathfrak{A}}(a),\; a,\; c_1^{\mathfrak{A}}, \ldots, c_s^{\mathfrak{A}}\Big) \;\cong\; \Big(\mathcal{N}_r^{\mathfrak{B}}(b),\; b,\; c_1^{\mathfrak{B}}, \ldots, c_s^{\mathfrak{B}}\Big)$

Beachte: Die Ausdrücke links und rechts des "$\cong$"-Zeichens sind hier Strukturen der Signatur σ', die aus sämtlichen Relationssymbolen von σ sowie aus den Konstantensymbolen $c_1, .., c_s$ und einem zusätzlichen Konstantensymbol $\hat{c}$ besteht, das mit dem Element a bzw. b interpretiert wird.

(c) Ist ρ ein r-Umgebungstyp, so bezeichnet

$$\#_\rho(\mathfrak{A}) \;:=\; |\{a \in A \,:\, \text{Typ}(r, a, \mathfrak{A}) \cong \rho\}|$$

die Anzahl der Elemente in A, deren r-Umgebungstyp isomorph zu ρ ist.

3.25 Theorem (Satz von Hanf, 1965)**.**
Sei σ eine Signatur, seien $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen und sei $m \in \mathbb{N}$. Falls gilt

(a) es gibt eine Zahl $e \in \mathbb{N}$, so dass jede 2^m-Umgebung eines Elements in $\mathfrak{A}$ oder $\mathfrak{B}$ weniger als e Elemente hat, und

(b) für jeden 2^m-Umgebungstyp ρ ist $\#_\rho(\mathfrak{A}) = \#_\rho(\mathfrak{B})$ oder $\#_\rho(\mathfrak{A}),\, \#_\rho(\mathfrak{B}) > m \cdot e$,

dann ist $\mathfrak{A} \equiv_m \mathfrak{B}$.

Beachte: Falls $\mathfrak{A}$ und $\mathfrak{B}$ *endlich* sind, so kann man Bedingung (a) z.B. dadurch erfüllen, dass man $e := \max\{|A|, |B|\} + 1$ wählt.

Beweis: Seien $\mathfrak{A}$ und $\mathfrak{B}$ zwei σ-Strukturen, die die Voraussetzungen des Theorems erfüllen. Wir wollen zeigen, dass Duplicator das m-Runden EF-Spiel $\mathcal{G}_m(\mathfrak{A}, \mathfrak{B})$ gewinnt. Dazu zeigen wir, dass Duplicator so spielen kann, dass nach jeder Runde $i \in \{0, 1, \ldots, m\}$ die folgende Invariante $(*)_i$ erfüllt ist, wobei $a_1, \ldots, a_i$ und $b_1, \ldots, b_i$ die in den Runden $1, \ldots, i$ gewählten Elemente in A und B bezeichnen. Wir werden im Folgenden oft $(\mathfrak{A}, a_1, \ldots, a_i)$ und $(\mathfrak{B}, b_1, \ldots, b_i)$ als Strukturen der Signatur $\sigma_i := \sigma \cup \{\hat{c}_1, \ldots, \hat{c}_i\}$ auffassen, bei denen die Konstantensymbole $\hat{c}_1, \ldots, \hat{c}_i$ durch die Elemente $a_1, \ldots, a_i$ bzw. $b_1, \ldots, b_i$ belegt sind.

$(*)_i$: Für jeden 2^{m-i}-Umgebungstyp ρ gilt:

$$\#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i)\big) \;=\; \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i)\big) \qquad \text{oder}$$
$$\#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i)\big),\ \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i)\big) \;>\; (m{-}i) \cdot e.$$

Für $i = m$ bedeutet das dann insbesondere, dass für jeden 1-Umgebungstyp ρ gilt, dass $\#_\rho\big((\mathfrak{A}, a_1, \ldots, a_m)\big) = 0 \iff \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_m)\big) = 0$. Dies wiederum impliziert unmittelbar, dass die Abbildung

$$a_1, \ldots, a_m,\ (c^{\mathfrak{A}})_{c\in\sigma} \;\mapsto\; b_1, \ldots, b_m,\ (c^{\mathfrak{B}})_{c\in\sigma}$$

ein partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$ ist (Details: *Übung*) und dass somit Duplicator das Spiel gewinnt.

Es genügt also, im Folgenden per Induktion nach i zu zeigen, dass Duplicator so spielen kann, dass nach jeder Runde $i \in \{0, 1, \ldots, m\}$ die Invariante $(*)_i$ erfüllt ist.

$i = 0$: Die Bedingung $(*)_0$ ist nach Voraussetzung (b) von Theorem 3.25 erfüllt.

$i \to i{+}1$: Gemäß Induktionsannahme sind bereits i Runden gespielt und die Bedingung $(*)_i$ ist nach der i-ten Runde erfüllt. Wir müssen zeigen, dass Duplicator in Runde $i{+}1$ so spielen kann, dass danach die Bedingung $(*)_{i+1}$ erfüllt ist.

Wir betrachten hier nur den Fall, dass Spoiler in der $i{+}1$-ten Runde ein Element a_{i+1} in A wählt. (Der Fall, dass Spoiler ein Element b_{i+1} in B wählt kann aus Symmetriegründen analog, durch Vertauschen der Rollen von $\mathfrak{A}$ und $\mathfrak{B}$ behandelt werden.)

Sei $d := 2^{m-(i+1)}$; dann gilt also $2d = 2^{m-i}$. Sei

$$\kappa \;:=\; \mathrm{Typ}\big(2d, a_{i+1}, (\mathfrak{A}, a_1, \ldots, a_i)\big).$$

Insbesondere ist $\#_\kappa\big((\mathfrak{A}, a_1, \ldots, a_i)\big) \geqslant 1$. Gemäß $(*)_i$ existiert daher (mindestens) ein Element b_{i+1} in B mit

$$\mathrm{Typ}\big(2d, b_{i+1}, (\mathfrak{B}, b_1, \ldots, b_i)\big) \;\cong\; \kappa.$$

Wir zeigen im Folgenden: Wenn Duplicator dieses b_{i+1} wählt, dann gilt $(*)_{i+1}$.
Zum Nachweis von $(*)_{i+1}$ sei ρ ein beliebiger d-Umgebungstyp. Wir müssen zeigen, dass

$$\begin{aligned} &\#_\rho\big((\mathfrak{A}, a_1, .., a_i, a_{i+1})\big) = \#_\rho\big((\mathfrak{B}, b_1, .., b_i, b_{i+1})\big) \quad \text{oder} \\ &\#_\rho\big((\mathfrak{A}, a_1, .., a_i, a_{i+1})\big),\ \#_\rho\big((\mathfrak{B}, b_1, .., b_i, b_{i+1})\big) \ > \ (m-(i+1)) \cdot e. \end{aligned} \tag{3.2}$$

Sei dazu ρ' der d-Umgebungstyp, der aus ρ entsteht, indem $C(\rho)$ ersetzt wird durch $C(\rho) \setminus \{(\hat{c}_{i+1}, a_{i+1})\}$.

Zwischenbehauptung 1: Es gilt

$$\begin{aligned} & |\big\{a \in N_d^{\mathfrak{A}}(a_{i+1}) \ : \ \text{Typ}\big(d, a, (\mathfrak{A}, a_1, .., a_i)\big) \ \cong \ \rho'\big\}| \\ = \ & |\big\{b \in N_d^{\mathfrak{B}}(b_{i+1}) \ : \ \text{Typ}\big(d, b, (\mathfrak{B}, b_1, .., b_i)\big) \ \cong \ \rho'\big\}|. \end{aligned} \tag{3.3}$$

Beweis: Gemäß unserer Wahl von b_{i+1} wissen wir, dass

$$\text{Typ}\big(2d, a_{i+1}, (\mathfrak{A}, a_1, .., a_i)\big) \ \cong \ \text{Typ}\big(2d, b_{i+1}, (\mathfrak{B}, b_1, .., b_i)\big).$$

Da $N_d^{\mathfrak{A}}(a) \subseteq N_{2d}^{\mathfrak{A}}(a_{i+1})$ für jedes $a \in N_d^{\mathfrak{A}}(a_{i+1})$, und da $N_d^{\mathfrak{B}}(b) \subseteq N_{2d}^{\mathfrak{B}}(b_{i+1})$ für jedes $b \in N_d^{\mathfrak{A}}(b_{i+1})$ gilt, folgt daraus insbesondere die Aussage von Zwischenbehauptung 1 (Details: *Übung*). □

Um zu beweisen, dass (3.2) gilt, betrachten nun zwei Fälle.

Fall 1: $\rho \neq \rho'$, *d.h.* $(\hat{c}_{i+1}, a_{i+1}) \in C(\rho)$.

Dann gilt für alle $a \in A$ mit $\text{Typ}\big(d, a, (\mathfrak{A}, a_1, .., a_i, a_{i+1})\big) \cong \rho$, dass $a \in N_d^{\mathfrak{A}}(a_{i+1})$. Analog dazu gilt auch für alle $b \in B$ mit $\text{Typ}\big(d, b, (\mathfrak{B}, b_1, .., b_i, b_{i+1})\big) \cong \rho$, dass $b \in N_d^{\mathfrak{B}}(b_{i+1})$.
Somit gilt offensichtlich

$$\begin{aligned} \#_\rho\big((\mathfrak{A}, a_1, .., a_i, a_{i+1})\big) \ &= \ |\{a \in N_d^{\mathfrak{A}}(a_{i+1}) \ : \ \text{Typ}\big(d, a, (\mathfrak{A}, a_1, .., a_i, a_{i+1})\big) \cong \rho\}| \\ &= \ |\{a \in N_d^{\mathfrak{A}}(a_{i+1}) \ : \ \text{Typ}\big(d, a, (\mathfrak{A}, a_1, .., a_i)\big) \cong \rho'\}| \end{aligned}$$

und

$$\begin{aligned} \#_\rho\big((\mathfrak{B}, b_1, .., b_i, b_{i+1})\big) \ &= \ |\{b \in N_d^{\mathfrak{B}}(b_{i+1}) \ : \ \text{Typ}\big(d, b, (\mathfrak{B}, b_1, .., b_i, b_{i+1})\big) \cong \rho\}| \\ &= \ |\{b \in N_d^{\mathfrak{B}}(b_{i+1}) \ : \ \text{Typ}\big(d, b, (\mathfrak{B}, b_1, .., b_i)\big) \cong \rho'\}|. \end{aligned}$$

Zwischenbehauptung 1 liefert daher $\#_\rho\big((\mathfrak{A}, a_1, .., a_i, a_{i+1})\big) = \#_\rho\big((\mathfrak{B}, b_1, .., b_i, b_{i+1})\big)$.
Somit ist (3.2) in Fall 1 erfüllt.

Fall 2: $\rho = \rho'$, *d.h.* $(\hat{c}_{i+1}, a_{i+1}) \notin C(\rho)$.

Dann gilt für alle $a \in A$ mit $\text{Typ}\big(d, a, (\mathfrak{A}, a_1, .., a_i, a_{i+1})\big) \cong \rho$, dass $a \notin N_d^{\mathfrak{A}}(a_{i+1})$.

Analog dazu gilt auch für alle $b \in B$ mit $\text{Typ}\big(d, b, (\mathfrak{B}, b_1, \ldots, b_i, b_{i+1})\big) \cong \rho$, dass $b \notin N_d^{\mathfrak{B}}(b_{i+1})$. Daher gilt offensichtlich

$$\begin{aligned}&\#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i, a_{i+1})\big) \;=\\ &\quad \#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i)\big) \;-\; |\{a \in N_d^{\mathfrak{A}}(a_{i+1}) \;:\; \text{Typ}\big(d, a, (\mathfrak{A}, a_1, \ldots, a_i)\big) \cong \rho\}|\end{aligned}$$

und (3.4)

$$\begin{aligned}&\#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i, b_{i+1})\big) \;=\\ &\quad \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i)\big) \;-\; |\{b \in N_d^{\mathfrak{B}}(b_{i+1}) \;:\; \text{Typ}\big(d, b, (\mathfrak{B}, b_1, \ldots, b_i)\big) \cong \rho\}|.\end{aligned}$$

Wegen $d \leqslant 2d = 2^{m-i}$ lässt sich außerdem aus $(*)_i$ leicht folgern (Details: *Übung*), dass

$$\begin{aligned}&\text{(I):} \qquad \#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i)\big) = \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i)\big) \quad \text{oder}\\ &\text{(II):} \qquad \#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i)\big),\ \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i)\big) \;>\; (m{-}i) \cdot e.\end{aligned} \tag{3.5}$$

Falls (I) gilt, so folgt mit (3.4) und mit (3.3) (wegen $\rho'{=}\rho$), dass $\#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i, a_{i+1})\big) = \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i, b_{i+1})\big)$. Falls (II) gilt, so folgt mit (3.4) und mit der Tatsache, dass gemäß der Voraussetzung (a) von Theorem 3.25 $|N_d^{\mathfrak{A}}(a_{i+1})|,\ |N_d^{\mathfrak{B}}(b_{i+1})| < e$ (denn $d \leqslant 2^m$), dass

$$\#_\rho\big((\mathfrak{A}, a_1, \ldots, a_i, a_{i+1})\big),\ \#_\rho\big((\mathfrak{B}, b_1, \ldots, b_i, b_{i+1})\big) \;>\; (m{-}i) \cdot e \;-e \;=\; (m{-}(i{+}1)) \cdot e.$$

Somit ist (3.2) auch in Fall 2 erfüllt.
Insgesamt sind wir daher fertig mit dem Beweis von Theorem 3.25. □

Eine vereinfachte Variante des Beweises von Theorem 3.25 führt zu folgendem Resultat:

3.26 Korollar (vereinfachte Version des Satzes von Hanf)**.** *Sei σ eine Signatur, seien $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen und sei $m \in \mathbb{N}$. Falls $\#_\rho(\mathfrak{A}) = \#_\rho(\mathfrak{B})$ für jeden 2^m-Umgebungstyp ρ gilt, so ist $\mathfrak{A} \equiv_m \mathfrak{B}$.*

Beweis: Übung. □

Eine Anwendung des Satzes von Hanf: Die Hanf-Lokalität der Logik erster Stufe

Der Satz von Hanf liefert ein hinreichendes Kriterium, mit dem man leicht zeigen kann, dass zwei Strukturen m-äquivalent sind. Der Satz von Hanf besagt, dass alle FO-Sätze der Quantorentiefe m in dem Sinne "lokal" sind, dass sie nur über Umgebungen vom Radius 2^m "sprechen können". Im Folgenden wird diese "Lokalität" der Logik erster Stufe etwas genauer dargestellt.

3.27 Definition (Hanf-Lokalität)**.** Sei σ eine Signatur.

(a) Seien $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen und sei $r \in \mathbb{N}$. $\mathfrak{A}$ und $\mathfrak{B}$ heißen *r-bijektiv*, kurz:

$$\mathfrak{A} \leftrightarrows_r \mathfrak{B},$$

falls es eine Bijektion $f : A \to B$ gibt, so dass für alle $a \in A$ gilt:

$$\text{Typ}(r, a, \mathfrak{A}) \;\cong\; \text{Typ}(r, f(a), \mathfrak{B})$$

(b) Sei S eine Klasse von σ-Strukturen und $\mathsf{C} \subseteq \mathsf{S}$.
C heißt *Hanf-lokal* in S, falls es eine Zahl $r \in \mathbb{N}$ gibt, so dass für alle $\mathfrak{A}, \mathfrak{B} \in \mathsf{S}$ gilt:

$$\text{Falls } \;\mathfrak{A} \leftrightarrows_r \mathfrak{B}\,, \quad \text{so } \;\big(\mathfrak{A} \in \mathsf{C} \iff \mathfrak{B} \in \mathsf{C}\big)\,.$$

Die folgende einfache Folgerung aus dem Satz von Hanf besagt, dass jede FO-definierbare Klasse von Strukturen Hanf-lokal ist.

3.28 Theorem (Hanf-Lokalität von FO)**.** *Sei σ eine Signatur und S eine Klasse von σ-Strukturen. Dann gilt für jeden* FO$[\sigma]$*-Satz φ:* $\text{Mod}_{\mathsf{S}}(\varphi)$ *ist Hanf-lokal in* S*.*

Beweis: Sei φ ein FO$[\sigma]$-Satz und $\mathsf{C} := \text{Mod}_{\mathsf{S}}(\varphi)$. Sei $m := qr(\varphi)$. Wir wählen $r := 2^m$. Für Strukturen $\mathfrak{A}, \mathfrak{B} \in \mathsf{S}$ mit $\mathfrak{A} \leftrightarrows_r \mathfrak{B}$ müssen wir zeigen, dass $\mathfrak{A} \in \mathsf{C} \iff \mathfrak{B} \in \mathsf{C}$, d.h. wir müssen zeigen, dass $\mathfrak{A} \models \varphi \iff \mathfrak{B} \models \varphi$.

Wegen $\mathfrak{A} \leftrightarrows_r \mathfrak{B}$ und $r = 2^m$ gibt es eine Bijektion $f : A \to B$, so dass für alle $a \in A$ gilt:

$$\text{Typ}(2^m, a, \mathfrak{A}) \;\cong\; \text{Typ}(2^m, f(a), \mathfrak{B}).$$

Da f eine *Bijektion* ist, gilt also für jeden 2^m-Umgebungstyp ρ, dass $\#_\rho(\mathfrak{A}) = \#_\rho(\mathfrak{B})$. Aus Korollar 3.26 folgt daher, dass $\mathfrak{A} \equiv_m \mathfrak{B}$. D.h. $\mathfrak{A}$ und $\mathfrak{B}$ erfüllen dieselben FO$[\sigma]$-Sätze vom Quantorenrang m. Wegen $qr(\varphi) = m$ gilt insbesondere: $\mathfrak{A} \models \varphi \iff \mathfrak{B} \models \varphi$. $\square$

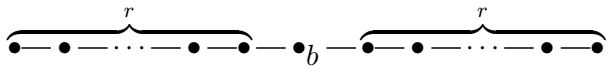
3.29 Bemerkung. Indem man zeigt, dass eine Klasse C *nicht Hanf-lokal* in S ist, kann man (unter Verwendung von Theorem 3.28) zeigen, dass C *nicht* FO*-definierbar* in S ist.
Dass C nicht Hanf-lokal in S ist, kann man dadurch zeigen, dass man für jede Zahl $r \in \mathbb{N}$ eine Struktur $\mathfrak{A} \in \mathsf{C}$ und eine Struktur $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$ mit $\mathfrak{A} \leftrightarrows_r \mathfrak{B}$ findet.

3.30 Beispiel. Die Verwendung der Hanf-Lokalität von FO liefert einen alternativen Beweis von Satz 3.20 (b): Conn ist nicht FO-definierbar in UGraphs.[5]

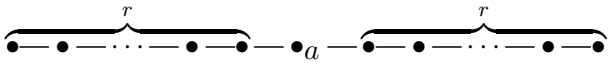
Beweis: Gemäß Theorem 3.28 reicht es zu zeigen, dass Conn nicht Hanf-Lokal in UGraphs ist. Wir müssen also für jede Zahl $r \in \mathbb{N}$ einen zusammenhängenden Graphen $\mathfrak{A}$ und einen nicht-zusammenhängenden Graphen $\mathfrak{B}$ finden, so dass $\mathfrak{A} \leftrightarrows_r \mathfrak{B}$.
Sei $r \in \mathbb{N}$ beliebig. Als $\mathfrak{B}$ wählen wir einen Graph, der aus zwei disjunkten Kreisen auf je

[5] Zur Erinnerung: UGraphs ist die Klasse aller endlichen ungerichteten Graphen, Conn ist die Klasse aller zusammenhängenden endlichen ungerichteten Graphen.

$\ell{+}1$ Knoten besteht, wobei $\ell > 2r$ ist. Wegen $\ell > 2r$ sieht jede r-Umgebung eines Knotens b von B folgendermaßen aus:



Als Struktur $\mathfrak{A}$ wählen wir einen Kreis, der genausoviele Knoten wie $\mathfrak{B}$ hat, d.h. $\mathfrak{A}$ ist ein Kreis auf $2\ell{+}2$ Knoten. Jede r-Umgebung eines Knotens a von A sieht folgendermaßen aus:



Sie ist also isomorph zu jeder r-Umgebung eines Knotens b von B. D.h.: für alle $a \in A$ und alle $b \in B$ ist $\text{Typ}(r, a, \mathfrak{A}) \cong \text{Typ}(r, b, \mathfrak{B})$. Wir können also *irgendeine* Bijektion f von A nach B wählen (und die gibt es, da $|A| = |B|$) und haben damit gezeigt, dass $\mathfrak{A} \leftrightarrows_r \mathfrak{B}$. Wegen $\mathfrak{A} \in \mathsf{Conn}$ und $\mathfrak{B} \in \mathsf{UGraphs} \setminus \mathsf{Conn}$ haben wir also gezeigt, dass Conn nicht Hanf-lokal in $\mathsf{UGraphs}$ ist. □

3.1.5 Der Satz von Fraïssé

Die Charakterisierung der m-Äquivalenz $\equiv_m$ durch Ehrenfeucht-Fraïssé Spiele ist eine gute Sichtweise, um Beweis*ideen zu finden*, indem man nach einer Gewinnstrategie für Duplicator im m-Runden EF-Spiel sucht. Um Nichtausdrückbarkeits-Beweise *exakt aufschreiben* zu können, ist die im Folgenden vorgestellte Charakterisierung von Fraïssé oft sehr nützlich.

3.31 Definition (Gewinnpositionen $W_m(\mathfrak{A}, \mathfrak{B})$)**.**
Sei σ eine Signatur, $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen und $m \in \mathbb{N}$. Die Menge $W_m(\mathfrak{A}, \mathfrak{B})$ aller *Gewinnpositionen für Duplicator* besteht aus allen Abbildungen

$$p \;:\; \vec{a}', (c^{\mathfrak{A}})_{c\in\sigma} \;\mapsto\; \vec{b}', (c^{\mathfrak{B}})_{c\in\sigma}\,,$$

für die $\vec{a}' = a'_1, \ldots, a'_k \in A$, $\vec{b}' = b'_1, \ldots, b'_k \in B$, $k \in \mathbb{N}$, so dass Duplicator das Spiel $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$ gewinnt.

3.32 Definition (Hin-und-Her-Systeme und m-Isomorphie)**.** Sei σ eine Signatur und sei $m \in \mathbb{N}$. Zwei σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ heißen *m-isomorph* (kurz: $\mathfrak{A} \cong_m \mathfrak{B}$), falls es eine Folge $(I_j)_{j=0,\ldots,m}$ mit den folgenden drei Eigenschaften gibt:

(1) Für jedes $j \in \{0, \ldots, m\}$ ist $\varnothing \neq I_j \subseteq \text{Part}(\mathfrak{A}, \mathfrak{B})$ (d.h. I_j ist eine nicht-leere Menge partieller Isomorphismen von $\mathfrak{A}$ nach $\mathfrak{B}$).

(2) *"Hin-Eigenschaft":* Für jedes $j < m$, jedes $p \in I_{j+1}$ und jedes $a \in A$ gibt es ein $q \in I_j$, so dass $q \supseteq p$ und $a \in \text{def}(q)$ (d.h. es gibt eine Erweiterung q von p, in deren Definitionsbereich a liegt).

(3) *"Her-Eigenschaft":* Für jedes $j < m$, jedes $p \in I_{j+1}$ und jedes $b \in B$ gibt es ein $q \in I_j$, so dass $q \supseteq p$ und $b \in \text{bild}(q)$ (d.h. es gibt eine Erweiterung q von p, in deren Bildbereich b liegt).

Falls $(I_j)_{j\leqslant m}$ die Eigenschaften (1), (2) und (3) hat, so nennen wir $(I_j)_{j\leqslant m}$ ein *Hin-und-Her-System* (der Ordnung m), schreiben $(I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$ und sagen $\mathfrak{A}$ *und* $\mathfrak{B}$ *sind* m*-isomorph vermöge* $(I_j)_{j\leqslant m}$.

Anschaulich bedeuten die Bedingungen (2) und (3) folgendes: In I_{j+1} liegen nur solche partiellen Isomorphismen p, die sich $j+1$-mal erweitern lassen. Die Erweiterungen $p_j, p_{j-1}, \ldots, p_0$, die man dabei nacheinander erhält, sind allesamt partielle Isomorphismen, die in den Mengen $I_j, I_{j-1}, \ldots, I_0$ liegen.

Das folgende Theorem besagt, dass zwei Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ genau dann m-isomorph sind, wenn Duplicator das m-Runden EF-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$ gewinnt.

3.33 Theorem. *Sei σ eine Signatur. $\mathfrak{A}$ und $\mathfrak{B}$ seien σ-Strukturen, $k, m \in \mathbb{N}$ und $\vec{a}' = a'_1, \ldots, a'_k \in A$ und $\vec{b}' = b'_1, \ldots, b'_k \in B$. Dann sind äquivalent:*

(a) Duplicator gewinnt $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$.

(b) $\big(W_j(\mathfrak{A}, \mathfrak{B})\big)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$ und $\big(\vec{a}', (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}', (c^{\mathfrak{B}})_{c\in\sigma}\big) \in W_m(\mathfrak{A}, \mathfrak{B})$.

(c) Es gibt $(I_j)_{j\leqslant m}$, so dass $(I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$ und $\big(\vec{a}', (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}', (c^{\mathfrak{B}})_{c\in\sigma}\big) \in I_m$.

Beweis: "(a) $\Longrightarrow$ (b)": Gilt gemäß der Definition der Gewinnpositionen $W_j(\mathfrak{A}, \mathfrak{B})$.
"(b) $\Longrightarrow$ (c)": Gilt mit $(I_j)_{j\leqslant m} := \big(W_j(\mathfrak{A}, \mathfrak{B})\big)_{j\leqslant m}$.
"(c) $\Longrightarrow$ (a)": Gemäß Voraussetzung gibt es $(I_j)_{j\leqslant m}$, so dass $(I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$ und $\Big(\vec{a}', (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}', (c^{\mathfrak{B}})_{c\in\sigma}\Big) \in I_m$. Per Induktion nach i zeigen wir, dass Duplicator $(I_j)_{j\leqslant m}$ nutzen kann, um das Spiel $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$ so zu spielen, dass für jedes $i \in \{0, \ldots, m\}$ gilt:

$(*)_i$: Sind $a_1, \ldots, a_i$ bzw. $b_1, \ldots, b_i$ die in den Runden $1, \ldots, i$ in A bzw. B gewählten Elemente, so gibt es einen partiellen Isomorphismus $p \in I_{m-i}$, so dass $a'_1, \ldots, a'_k$, $a_1, \ldots, a_i \in \text{def}(p)$ und

$$p(a'_j) = b'_j \quad \text{für alle } j \in \{1, \ldots, k\} \quad \text{und}$$
$$p(a_j) = b_j \quad \text{für alle } j \in \{1, \ldots, i\}.$$

<u>$i = 0$:</u> $(*)_0$ gilt, da $\Big(\vec{a}', (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}', (c^{\mathfrak{B}})_{c\in\sigma}\Big) \in I_m$.
<u>$i \mapsto i+1$:</u> Sei p der partielle Isomorphismus aus I_{m-i}, der gemäß der Induktionsannahme $(*)_i$ existiert.
Wir betrachten zunächst den Fall, dass Spoiler in Runde $i+1$ ein Element $a_{i+1} \in A$ wählt.

Gemäß der "Hin-Eigenschaft" gibt es eine Erweiterung $q \supseteq p$ in $I_{(m-i)-1}$, in deren Definitionsbereich a_{i+1} liegt. Duplicator kann in Runde $i{+}1$ daher mit $b_{i+1} := q(a_{i+1})$ antworten und hat damit die Bedingung $(*)_{i+1}$ erfüllt.
In dem Fall, dass Spoiler in Runde $i{+}1$ ein Element $b_{i+1} \in B$ wählt, kann Duplicator eine Erweiterung $q \supseteq p$ in $I_{(m-i)-1}$ finden, in deren Bildbereich b_{i+1} liegt. Er kann daher mit einem a_{i+1} antworten, für das $q(a_{i+1}) = b_{i+1}$ gilt, und hat damit $(*)_{i+1}$ erfüllt. □

Als direkte Folgerung aus Theorem 3.16 und Theorem 3.33 ergibt sich:

3.34 Korollar. *Sei $m \in \mathbb{N}$, σ eine Signatur und $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen. Dann sind äquivalent:*

(a) Duplicator gewinnt $\mathcal{G}_m(\mathfrak{A}, \mathfrak{B})$.

(b) $\mathfrak{A} \equiv_m \mathfrak{B}$.

(c) $\mathfrak{A} \cong_m \mathfrak{B}$.

(d) $\mathfrak{B} \models \varphi^m_{\mathfrak{A}}$.

Die Äquivalenz von (b) und (c) ist als *Satz von Fraïssé* (1954) bekannt.

3.35 Beispiel. Für jedes $\ell \in \mathbb{N}_{\geqslant 1}$ sei G_ℓ ein ungerichteter Kreis der Länge ℓ, d.h. G_ℓ hat Knotenmenge $\{1, \ldots, \ell\}$ und Kantenmenge

$$E^{G_\ell} := \{(i, i{+}1) : i < \ell\} \cup \{(\ell, 1)\} \cup \{(i{+}1, i) : i < \ell\} \cup \{(1, \ell)\}.$$

Für $\ell, k \in \mathbb{N}$ sei $G_{\ell,k}$ die disjunkte Vereinigung von G_ℓ und G_k, d.h. $G_{\ell,k}$ besteht aus zwei ungerichteten Kreisen der Längen ℓ und k.
Wir zeigen, dass für alle $m \in \mathbb{N}$ gilt:

$$\text{Sind } \ell, k > 2^m, \text{ so gilt } G_\ell \cong_m G_{\ell,k}, \text{ d.h.}$$

die Graphen G_ℓ und $G_{\ell,k}$ lassen sich nicht durch FO-Sätze der Quantorentiefe $\leqslant m$ unterscheiden lassen.

Beweis: Für einen ungerichteten Graphen G sei $\mathrm{Dist}^G(\cdot,\cdot)$ die in Definition 3.22 eingeführte Distanzfunktion. Für jedes $j \in \{0, \ldots, m\}$ sei I_j die Menge aller partiellen Isomorphismen p von G_ℓ nach $G_{\ell,k}$, für die gilt:

- $|\mathrm{def}(p)| \leqslant m{-}j$, und
- für alle $a, a' \in \mathrm{def}(p)$ gilt:
  $\mathrm{Dist}^{G_\ell}(a, a') = \mathrm{Dist}^{G_{\ell,k}}(p(a), p(a'))$ oder $\mathrm{Dist}^{G_\ell}(a, a'),\ \mathrm{Dist}^{G_{\ell,k}}(p(a), p(a')) \geqslant 2^{j+1}$.

I_m besteht gerade aus der Abbildung "$\varnothing$", deren Definitionsbereich leer ist. Per Induktion kann man (ähnlich wie im Beweis von Satz 3.12) nachweisen, dass I_j für jedes $j \in \{m, m{-}1, \ldots, 0\}$ die Hin- und die Her-Eigenschaft hat und dass $I_j \neq \varnothing$.
Insgesamt haben wir damit gezeigt, dass $(I_j)_{j \leqslant m} : G_\ell \cong_m G_{\ell,k}$. □

3.2 EF-Spiele für Fragmente der Logik zweiter Stufe

3.2.1 Das EF-Spiel für die existentielle Logik zweiter Stufe

Gemäß Definition 2.6 haben ESO-Formeln $\psi(\vec{x})$ die Gestalt

$$\exists X_1 \cdots \exists X_l \, \varphi(\vec{x}, X_1, .., X_l),$$

wobei $X_1, .., X_l$ Relationsvariablen sind und $\varphi \in \text{FO}[\sigma \dot\cup \{X_1, \ldots, X_l\}]$. Das Ehrenfeucht-Fraïssé-Spiel wird nun entsprechend dem Formelaufbau auf naheliegende Weise für die Logik ESO erweitert.

3.36 Definition (EF-Spiel für ESO)**.** Seien $l, m \in \mathbb{N}_{\geqslant 1}$ und seien $s_1, .., s_l \in \mathbb{N}_{\geqslant 1}$. Sei σ eine Signatur und $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen. Das $((s_1, .., s_l), m)$*-EF-Spiel auf* $\mathfrak{A}$ *und* $\mathfrak{B}$ ist wie folgt definiert.

Phase 1: Spoiler wählt Relationen $S_1 \subseteq A^{s_1}, \ldots, S_l \subseteq A^{s_l}$ über dem Universum von $\mathfrak{A}$. Duplicator antwortet mit Relationen $S'_1 \subseteq B^{s_1}, \ldots, S'_l \subseteq B^{s_l}$ über dem Universum von $\mathfrak{B}$.

Phase 2: Spoiler und Duplicator spielen das m-Runden EF-Spiel auf den Strukturen $(\mathfrak{A}, S_1, \ldots, S_l)$ und $(\mathfrak{B}, S'_1, .., S'_l)$, d.h. sie spielen das in Abschnitt 3.1.2 eingeführte Spiel $\mathcal{G}_m\big((\mathfrak{A}, S_1, .., S_l), (\mathfrak{B}, S'_1, .., S'_l)\big)$.

Dieses Spiel charakterisiert die Definierbarkeit durch einen ESO-Satz folgendermaßen:

3.37 Satz. *Seien $l, m \in \mathbb{N}_{\geqslant 1}$ und seien $s_1, .., s_l \in \mathbb{N}_{\geqslant 1}$. Sei σ eine Signatur und $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen. Dann sind äquivalent:*

(a) Spoiler hat eine Gewinnstrategie im $((s_1, .., s_l), m)$-EF-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$.

(b) Es gibt einen ESO$[\sigma]$*-Satz $\psi := \exists X_1 \cdots \exists X_l \, \varphi$ mit $\varphi \in$ FO, $qr(\varphi) \leqslant m$ und $ar(X_i) = s_i$ (für alle $i \in \{1, .., l\}$), so dass $\mathfrak{A} \models \psi$ und $\mathfrak{B} \not\models \psi$.*

Beweis: "(b) $\Longrightarrow$ (a)": Sei $\psi := \exists X_1 \cdots \exists X_l \, \varphi$ ein ESO-Satz mit $\varphi \in$ FO, $qr(\varphi) \leqslant m$ und $ar(X_i) = s_i$ für $1 \leqslant i \leqslant l$, so dass $\mathfrak{A} \models \psi$ und $\mathfrak{B} \not\models \psi$. Spoiler hat folgende Gewinnstrategie für das $((s_1, .., s_l), m)$-Runden Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$. Da $\mathfrak{A} \models \psi$, existieren Relationen $S_1 \subseteq A^{s_1}, \ldots, S_l \subseteq A^{s_l}$, so dass $(\mathfrak{A}, S_1, .., S_l) \models \varphi$. In Phase 1 des Spiels wählt Spoiler diese Relationen. Seien $S'_1 \subseteq B^{s_1}, \ldots, S'_l \subseteq B^{s_l}$ die Relationen, mit denen Duplicator in Phase 1 antwortet. Da $\mathfrak{B} \not\models \psi$, gilt $(\mathfrak{B}, S'_1, .., S'_l) \not\models \varphi$. Wegen $qr(\varphi) \leqslant m$ gilt also $(\mathfrak{A}, S_1, .., S_l) \not\equiv_m (\mathfrak{B}, S'_1, .., S'_l)$. Gemäß des Sates von Ehrenfeucht (Theorem 3.16) hat Spoiler somit in Phase 2 eine Gewinnstrategie für das Spiel $\mathcal{G}_m\big((\mathfrak{A}, S_1, .., S_l), (\mathfrak{B}, S'_1, .., S'_l)\big)$.

"(a) $\Longrightarrow$ (b)": Nach Voraussetzung hat Spoiler eine Gewinnstrategie im $((s_1, .., s_l), m)$-EF-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$. D.h. es gibt $S_1 \subseteq A^{s_1}, \ldots, S_l \subseteq A^{s_l}$, so dass für alle $S'_1 \subseteq$

$B^{s_1}, \ldots, S'_l \subseteq B^{s_l}$ gilt: $(\mathfrak{A}, S_1, .., S_l) \not\equiv_m (\mathfrak{B}, S'_1, \ldots, S'_l)$. Insbesondere gibt es also für jede Wahl von $S'_1, .., S'_l$ einen FO$[\sigma \dot\cup \{X_1, .., X_l\}]$-Satz $\varphi_{S'_1,..,S'_l}$ vom Quantorenrang $\leqslant m$, so dass $(\mathfrak{A}, S_1, .., S_l) \models \varphi_{S'_1,..,S'_l}$ aber $(\mathfrak{B}, S'_1, .., S'_l) \not\models \varphi_{S'_1,..,S'_l}$. Mit[6]

$$\varphi \;:=\; \bigwedge \left\{ \varphi_{S'_1,..,S'_l} \;:\; S'_1 \subseteq B^{s_1},\; \ldots,\; S'_l \subseteq B^{s_l} \right\}$$

erhält man eine FO-Formel mit $qr(\varphi) \leqslant m$, $(\mathfrak{A}, S_1, .., S_l) \models \varphi$ aber für kein $S'_1 \subseteq B^{s_1}, \ldots, S'_l \subseteq B^{s_l}$ gilt $(\mathfrak{B}, S'_1, .., S'_l) \models \varphi$. Daraus folgt sofort $\mathfrak{A} \models \exists X_1 \cdots X_l\, \varphi$ und $\mathfrak{B} \not\models \exists X_1 \cdots \exists X_l\, \varphi$. □

Die soeben eingeführten Spiele liefern einen prinzipiellen Ansatz um co-NP und NP zu trennen, und damit auch PTIME und NP: Aus dem Satz von Fagin folgt, dass ein Problem genau dann in NP liegt, wenn es ESO-definierbar ist. Nehmen wir nun ein co-NP-vollständiges Problem C (z.B. das Problem *"Nicht-3-Färbbarkeit"*), so ist $\mathsf{C} \in$ NP genau dann, wenn co-NP $=$ NP. Es gilt also

$$\begin{aligned}
& \text{co-NP} \neq \text{NP} \\
\iff\; & \mathsf{C} \notin \text{NP} \\
\iff\; & \mathsf{C} \text{ ist nicht ESO-definierbar in } \mathsf{Fin} \\
\iff\; & \text{für alle } l, m \in \mathbb{N}_{\geqslant 1} \text{ und alle } s_1, \ldots, s_l \in \mathbb{N}_{\geqslant 1} \text{ gibt es } \mathfrak{A} \in \mathsf{C} \\
& \text{und } \mathfrak{B} \in \mathsf{Fin} \setminus \mathsf{C}, \text{ so dass Duplicator eine Gewinnstrategie im} \\
& ((s_1, \ldots, s_l), m)\text{-EF-Spiel auf } \mathfrak{A} \text{ und } \mathfrak{B} \text{ hat.}
\end{aligned}$$

Leider ist bis heute dieser Ansatz, genau wie alle anderen Verfahren zum Trennen von co-NP und NP, gescheitert — unter anderem an der enormen Komplexität des Nachweises von Gewinnstrategien.

Das in Definition 3.36 eingeführte EF-Spiel für ESO kann selbstverständlich auf naheliegende Weise für die gesamte Logik zweiter Stufe erweitert werden. Das Finden von Gewinnstrategien wird dabei natürlich nicht unbedingt einfacher.

3.2.2 Das Ajtai-Fagin-Spiel für monadische existentielle Logik zweiter Stufe

Interessiert man sich für Anwendungen der existentiellen Logik zweiter Stufe in der Komplexitätstheorie, wie sie etwa am Ende des vorherigen Abschnitts angesprochen wurden, so nimmt das Fragment von ESO, bei dem nur über *Mengen* (also 1-stellige Relationen) quantifiziert werden darf, eine besondere Rolle ein. Zum einen können viele NP-vollständige Probleme, z.B. 3-Färbbarkeit oder das aussagenlogische Erfüllbarkeitsproblem (bei geeigneter Repräsentation von aussagenlogischen Formeln durch endliche Strukturen) durch ESO-Formeln beschrieben werden, die nur über Mengen quantifizieren. Andererseits wird

[6] Beachte: φ ist eine FO-Formel, da es nur *endlich* viele nicht-äquivalente FO-Formeln der Quantorentiefe m gibt, siehe Bemerkung 3.18.

die Analyse von Formeln natürlich tendenziell eher leichter, wenn man über keine Relationen höherer Stelligkeit quantifizieren darf. In diesem Abschnitt werden daher spezielle Spiele für diese Klasse von Formeln eingeführt.

3.38 Definition (*Mon*-Σ^1_1 und *Mon*-Π^1_1)**.**
Die *monadische existenzielle Logik zweiter Stufe* (kurz: *Mon*-Σ^1_1; in der Literatur wird diese Klasse oft auch *monadic* ESO, *monadic* Σ^1_1 oder *monadic* NP genannt) besteht aus allen SO-Formeln, die von der Form $\exists X_1 \cdots \exists X_l \, \varphi$ sind, wobei $l \geqslant 0$ ist, $X_1, \ldots, X_l$ Relationsvariablen der Stelligkeit 1 sind (so genannte *Mengenvariablen* oder auch *monadische* bzw. *unäre* Relationsvariablen) und φ eine FO-Formel ist.

Analog dazu besteht die *monadische universelle Logik zweiter Stufe* (kurz: *Mon*-Π^1_1) aus allen SO-Formeln der Form $\forall X_1 \cdots \forall X_l \, \varphi$, wobei $l \geqslant 0$ ist, $X_1, \ldots, X_l$ Mengenvariablen sind und φ eine FO-Formel ist.

3.39 Bemerkung. Man sieht leicht, dass die Ausdrucksstärke der monadischen existentiellen Logik zweiter Stufe echt größer ist als die der Logik erster Stufe (kurz: FO $<$ *Mon*-Σ^1_1 auf Fin). Beispielsweise lässt sich leicht ein *Mon*-Σ^1_1-Satz ψ der Form $\exists X \, \varphi$ finden, der von genau denjenigen endlichen linearen Ordnungen erfüllt wird, deren Universum aus einer *geraden* Anzahl von Elementen besteht (Details: *Übung*). D.h.: $\mathsf{Even}_<$ ist *Mon*-Σ^1_1-definierbar in $\mathsf{S}_<$. Von Satz 3.20 wissen wir schon, dass $\mathsf{Even}_<$ *nicht* FO-definierbar in $\mathsf{S}_<$ ist.

Das folgende Spiel wurde im Jahr 1988 von Ajtai und Fagin eingeführt um die *Mon*-Σ^1_1-Definierbarkeit zu charakterisieren.

3.40 Definition (Ajtai-Fagin-Spiel)**.** Sei σ eine Signatur, S eine Klasse von σ-Strukturen und sei $\mathsf{C} \subseteq \mathsf{S}$. Seien $l, m \in \mathbb{N}_{\geqslant 1}$. Das (l,m)-*Ajtai-Fagin-Spiel für* C *auf* S wird wie folgt gespielt.

Phase 1: Duplicator wählt eine Struktur $\mathfrak{A} \in \mathsf{C}$.
Danach wählt Spoiler l Mengen $S_1 \subseteq A, \ \ldots, \ S_l \subseteq A$.

Phase 2: Duplicator wählt eine Struktur $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$ und l Mengen $S'_1 \subseteq B, \ldots, S'_l \subseteq B$.

Phase 3: Spoiler und Duplicator spielen das m-Runden EF-Spiel auf $(\mathfrak{A}, S_1, \ldots, S_l)$ und $(\mathfrak{B}, S'_1, \ldots, S'_l)$, d.h. sie spielen das Spiel $\mathcal{G}_m\big((\mathfrak{A}, S_1, \ldots, S_l), (\mathfrak{B}, S'_1, \ldots, S'_l)\big)$.

3.41 Satz. *Sei σ eine Signatur, sei S eine Klasse von σ-Strukturen und sei $\mathsf{C} \subseteq \mathsf{S}$. Dann sind äquivalent:*

(a) C *ist Mon-*Σ^1_1*-definierbar in* S*.*

(b) Es gibt $l, m \in \mathbb{N}_{\geqslant 1}$, so dass Spoiler eine Gewinnstrategie im (l,m)-Ajtai-Fagin-Spiel für C *auf* S *hat.*

Beweis: "(a) $\Longrightarrow$ (b)": Sei $\psi := \exists X_1 \cdots \exists X_l \, \varphi$ ein *Mon*-Σ^1_1-Satz so dass $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\psi)$, $l \geqslant 0$ und $\varphi \in$ FO. Sei $m := qr(\varphi)$. Spoiler hat folgende Gewinnstrategie im (l,m)-Ajtai-Fagin-Spiel für C auf S:

Sei $\mathfrak{A} \in \mathsf{C}$ die von Duplicator in Phase 1 gewählte Struktur. Wegen $\mathfrak{A} \in \mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\psi)$ gibt es Mengen $S_1 \subseteq A, \ldots, S_l \subseteq A$ so dass $(\mathfrak{A}, S_1, \ldots, S_l) \models \varphi$. In Phase 1 wählt Spoiler diese Mengen $S_1, \ldots, S_l$.

Sei $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$ und $S'_1 \subseteq B, \ldots, S'_l \subseteq B$ die Wahl von Duplicator in Phase 2. Wegen $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$ und $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\psi)$ wissen wir, dass $\mathfrak{B} \not\models \exists X_1 \cdots \exists X_l \, \varphi$, also $(\mathfrak{B}, S'_1, \ldots, S'_l) \not\models \varphi$.

Wegen $qr(\varphi) = m$ folgt damit: $(\mathfrak{A}, S_1, \ldots, S_l) \not\equiv_m (\mathfrak{B}, S'_1, \ldots, S'_l)$. Somit hat Spoiler eine Gewinnstrategie im m-Runden EF-Spiel auf $(\mathfrak{A}, S_1, \ldots, S_l)$ und $(\mathfrak{B}, S'_1, \ldots, S'_l)$. Insgesamt haben wir also eine Gewinnstrategie für Spoiler im (l,m)-Ajtai-Fagin-Spiel für C auf S konstruiert.

"(b) $\Longrightarrow$ (a)": Nach Voraussetzung gibt es $l, m \geqslant 1$ so dass Spoiler eine Gewinnstrategie im (l,m)-Ajtai-Fagin-Spiel für C auf S hat. Das heißt, für jedes $\mathfrak{A} \in \mathsf{C}$ gibt es Mengen $\vec{S} = S_1, \ldots, S_l \subseteq A$ so dass für alle $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$ und alle $\vec{S'} = S'_1, \ldots, S'_l \subseteq B$ gilt:

$$(\mathfrak{A}, S_1, \ldots, S_l) \not\equiv_m (\mathfrak{B}, S'_1, \ldots, S'_l).$$

Insbesondere gibt es also für jede Wahl von $\vec{S'} = S'_1, \ldots, S'_l \subseteq B$ einen FO$[\sigma \cup \{X_1, \ldots, X_l\}]$-Satz $\varphi_{\mathfrak{A};\mathfrak{B},\vec{S'}}$ vom Quantorenrang $\leqslant m$, so dass

$$(\mathfrak{A}, \vec{S}) \models \varphi_{\mathfrak{A};\mathfrak{B},\vec{S'}} \qquad \text{und} \qquad (\mathfrak{B}, \vec{S'}) \not\models \varphi_{\mathfrak{A};\mathfrak{B},\vec{S'}}\,.$$

Setze[7]

$$\begin{aligned} \varphi_{\mathfrak{A}} &:= \bigwedge \{\, \varphi_{\mathfrak{A};\mathfrak{B},\vec{S'}} \;:\; \mathfrak{B} \in \mathsf{S} \setminus \mathsf{C},\ \vec{S'} = S'_1, \ldots, S'_l \subseteq B \,\}, \\ \varphi &:= \bigvee \{\, \varphi_{\mathfrak{A}} \;:\; \mathfrak{A} \in \mathsf{C} \,\}, \qquad \text{und} \\ \psi &:= \exists X_1 \cdots \exists X_l \, \varphi. \end{aligned}$$

Diese Formel ψ ist ein *Mon*-Σ^1_1-Satz, so dass für alle $\mathfrak{A} \in \mathsf{C}$ gilt: $\mathfrak{A} \models \psi$, und für alle $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$ gilt: $\mathfrak{B} \not\models \psi$. □

In Satz 3.20 haben wir bereits gesehen dass *Graph-Zusammenhang* nicht in der Logik FO definiert werden kann. Unter Verwendung des Ajtai-Fagin-Spiels können wir nun das analoge Resultat auch für die (stärkere) Logik *Mon*-Σ^1_1 zeigen:

3.42 Theorem (Fagin, 1975)**.** Conn *ist nicht Mon-*Σ^1_1*-definierbar in* UGraphs.[8]

Beweis: Gemäß Satz 3.41 genügt es, für jede Wahl von $l, m \in \mathbb{N}_{\geqslant 1}$ zu zeigen, dass Duplicator eine Gewinnstrategie im (l,m)-Ajtai-Fagin-Spiel für Conn auf UGraphs hat. Seien

[7] Beachte: $\varphi_{\mathfrak{A}}$ und φ sind FO-Formeln, da es nur *endlich* viele nicht-äquivalente FO-Formeln der Quantorentiefe m gibt, siehe Bemerkung 3.18.

[8] Zur Erinnerung: UGraphs ist die Klasse aller endlichen ungerichteten Graphen, und Conn ist die Klasse aller zusammenhängenden endlichen ungerichteten Graphen.

also $l, m \in \mathbb{N}_{\geqslant 1}$. Duplicator spielt das (l, m)-Ajtai-Fagin-Spiel für Conn auf UGraphs gemäß folgender Strategie:

Phase 1: Duplicator wählt als $\mathfrak{A} = (A, E^{\mathfrak{A}}) \in$ Conn einen ungerichteten Kreis auf n Knoten, für n hinreichend groß. "Hinreichend groß" heißt hier, dass n so groß sein soll, dass nach Spoilers Wahl der Mengen $\vec{S} = S_1, \ldots, S_l \subseteq A$ für die resultierende Struktur $\hat{\mathfrak{A}} := (\mathfrak{A}, \vec{S})$ gilt: Es gibt zwei Knoten v und w in A, deren gerichtete 2^m-Umgebungen disjunkt und isomorph sind, d.h. $\text{Dist}^{\hat{\mathfrak{A}}}(v, w) \geqslant 2 \cdot 2^m + 1$ und $\text{Typ}\big(2^m, v, \hat{\mathfrak{A}}_{\circlearrowright}\big) \cong \text{Typ}\big(2^m, w, \hat{\mathfrak{A}}_{\circlearrowright}\big)$, wobei $\hat{\mathfrak{A}}_{\circlearrowright}$ die *gerichtete* Variante von $\hat{\mathfrak{A}}$ bezeichnet, in der die Kanten im Uhrzeigersinn ausgerichtet sind. Zur genauen Wahl von n schauen wir uns die möglichen 2^m-Umgebungstypen in $\hat{\mathfrak{A}}_{\circlearrowright}$ an. Ist $|A| = n > 2 \cdot 2^m + 1$, so gilt für jeden Knoten v in $\hat{\mathfrak{A}}_{\circlearrowright}$: Die 2^m-Umgebung von v ist ein gerichteter Pfad auf $2 \cdot 2^m + 1$ Knoten, und v ist der Knoten in der Mitte dieses Pfads. Durch die unären Relationen $\vec{S} = S_1, \ldots, S_l$ ist außerdem jeder dieser $2 \cdot 2^m + 1$ Knoten mit einer von 2^l möglichen "Farben" markiert (nämlich der Teilmenge von $\{1, \ldots, l\}$, die aus allen j besteht, so dass der Knoten zur Menge S_j gehört). Somit gibt es höchstens $I := (2^l)^{2 \cdot 2^m + 1}$ mögliche 2^m-Umgebungstypen. Wir setzen nun

$$n \;:=\; (I{+}1) \cdot (2 \cdot 2^m + 1)$$

und lassen Duplicator in Phase 1 einen ungerichteten Kreis $\mathfrak{A}$ auf n Knoten wählen. Seien $\vec{S} := S_1, \ldots, S_l \subseteq A$ die von Spoiler in Phase 1 gewählten Mengen, sei $\hat{\mathfrak{A}} := (\mathfrak{A}, \vec{S})$, und sei $\hat{\mathfrak{A}}_{\circlearrowright}$ die *gerichtete* Variante von $\hat{\mathfrak{A}}$, in der die Kanten im Uhrzeigersinn ausgerichtet sind.

Phase 2: Gemäß unserer Wahl von n wissen wir, dass es zwei Knoten $v, w \in A$ geben muss, so dass $\text{Dist}^{\hat{\mathfrak{A}}}(v, w) \geqslant 2 \cdot 2^m + 1$ und $\text{Typ}\big(2^m, v, \hat{\mathfrak{A}}_{\circlearrowright}\big) \cong \text{Typ}\big(2^m, w, \hat{\mathfrak{A}}_{\circlearrowright}\big)$.

Seien v^-, v^+, w^-, w^+ diejenigen vier Knoten in A, für die gilt

$$(v^-, v) \in E^{\mathfrak{A}}, \quad (v, v^+) \in E^{\mathfrak{A}}, \quad (w^-, w) \in E^{\mathfrak{A}}, \quad (w, w^+) \in E^{\mathfrak{A}}$$

(und zwar so angeordnet, dass man beim Durchlaufen des Kreises $\mathfrak{A}$ im Uhrzeigersinn diese Knoten in der Reihenfolge $v^-, v, v^+, \ldots, w^-, w, w^+$ betritt).

In Phase 2 lassen wir Duplicator als $\mathfrak{B} \in$ UGraphs $\setminus$ Conn den Graphen $\mathfrak{B} = (B, E^{\mathfrak{B}})$ wählen, dessen Knotenmenge $B := A$ ist, und dessen Kantenmenge aus der Kantenmenge $E^{\mathfrak{A}}$ entsteht, indem man die Kanten zwischen v und v^+ sowie die Kanten zwischen w und w^+ löscht und stattdessen neue Kanten zwischen v und w^+ sowie zwischen w und v^+ einfügt. Der so konstruierte Graph besteht offensichtlich aus zwei disjunkten Kreisen.

Des Weiteren wählt Duplicator die Mengen $S_1' := S_1, \ldots, S_l' := S_l$. Man sieht leicht, dass für $\vec{S}' := S_1', \ldots, S_l'$ in der zugehörigen Struktur $\hat{\mathfrak{B}} := (\mathfrak{B}, \vec{S}')$

$$\text{für jeden Knoten } u \in B = A \text{ gilt:} \quad \text{Typ}(2^m, u, \hat{\mathfrak{B}}) \;\cong\; \text{Typ}(2^m, u, \hat{\mathfrak{A}}). \tag{3.6}$$

Phase 3: Wir müssen zeigen, dass Duplicator eine Gewinnstrategie im m-Runden EF-Spiel auf den Strukturen $\hat{\mathfrak{A}}$ und $\hat{\mathfrak{B}}$ hat. Dazu können wir den Satz von Hanf (Korollar 3.26) anwenden, denn aus (3.6) folgt, dass für jeden 2^m-Umgebungstyp ρ gilt: $\#_\rho(\hat{\mathfrak{A}}) = \#_\rho(\hat{\mathfrak{B}})$. Korollar 3.26 liefert daher, dass $\hat{\mathfrak{A}} \equiv_m \hat{\mathfrak{B}}$. D.h. Duplicator gewinnt das m-Runden EF-Spiel auf $\hat{\mathfrak{A}}$ und $\hat{\mathfrak{B}}$. Insgesamt haben wir damit eine Gewinnstrategie für Duplicator im (l, m)-Ajtai-Fagin-Spiel für Conn auf UGraphs konstruiert. □

Das obige Theorem besagt, dass Graph-Zusammenhang nicht in *Mon*-Σ^1_1 definierbar ist. In der monadischen *universellen* Logik zweiter Stufe *Mon*-Π^1_1 kann Graph-Zusammenhang allerdings leicht definiert werden (siehe Beispiel 2.5 (b)).

Des Weiteren ist Graph-Zusammenhang ein Problem, das zur Klasse NP gehört (es gehört sogar zur Klasse PTIME) und kann daher nach dem Satz von Fagin (Theorem 2.8) auch durch eine ESO-Formel beschrieben werden. Zusammen mit Bemerkung 3.39 erhalten wir also:

$$\text{FO} \;<\; \textit{Mon}\text{-}\Sigma^1_1 \;<\; \text{ESO} \quad \text{auf } \mathsf{Fin}.$$

Abschließend sei noch angemerkt, dass sich natürlich leicht eine Variante des Ajtai-Fagin Spiels finden lässt, die an Stelle der *Mon*-Σ^1_1-Definierbarkeit allgemein ESO-Definierbarkeit charakterisiert: an Stelle einer gegebenen Zahl l (die angibt, wie viele 1-stellige Relationen in den Phasen 1 und 2 gewählt werden) brauchen wir dazu nur eine Liste $(s_1, \ldots, s_l)$ von Zahlen zu betrachten, die angibt, dass in den Phasen 1 und 2 des Spiels Relationen der Stelligkeiten $s_1, \ldots, s_l$ gewählt werden.

3.3 Pebble-Spiele und infinitäre Logiken

In diesem Abschnitt werden wir sogenannte *infinitäre* Logiken behandeln, d.h. Logiken, deren Formeln unendliche Länge haben können. Solche Logiken werden vor allem im Bereich der unendlichen Modelltheorie untersucht. Für die endliche Modelltheorie, mit der wir uns hier beschäftigen, werden sie sich in ihrer allgemeinen Form als zu ausdrucksstark herausstellen. Schränkt man hingegen die Anzahl der erlaubten Variablen ein, so erhält man schwächere Logiken, die für die endliche Modelltheorie wichtige Erkenntnisse liefern.

3.3.1 Die infinitäre Logik $L_{\infty\omega}$

3.43 Definition. Sei σ eine Signatur.
Die Formelmenge $L_{\infty\omega}[\sigma]$ ist induktiv wie folgt definiert.

- $L_{\infty\omega}[\sigma]$ enthält alle atomaren σ-Formeln.
- Ist φ eine $L_{\infty\omega}[\sigma]$-Formel, so ist auch $\neg\varphi$ eine $L_{\infty\omega}[\sigma]$-Formel.
- Ist φ eine $L_{\infty\omega}[\sigma]$-Formel und ist x eine Variable erster Stufe, so sind auch $\exists x\varphi$ und $\forall x\varphi$ Formeln in $L_{\infty\omega}[\sigma]$.
- Ist Ψ eine *Menge* von $L_{\infty\omega}[\sigma]$-Formeln, so sind auch $\bigvee \Psi$ und $\bigwedge \Psi$ Formeln in $L_{\infty\omega}[\sigma]$. (*Beachte:* Hierbei darf Ψ auch unendlich sein.)

Die Semantik der Logik $L_{\infty\omega}$ ist die naheliegende Erweiterung der Semantik für FO. Hierbei wird $\bigvee \Psi$ als Disjunktion über alle Formeln in Ψ und entsprechend $\bigwedge \Psi$ als Konjunktion über alle Formeln in Ψ interpretiert. Das heißt (für eine Satzmenge Ψ), dass $\mathfrak{A} \models \bigvee \Psi$ genau dann gilt, wenn es (mindestens) einen Satz $\psi \in \Psi$ gibt mit $\mathfrak{A} \models \psi$. Analog dazu gilt

$\mathfrak{A} \models \bigwedge \Psi$ genau dann, wenn für jeden Satz $\psi \in \Psi$ gilt $\mathfrak{A} \models \psi$.
Offensichtlich ist $L_{\infty\omega}$ eine Erweiterung der Logik erster Stufe. Wir schauen uns zunächst einige Beispiele für $L_{\infty\omega}$-Formeln an:

3.44 Beispiel. Für jedes $n \in \mathbb{N}_{\geqslant 1}$ sei φ_n der FO-Satz

$$\varphi_n \;:=\; \exists x_1 \cdots \exists x_n \; \Big(\bigwedge_{1 \leqslant i < j \leqslant n} \neg x_i{=}x_j \;\wedge\; \forall y \bigvee_{i=1}^{n} x_i{=}y \Big),$$

der besagt, dass es genau n Elemente in den Modellen von φ_n gibt. Nun gilt folgendes:

(1) Für jede Signatur σ ist $\psi := \bigvee \{\varphi_n : n \in \mathbb{N}\}$ eine $L_{\infty\omega}[\sigma]$-Formel, die die Klasse aller *endlichen* σ-Strukturen definiert, das heißt: $\mathrm{Mod}_{\mathsf{All}}(\psi) = \mathsf{Fin}$.

(2) Analog definiert der $L_{\infty\omega}[\sigma]$-Satz $\psi_{\mathsf{Even}} := \bigvee\{\varphi_n : n \in \mathbb{N} \text{ und } n \text{ gerade }\}$ die Klasse aller endlichen Strukturen gerader Kardinalität, das heißt $\mathrm{Mod}_{\mathsf{All}}(\psi_{\mathsf{Even}}) = \mathsf{Even}$, wobei Even die Klasse aller endlichen σ-Strukturen $\mathfrak{A}$ bezeichnet, deren Universum aus einer geraden Anzahl von Elementen besteht.

Wir wissen bereits, dass die Klasse aller endlichen Strukturen gerader Kardinalität nicht in FO definierbar ist. Die Logik $L_{\infty\omega}$ ist also echt ausdruckstärker als FO, was angesichts der sehr allgemeinen Definition auch nicht verwundern dürfte.

Wie anfangs erwähnt, spielt die Logik $L_{\infty\omega}$ in der unendlichen Modelltheorie eine wichtige Rolle. Das nächste Beispiel zeigt jedoch, dass sie für die *endliche* Modelltheorie schon zu ausdrucksstark ist, weil einfach *jede* Klasse endlicher Strukturen durch eine $L_{\infty\omega}$-Formel beschrieben werden kann.

3.45 Beispiel. Sei σ eine Signatur und C eine beliebige unter Isomorphie abgeschlossene Klasse endlicher σ-Strukturen. C wird definiert durch den $L_{\infty\omega}[\sigma]$-Satz $\varphi_{\mathsf{C}} := \bigvee\{\varphi_{\mathfrak{A}} : \mathfrak{A} \in \mathsf{C}\}$, wobei $\varphi_{\mathfrak{A}}$ die FO-Formel aus Proposition 3.2 ist, die die Struktur $\mathfrak{A}$ bis auf Isomorphie beschreibt. Das heißt: $\mathrm{Mod}_{\mathsf{All}}(\varphi_{\mathsf{C}}) = \mathsf{C}$.

Wie das Beispiel zeigt, ist also jede Klasse endlicher Strukturen in $L_{\infty\omega}$ definierbar. Wir werden daher geeignete Einschränkungen der Logik definieren müssen, um für die endliche Modelltheorie interessante Aussagen treffen zu können.

3.3.2 Das k-Variablen Fragment von FO und $L_{\infty\omega}$

3.46 Definition (FO^k)**.** Sei σ eine Signatur und sei $k \in \mathbb{N}$. Die Klasse FO$^k[\sigma]$ besteht aus allen FO$[\sigma]$-Formeln, in denen höchstens k verschiedene Variablen erster Stufe vorkommen.

3.47 Beispiel. Für jedes $\ell \in \mathbb{N}$ gibt es eine FO$^2[<]$-Formel $\psi_\ell(x)$, so dass für jede endliche linear geordnete Struktur $\mathfrak{A} := (A, <^{\mathfrak{A}})$ und jedes $a \in A$ gilt: $\mathfrak{A} \models \psi_\ell[a] \iff a$ ist das

ℓ-te Element bezüglich der Ordnung $<^{\mathfrak{A}}$ ist, d.h. $\ell = rg_{<^{\mathfrak{A}}}(a)$. Die Formel ψ_ℓ definieren wir induktiv durch

$$\begin{aligned} \psi_0(x) &:= \forall y\, \neg y{<}x \\ \psi_{\ell+1}(x) &:= \forall y \Big(y{<}x \leftrightarrow \bigvee_{i=0}^{\ell} \exists x \, \big(x{=}y \wedge \psi_i(x) \big) \Big). \end{aligned}$$

Die Konstruktion $\exists x \, \big(x{=}y \wedge \psi_i(x) \big)$ wird benutzt, da ψ_ℓ die Variable x und nicht y als freie Variable enthält.

Die Formel $\psi_{\ell+1}(x)$ besagt, dass alle Elemente $y < x$ höchstens den Rang ℓ in der Ordnung haben können. Also kann x höchstens den Rang $\ell{+}1$ haben. Andererseits kann x keinen Rang $\leqslant \ell$ haben, da andernfalls für $y = x$ die rechte Seite der Biimplikation erfüllt wäre, die linke aber nicht.

An dieser Stelle sei darauf hingewiesen, dass man mit Substitutionen im Zusammenhang mit k-Variablen Logiken vorsichtig sein muss. Substituiert man einfach x durch y in $\psi_i(x)$, so würde, gemäß der Definition von Substitutionen, zunächst die gebunden vorkommende Variable y umbenannt, d.h. durch ein neues Variablensymbol ersetzt, und danach dann jedes frei vorkommende x durch y ersetzt. Hierbei ist aber nicht von vorneherein klar, dass damit nicht mehr als insgesamt zwei Variablen benutzt werden. Daher werden wir im Folgenden Substitutionen im Bezug auf k-Variablen Logiken vermeiden und lieber explizite Variablenumbenennungen verwenden.

Wir definieren nun das entsprechende k-Variablen Fragment der infinitären Logik $L_{\infty\omega}$.

3.48 Definition ($L^k_{\infty\omega}$)**.** Sei σ eine Signatur und sei $k \in \mathbb{N}$. Die Formelklasse $L^k_{\infty\omega}[\sigma]$ ist definiert als die Klasse aller $L_{\infty\omega}[\sigma]$-Formeln, die höchstens k verschiedene Variablen erster Stufe enthalten. Des Weiteren sei $L^\omega_{\infty\omega}[\sigma] := \bigcup_{k\in\mathbb{N}} L^k_{\infty\omega}[\sigma]$.

$L^\omega_{\infty\omega}$ ist also die Klasse aller $L_{\infty\omega}$-Formeln, in denen nur endlich viele Variablen benutzt werden. Als Ausblick auf Kapitel 5 sei erwähnt, dass sich die bisher behandelten Fixpunktlogiken LFP, IFP, PFP sämtlich in $L^\omega_{\infty\omega}$ einbetten lassen, es gilt also $\text{PFP} \leqslant L^\omega_{\infty\omega}$. Insbesondere übertragen sich also Nicht-Definierbarkeits-Resultate für $L^\omega_{\infty\omega}$ auch auf die Fixpunktlogiken.

3.49 Beispiel. Für jede Menge $J \subseteq \mathbb{N}_{\geqslant 1}$ gibt es einen $L^3_{\infty\omega}[<]$-Satz φ_J, so dass

$$\mathrm{Mod}_{\mathsf{All}}(\varphi_J) \;=\; \{\, \mathfrak{A} = (A, <^{\mathfrak{A}}) \;:\; <^{\mathfrak{A}} \text{ ist eine lineare Ordnung auf } A \text{ und } |A| \in J \,\}.$$

Der Satz φ_J ist folgendermaßen konstruiert: Sei $\psi_{\text{Ord}} \in \text{FO}^3[<]$ ein Satz, der besagt, dass $<$ eine lineare Ordnung ist. Außerdem sei $\psi_\ell(x)$, für jedes $\ell \in \mathbb{N}$, der $\text{FO}^2[<]$-Satz aus Beispiel 3.47. Dann ist ψ_J definiert als

$$\psi_J \;:=\; \psi_{\text{Ord}} \wedge \bigvee \{\, \exists x\, \psi_{\ell-1}(x) \wedge \neg\exists x\, \psi_\ell(x) \;:\; \ell \in J \,\}.$$

Wie dieses Beispiel zeigt, gibt es Strukturklassen, die schon in der 3-Variablen-Logik $L^3_{\infty\omega}$ definiert werden können, die aber nicht in FO definierbar sind (mit beliebig vielen Variablen). Wie der nächste Satz allerdings zeigt, können zwei *endliche* Strukturen, die in $L^k_{\infty\omega}$ unterschieden werden können, auch schon in FO^k unterschieden werden.

3.50 Definition. Sei $k \in \mathbb{N}_{\geqslant 1}$, σ eine Signatur und $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen. Wir schreiben $\mathfrak{A} \equiv_{\text{FO}^k} \mathfrak{B}$ (bzw. $\mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$), falls $\mathfrak{A}$ und $\mathfrak{B}$ dieselben $\text{FO}^k[\sigma]$-Sätze (bzw. $L^k_{\infty\omega}[\sigma]$-Sätze) erfüllen.

3.51 Satz. *Für alle endlichen σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ gilt:* $\mathfrak{A} \equiv_{\text{FO}^k} \mathfrak{B} \iff \mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$.

Beweis: "$\Longleftarrow$": klar, da $\text{FO}^k \subseteq L^k_{\infty\omega}$.
"$\Longrightarrow$": Per Induktion nach dem Aufbau von $L^k_{\infty\omega}$. zeigen wir, dass es zu jeder $L^k_{\infty\omega}[\sigma]$-Formel $\varphi(\vec{x})$ eine $\text{FO}^k[\sigma]$-Formel $\tilde{\varphi}(\vec{x})$ gibt, so dass für alle $\vec{a} \in A$, $\vec{b} \in B$ gilt:

$$\mathfrak{A} \models \varphi[\vec{a}] \iff \mathfrak{A} \models \tilde{\varphi}[\vec{a}] \qquad \text{und} \qquad \mathfrak{B} \models \varphi[\vec{b}] \iff \mathfrak{B} \models \tilde{\varphi}[\vec{b}]. \tag{3.7}$$

Der einzige nicht-triviale Fall ist, dass φ von der Form $\bigvee \Psi$ oder $\bigwedge \Psi$ ist, wobei Ψ eine Menge von $L^k_{\infty\omega}$-Formeln ist. Wir betrachten hier den Fall $\bigvee \Psi$, der andere ist dann analog.

Sei also Ψ eine Menge von $L^k_{\infty\omega}$-Formeln und $\varphi := \bigvee \Psi$. Für jedes $\vec{a} \in A$ mit $\mathfrak{A} \models \varphi[\vec{a}]$ wähle eine Formel $\psi_{\vec{a}} \in \Psi$, so dass $\mathfrak{A} \models \psi[\vec{a}]$. Analog wähle für jedes $\vec{b} \in B$ mit $\mathfrak{B} \models \varphi[\vec{b}]$ eine Formel $\psi_{\vec{b}} \in \Psi$, so dass $\mathfrak{B} \models \psi_{\vec{b}}[\vec{b}]$. Nun definieren wir

$$\Psi_{\mathfrak{A},\mathfrak{B}} := \left\{ \psi_{\vec{a}} : \vec{a} \in A \text{ und } \mathfrak{A} \models \varphi[\vec{a}] \right\} \cup \left\{ \psi_{\vec{b}} : \vec{b} \in B \text{ und } \mathfrak{B} \models \varphi[\vec{b}] \right\}.$$

Nach Konstruktion ist $\Psi_{\mathfrak{A},\mathfrak{B}}$ eine *endliche* Teilmenge von Ψ. Weiterhin gilt für alle $\vec{a} \in A, \vec{b} \in B$:

$$\mathfrak{A} \models \bigvee \Psi_{\mathfrak{A},\mathfrak{B}}[\vec{a}] \iff \mathfrak{A} \models \bigvee \Psi[\vec{a}] \iff \mathfrak{A} \models \varphi[\vec{a}]$$

sowie

$$\mathfrak{B} \models \bigvee \Psi_{\mathfrak{A},\mathfrak{B}}[\vec{b}] \iff \mathfrak{B} \models \bigvee \Psi[\vec{b}] \iff \mathfrak{B} \models \varphi[\vec{b}].$$

Gemäß Induktionsvoraussetzung ist jede Formel $\psi \in \Psi_{\mathfrak{A},\mathfrak{B}}$ äquivalent zu einer Formel in FO^k. Also ist auch $\bigvee \Psi_{\mathfrak{A},\mathfrak{B}}$ äquivalent zu einer Formel $\tilde{\varphi}$ in FO^k. Es gilt also für alle $\vec{a} \in A$ und $\vec{b} \in B$: $\mathfrak{A} \models \varphi[\vec{a}] \iff \mathfrak{A} \models \tilde{\varphi}[\vec{a}]$ und $\mathfrak{B} \models \varphi[\vec{b}] \iff \mathfrak{B} \models \tilde{\varphi}[\vec{b}]$. Somit ist (3.7) gezeigt. Aus (3.7) folgt insbesondere folgendes: Falls es einen $L^k_{\infty\omega}$-Satz φ gibt, der zwischen $\mathfrak{A}$ und $\mathfrak{B}$ unterscheidet, so gibt es auch einen FO^k-Satz, der zwischen $\mathfrak{A}$ und $\mathfrak{B}$ unterscheidet. Dies schließt den Beweis von "$\Longrightarrow$" ab. □

3.3.3 Pebble-Spiele

Ziel dieses Abschnitts ist es, Ehrenfeucht-Fraïssé-Spiele für die Logiken FO^k und $L^k_{\infty\omega}$ einzuführen. Dazu zunächst ein paar Notationen.

Der Einfachheit halber werden wir in diesem Abschnitt nur Signaturen betrachten, die *keine Konstantensymbole* enthalten. Solche Signaturen nennen wir *relationale Signaturen*.

3.52 Notation. Sei σ eine relationale Signatur, $k \in \mathbb{N}_{\geqslant 1}$ und $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen.

- Wir vereinbaren, dass das Symbol "$*$" in keinem Universum einer Struktur vorkommt.
- Für $\vec{a} := a_1, \ldots, a_k \in (A \dot\cup \{*\})^k$ definieren wir den *Träger* $\mathrm{Tr}(\vec{a})$ *von* $\vec{a}$ als
$$\mathrm{Tr}(\vec{a}) \;:=\; \{i : a_i \neq *\}.$$
- Für $i \in \{1, \ldots, k\}$, $a \in A$ und $\vec{a} := a_1 \ldots a_k \in (A \cup \{*\})^k$ setzen wir
$$\vec{a}\tfrac{a}{i} \;:=\; a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_k\,.$$
Das heißt, wir ersetzen die i-te Stelle von $\vec{a}$ durch a.
- Für $\vec{a} = a_1, \ldots, a_k \in (A \cup \{*\})^k$ schreiben wir $\vec{a}_{|\mathrm{Tr}(\vec{a})}$ um das Tupel über A zu bezeichnen, das aus $\vec{a}$ durch Löschen der "$*$"-Symbole entsteht.
- Für $\vec{a}$ und $\vec{b}$ mit $\mathrm{Tr}(\vec{a}) = \mathrm{Tr}(\vec{b})$ schreiben wir $(\vec{a} \mapsto \vec{b})_{|\mathrm{Tr}(\vec{a})}$ um die Abbildung $\big(\vec{a}_{|\mathrm{Tr}(\vec{a})} \mapsto \vec{b}_{|\mathrm{Tr}(\vec{b})}\big)$ zu bezeichnen.

3.53 Definition (k-partieller Isomorphismus)**.** Sei σ eine relationale Signatur, $k \in \mathbb{N}_{\geqslant 1}$, $\mathfrak{A}, \mathfrak{B}$ seien σ-Strukturen, und seien $\vec{a} \in (A \cup \{*\})^k$ und $\vec{b} \in (B \cup \{*\})^k$. Die Abbildung $(\vec{a} \mapsto \vec{b})$ heißt *k-partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$*, falls

(1) $\mathrm{Tr}(\vec{a}) = \mathrm{Tr}(\vec{b})$ und

(2) $(\vec{a} \mapsto \vec{b})_{|\mathrm{Tr}(\vec{a})}$ ist ein partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

$\mathrm{Part}^k(\mathfrak{A}, \mathfrak{B})$ bezeichnet die Menge aller k-partiellen Isomorphismen von $\mathfrak{A}$ nach $\mathfrak{B}$.

Man beachte, dass der Definitionsbereich jedes k-partiellen Isomorphismus höchstens k Elemente enthält.

Wir sind nun bereit, die Ehrenfeucht-Fraïssé-Spiele für die Logiken FO^k und $L^k_{\infty\omega}$ einzuführen.

3.54 Definition (Pebble-Spiele)**.** Sei σ eine relationale Signatur und seien $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen. Seien $k \in \mathbb{N}_{\geqslant 1}$, $\vec{a} \in (A \dot\cup \{*\})^k$ und $\vec{b} \in (B \dot\cup \{*\})^k$ mit $\mathrm{Tr}(\vec{a}) = \mathrm{Tr}(\vec{b})$.

Das *k-Pebble-Spiel* $\mathcal{G}^k_\infty(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})$ wird zwischen zwei Spielern, Spoiler und Duplicator, gespielt. Den Spielern stehen insgesamt $2{\cdot}k$ Spielsteine $\alpha_1, \ldots, \alpha_k$, $\beta_1, \ldots, \beta_k$ zur Verfügung, die auf Elemente der Strukturen gelegt werden können. Zu Beginn des Spiels liegt für jedes $i \in \mathrm{Tr}(\vec{a})$ der Stein α_i auf dem Element a_i und β_i auf b_i. Die übrigen Steine liegen "nebem dem Spielbrett".

Eine Partie des Spiels besteht aus einer unbegrenzten Anzahl von Runden. In jeder Runde wählt Spoiler zunächst ein beliebiges $i \in \{1, \ldots, k\}$ und nimmt entweder den Stein α_i und legt ihn auf ein beliebiges Element in A, oder er nimmt Stein β_i und legt ihn auf ein beliebiges Element in B. Duplicator antwortet, indem er den entsprechenden Stein (also β_i oder α_i) auf ein beliebiges Element in der anderen Struktur legt. D.h., Duplicator legt Stein β_i auf ein Element in B, falls Spoiler den Stein α_i gelegt hat; bzw. Duplicator legt Stein α_i auf ein Element in A, falls Spoiler den Stein β_i gelegt hat. Beachte:

- Steine, die bereits auf dem Spielfeld liegen, dürfen wiederverwendet werden.
- Es dürfen durchaus mehrere Steine auf demselben Element liegen.

Am Ende jeder Runde wird entschieden, ob Spoiler gewonnen hat (und das Spiel beendet ist) oder ob weitergespielt wird. Dazu seien $\vec{a} = a_1, \ldots, a_k \in (A \cup \{*\})^k$ die am Ende der Runde durch die Steine $\alpha_1, \ldots, \alpha_k$ in $\mathfrak{A}$ markierten Elemente (mit $a_j = *$ falls der Stein α_j noch "neben dem Spielbrett" liegt), und $\vec{b} = b_1, \ldots, b_k \in (B \cup \{*\})^k$ seien die entsprechenden durch die Steine $\beta_1, \ldots, \beta_k$ in $\mathfrak{B}$ markierten Elemente. Ist die Abbildung $(\vec{a} \mapsto \vec{b})$ *kein* k-partieller Isomorphismus, so endet das Spiel nach dieser Runde und *Spoiler gewinnt*. Andernfalls wird das Spiel mit einer weiteren Runde fortgesetzt.

Duplicator gewinnt, wenn unendlich lange gespielt wird, also nach jeder Runde die Abbildung $(\vec{a} \mapsto \vec{b})$ ein k-partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$ ist.

Bemerkung:

- Ist $\vec{a} = \vec{b} = \vec{*} = *, \ldots, *$, so schreiben wir $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$ an Stelle von $\mathcal{G}^k_\infty(\mathfrak{A}, \vec{*}, \mathfrak{B}, \vec{*})$.
- Strategien und Gewinnstrategien im k-Pebble-Spiel sind analog zum herkömmlichen Ehrenfeucht-Fraïssé-Spiel definiert und werden daher hier nicht mehr formal eingeführt.
- Sind die Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ *endlich*, so gibt es nur eine endliche Zahl verschiedener Spielpositionen $(\vec{a}, \vec{b}) \in (A \cup \{*\})^k \times (B \cup \{*\})^k$. In diesem Fall steht also schon nach einer *endlichen* Zahl von Zügen fest, wer das Spiel gewinnen kann.

3.55 Beispiele. (a) Sei $\sigma := \varnothing$ und $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen mit $|A|, |B| \geqslant k$. Dann hat *Duplicator* eine Gewinnstrategie in $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$, indem er immer wenn Spoiler zwei Steine auf dasselbe Element legt ebenso zieht und ansonsten immer ein neues Element mit einem Stein belegt. Da es nicht mehr Steine als Elemente in den Strukturen gibt, kann er dies immer sicherstellen.

(b) Ist $\sigma = \varnothing$ und $\mathfrak{A}, \mathfrak{B}$ sind σ-Strukturen mit $|A| < k$ und $|B| \neq |A|$, so hat *Spoiler* eine Gewinnstrategie in $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$.

(c) Seien jetzt $\mathfrak{A}, \mathfrak{B}$ endliche, linear geordnete $\{<\}$-Strukturen. Dann hat Duplicator genau dann eine Gewinnstrategie im Spiel $\mathcal{G}^2_\infty(\mathfrak{A}, \mathfrak{B})$, wenn $|A| = |B|$. Dies kann man folgendermaßen sehen:

Gilt $|A| = |B|$ so sind $\mathfrak{A}$ und $\mathfrak{B}$ zwei endliche lineare Ordnungen gleicher Kardinalität und somit isomorph. Folglich hat Duplicator eine Gewinnstratgie, indem er immer zu Spoilers Wahl isomorphe Elemente wählt.

Gilt $|A| \neq |B|$ und o.B.d.A $|A| > |B|$, so hat Spoiler eine Gewinnstratgie in $\mathcal{G}^2_\infty(\mathfrak{A}, \mathfrak{B})$, indem er in jeder Runde $r \geqslant 1$ den Stein $\alpha_{1+(r-1 \bmod 2)}$ auf das Element mit Rang $r-1$ in $\mathfrak{A}$ legt.

In den ersten beiden Runden legt Spoiler also seine beiden Spielsteine α_1, α_2 auf die beiden kleinsten Elemente in $\mathfrak{A}$. In den folgenden Runden nimmt er jeweils den Stein auf dem kleineren Element und plaziert ihn auf das kleinste noch nicht im Spiel verwendete Element. Nach jeder Runde r liegen also die Steine α_1, α_2 auf den Elemeten mit Rang $r-2$ und $r-1$. Auf diese Weise werden im Verlauf des Spiels alle Elemente von $\mathfrak{A}$ in ihrer Reihenfolge gemäß der Ordnung durchlaufen.

Duplicator muss nun ebenfalls in jedem Zug den Stein auf dem kleineren der beiden Elemente in $\mathfrak{B}$ auf ein größeres legen, denn ansonsten wäre die Abbildung $\alpha_1, \alpha_2 \mapsto \beta_1, \beta_2$ kein k-partieller Isomorphismus und Duplicator hätte verloren. Da aber $|B| < |A|$ ist, kann Duplicator dies nach spätestens $|B|$ Runden nicht mehr gewährleisten und verliert daher das Spiel.

Analog zum Satz von Ehrenfeucht und Fraïssé werden wir nun den Zusammenhang zwischen Pebble-Spielen und der Logik $L^{\omega}_{\infty\omega}$ herstellen. Dazu benötigen wir zunächst eine geeignete Variante von Hin-und-Her-Systemen (vgl. Definition 3.32).

3.56 Definition. Sei σ eine relationale Signatur und $k \in \mathbb{N}_{\geqslant 1}$. Zwei σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ heißen *k-partiell isomorph* (kurz: $\mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$), falls es eine *nicht-leere* Menge I k-partieller Isomorphismen gibt, die die folgenden Eigenschaften erfüllt:

k-Hin-Eigenschaft: Für alle $(\vec{a} \mapsto \vec{b}) \in I$, alle $i \in \{1, \ldots, k\}$ und alle $a \in A$ gibt es ein $b \in B$, so dass $(\vec{a}\frac{a}{i} \mapsto \vec{b}\frac{b}{i}) \in I$.

k-Her-Eigenschaft: Für alle $(\vec{a} \mapsto \vec{b}) \in I$, alle $i \in \{1, \ldots, k\}$ und alle $b \in B$ gibt es ein $a \in A$, so dass $(\vec{a}\frac{a}{i} \mapsto \vec{b}\frac{b}{i}) \in I$.

Ein System mit diesen Eigenschaften nennen wir *k-Hin-und-Her-System*. Wir schreiben $I : \mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$ um anzudeuten, dass I ein k-Hin-und-Her-System zwischen $\mathfrak{A}$ und $\mathfrak{B}$ ist.

3.57 Bemerkung. Die Menge

$$W^k_\infty(\mathfrak{A}, \mathfrak{B}) := \left\{ (\vec{a} \mapsto \vec{b}) \in \text{Part}^k(\mathfrak{A}, \mathfrak{B}) \;:\; \begin{array}{l}\text{Duplicator hat eine Gewinnstrategie im} \\ k\text{-Pebble-Spiel } \mathcal{G}^k_\infty(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})\end{array} \right\}$$

hat die k-Hin-und-Her-Eigenschaft, ist aber möglicherweise leer.

3.58 Theorem. *Sei σ eine relationale Signatur, $k \in \mathbb{N}_{\geqslant 1}$, $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen, $\vec{a} \in (A \cup \{*\})^k, \vec{b} \in (B \cup \{*\})^k$ mit $\text{Tr}(\vec{a}) = \text{Tr}(\vec{b})$. Dann sind folgende Aussagen äquivalent:*

(a) Duplicator hat eine Gewinnstrategie im k-Pebble-Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})$.

(b) $(\vec{a} \mapsto \vec{b}) \in W^k_\infty(\mathfrak{A}, \mathfrak{B})$ und $W^k_\infty : \mathfrak{A} \cong^k_{part} \mathfrak{B}$.

(c) Es gibt ein k-Hin-und-Her-System $I : \mathfrak{A} \cong^k_{part} \mathfrak{B}$ mit $(\vec{a} \mapsto \vec{b}) \in I$.

(d) $\vec{a}$ in $\mathfrak{A}$ und $\vec{b}$ in $\mathfrak{B}$ erfüllen dieselben $L^k_{\infty\omega}[\sigma]$-Formeln.

Beweis: "(a) $\Longrightarrow$ (b)" folgt direkt aus Definition 3.56 und Bemerkung 3.57.
"(b) $\Longrightarrow$ (c)" ist trivial.
"(c) $\Longrightarrow$ (d)" folgt leicht per Induktion nach dem Aufbau von $L^k_{\infty\omega}$ (Details: *Übung*).
"(d) $\Longrightarrow$ (a)": Wir zeigen, dass Duplicator eine Strategie im Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})$ hat, so dass nach jeder Runde r des Spiels die folgende Invariante erhalten bleibt:

$(*)$: Sind $\vec{a}$ und $\vec{b}$ die am Ende der Runde mit den Steinen $\alpha_1, \ldots, \alpha_k$ und $\beta_1, \ldots, \beta_k$ in $\mathfrak{A}$ bzw. $\mathfrak{B}$ belegten Elemente (incl. $*$), so erfüllt $\vec{a}$ dieselben $L^k_{\infty\omega}[\sigma]$-Formeln in $\mathfrak{A}$ wie $\vec{b}$ in $\mathfrak{B}$.

Offensichtlich gilt nach jedem Zug, bei dem die Invariante $(*)$ erhalten bleibt, dass $(\vec{a} \mapsto \vec{b}) \in \text{Part}^k(\mathfrak{A}, \mathfrak{B})$. Um zu zeigen, dass Duplicator eine Gewinnstrategie in $\mathcal{G}^k_\infty(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})$ hat, genügt es also, zu zeigen, dass Duplicator so spielen kann, dass stets die Invariante $(*)$ erfüllt ist.

Nach Voraussetzung wissen wir, dass (d) gilt, und dass daher die Invariante $(*)$ zu Beginn des Spiels erfüllt ist. Für den Induktionsschritt gelte nun die Invariante $(*)$ nach der r-ten Runde (für $r \in \mathbb{N}$). Wir müssen zeigen, dass Duplicator die $r{+}1$-te Runde so spielen kann, dass $(*)$ auch nach der $r{+}1$-ten Runde erfüllt ist. Dazu betrachten wir den Fall, dass Spoiler in der $r{+}1$-ten Runde den Spielstein α_i (für ein $i \in \{1, \ldots, k\}$) auf ein Element $a \in A$ legt (der Fall, dass Spoiler in $\mathfrak{B}$ zieht, kann analog behandelt werden). Sei nun

$$\Psi \;:=\; \Big\{\, \psi \in L^k_{\infty\omega} \;:\; \big(\mathfrak{A}, \vec{a}\tfrac{a}{i}\big) \models \psi \,\Big\}.$$

Für diese Formelmenge Ψ gilt offensichtlich, dass $\big(\mathfrak{A}, \vec{a}\big) \;\models\; \exists x_i \bigwedge \Psi$.
Wegen $(*)$ gilt daher auch $\big(\mathfrak{B}, \vec{b}\big) \;\models\; \exists x_i \bigwedge \Psi$.
Also existiert ein $b \in B$, so dass für alle $\psi \in \Psi$ gilt: $(\mathfrak{B}, \vec{b}\tfrac{b}{i}) \models \psi$. Duplicator antwortet in Runde $r{+}1$ nun, indem er den Spielstein β_i auf dieses b legt. Dann gilt

$$(\mathfrak{A}, \vec{a}\tfrac{a}{i}) \;\models\; \bigwedge \Psi \qquad \text{und} \qquad (\mathfrak{B}, \vec{b}\tfrac{b}{i}) \;\models\; \bigwedge \Psi.$$

Außerdem gilt natürlich für jede $L^k_{\infty\omega}$-Formel χ, dass entweder $\chi \in \Psi$ oder $\neg\chi \in \Psi$. Daher gilt für alle $\chi \in L^k_{\infty\omega}$, dass $(\mathfrak{A}, \vec{a}\tfrac{a}{i}) \models \chi \iff (\mathfrak{B}, \vec{b}\tfrac{b}{i}) \models \chi$. Somit ist die Invariante $(*)$ nach Runde $r{+}1$ erfüllt. □

Folgendes Korollar folgt nun sofort aus Theorem 3.58 und Satz 3.51.

3.59 Korollar. *Sei σ eine relationale Signatur, $k \in \mathbb{N}_{\geqslant 1}$ und $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen. Dann sind folgende Aussagen äquivalent:*

(a) Duplicator hat eine Gewinnstrategie im k-Pebble-Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$.

(b) $W^k_\infty(\mathfrak{A}, \mathfrak{B}) : \mathfrak{A} \cong^k_{part} \mathfrak{B}$.

(c) $\mathfrak{A} \cong^k_{part} \mathfrak{B}$.

(d) $\mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$.

Sind $\mathfrak{A}$ und $\mathfrak{B}$ endlich, ist folgende Aussage auch noch äquivalent zu den vorherigen:

(e) $\mathfrak{A} \equiv_{\mathrm{FO}^k} \mathfrak{B}$.

3.60 Korollar. *Sei σ eine relationale Signatur,* S *eine Klasse von σ-strukturen,* $\mathsf{C} \subseteq \mathsf{S}$. *Falls die folgende Bedingung $(*)$ erfüllt ist, so ist* C nicht $L^\omega_{\infty\omega}$*-definierbar in* S.

$(*)$: *Für jedes $k \in \mathbb{N}_{\geqslant 1}$ gibt es $\mathfrak{A} \in \mathsf{C}$ und $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$, so dass Duplicator eine Gewinnstrategie im k-Pebble-Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$ hat.*

Beweis: Es sei $(*)$ erfüllt. Angenommen, C ist $L^\omega_{\infty\omega}$-definierbar in C, etwa durch den $L^k_{\infty\omega}[\sigma]$-Satz φ, für ein $k \in \mathbb{N}$. Es gilt also $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\varphi)$. Nach Voraussetzung gilt $(*)$, d.h. es gibt es Strukturen $\mathfrak{A} \in \mathsf{C}$ und $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$, so dass $\mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$ (wegen Korollar 3.59). Da $\mathfrak{A} \in \mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\varphi)$ ist, gilt $\mathfrak{A} \models \varphi$ und somit, wegen $\mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$, auch $\mathfrak{B} \models \varphi$. Dies ist aber ein Widerspruch zu $\mathfrak{B} \in \mathsf{S} \setminus \mathsf{C}$. □

3.61 Beispiel. Die Klasse

$$\mathsf{Even} := \left\{ \mathfrak{A} \,:\, \mathfrak{A} \text{ ist eine endliche } \varnothing\text{-Struktur mit } |A| \text{ gerade} \right\}$$

ist nicht $L^\omega_{\infty\omega}$-definierbar in der Klasse aller endlichen Strukturen, denn:

Wie in Beispiel 3.55 gezeigt, hat Duplicator eine Gewinnstrategie im Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$, wenn $|A|, |B| \geqslant k$. Die Behauptung folgt jetzt sofort aus Korollar 3.60, indem man für jedes $k \in \mathbb{N}_{\geqslant 1}$ zwei Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ mit $|A|, |B| \geqslant k$ und $|A|$ gerade und $|B|$ ungerade betrachtet.

Ein etwas komplexeres Beispiel liefert das folgende Theorem.

3.62 Theorem (de Rougemont, 1987)**.** *Die Klasse*

$$\mathsf{Hamilton} := \left\{ G \,:\, \begin{array}{l} G \text{ ist ein endlicher gerichteter Graph, der einen} \\ \text{Hamiltonpfad besitzt} \end{array} \right\}$$

ist nicht $L^\omega_{\infty\omega}$-definierbar in der Klasse SConn *aller endlichen gerichteten stark zusammenhängenden Graphen.*

Zur Erinnerung: Ein *Hamiltonpfad* in einem Graph ist ein Pfad, der jeden Knoten des Graphen genau einmal besucht. Ein gerichteter Graph heißt *stark zusammenhängend,* falls jeder Knoten von jedem anderen Knoten aus errreichbar ist.

Beweis: Nach Korollar 3.60 reicht es, für jedes k endliche gerichtete stark zusammenhängende Graphen $\mathfrak{A}$ und $\mathfrak{B}$ zu finden, so dass $\mathfrak{A}$ aber nicht $\mathfrak{B}$ einen Hamiltonpfad besitzt und Duplicator eine Gewinnstrategie im k-Pebble-Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$ hat.

Für $m, n \in \mathbb{N}_{\geqslant 1}$ definieren wir den Graph $H_{m,n} := (V_{m,n}, E_{m,n})$ mit Knotenmenge

$$V_{m,n} := \{ v_1, \ldots, v_m,\ w_1, \ldots, w_n \}$$

und Kantenmenge

$$E_{m,n} := \{ (w_i, w_{i+1}) : 1 \leqslant i < n \} \cup \{ (w_n, w_1) \} \cup \{ (v_i, w_j), (w_j, v_i) : 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n \}.$$

Der Graph $H_{m,n}$ besteht also aus einem gerichteten Kreis der Länge n auf den Knoten $w_1, \ldots w_n$ sowie aus m weiteren Knoten $v_1, \ldots, v_m$, von denen jeder einzelne mit jedem w_j durch eine ungerichtete Kante verbunden sind. Zwischen den Knoten $v_1, \ldots, v_m$ gibt es jedoch keine direkten Kanten.

Behauptung 1: $H_{m,n}$ hat genau dann einen Hamiltonpfad, wenn $n \geqslant m - 1$.
Beweis: "$\Longleftarrow$": Sei $n \geqslant m - 1$. Dann ist

$$p := v_1, w_1, v_2, w_2, \ldots, v_{m-1}, w_{m-1}, v_m, w_m, w_{m+1}, \ldots, w_n$$

ein Hamiltonpfad in $H_{m,n}$.
"$\Longrightarrow$": Sei

$$p := u_1, u_2, \ldots, u_{m+n}$$

ein Hamiltonpfad in $H_{m,n}$. Sei $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n+m$, so dass $\{u_{i_1}, \ldots, u_{i_m}\} = \{v_1, \ldots, v_m\}$. Gemäß der Definition von $E_{m,n}$ gibt es keine Kanten zwischen Knoten aus $\{v_1, \ldots, v_m\}$. Somit muss für alle $j \in \{1, \ldots, m-1\}$ gelten: Auf dem Pfad p liegt zwischen u_{i_j} und $u_{i_{j+1}}$ mindestens ein Knoten $u_{i_j+1} \in \{w_1, \ldots, w_n\}$. Insbesondere ist daher $n \geqslant m - 1$. Damit ist Behauptung 1 bewiesen. □

Aus Behauptung 1 folgt für $k \in \mathbb{N}_{\geqslant 1}$, dass $\mathfrak{A} := H_{k+1,k}$ einen Hamiltonpfad besitzt, $\mathfrak{B} := H_{k+2,k}$ jedoch nicht.

Behauptung 2: Für jedes $k \in \mathbb{N}_{\geqslant 1}$ gilt: Duplicator hat eine Gewinnstrategie im Spiel $\mathcal{G}^k_\infty(H_{k+1,k}, H_{k+2,k})$.
Beweis: Setze $\mathfrak{A} := H_{k+1,k}$, $\mathfrak{B} := H_{k+2,k}$. Seien $A := \{v_1, \ldots, v_{k+1}, w_1, \ldots, w_k\}$ und $B := \{v'_1, \ldots, v'_{k+2}, w'_1, \ldots, w'_k\}$ die Knotenmengen von $\mathfrak{A}$ und $\mathfrak{B}$. Gemäß Korollar 3.59 reicht es zu zeigen, dass $\mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$. Dazu betrachte folgende Menge I:

$$I := \Big\{ (\vec{a} \mapsto \vec{b}) : \vec{a} = a_1, .., a_k \in (A \dot\cup \{*\})^k,\ \vec{b} = b_1, .., b_k \in (B \dot\cup \{*\})^k, \text{so dass für alle } i \in \{1, \ldots, k\} \text{ gilt:}$$
$$(1)\ a_i = * \iff b_i = *$$
$$(2)\ a_i \in \{v_1, \ldots, v_{k+1}\} \iff b_i \in \{v'_1, \ldots, v'_{k+2}\}$$
$$(3)\ \text{falls } a_i = w_j \text{ für ein } j \in \{1, \ldots, k\}, \text{ so } b_i = w'_j$$
$$(4)\ \text{für alle } i' \in \{1, \ldots, k\} \text{ gilt: } a_i = a_{i'} \iff b_i = b_{i'} \Big\}.$$

Da $(\vec{*} \mapsto \vec{*}) \in I$, ist offensichtlich $I \neq \varnothing$.
Weiterhin ist $I \subseteq \text{Part}^k(\mathfrak{A}, \mathfrak{B})$ (beachte dazu: jedes v_i ist mit jedem w_j in $\mathfrak{A}$ verbunden, und

jedes v'_i ist mit jedem w'_j in $\mathfrak{B}$ verbunden).
Schließlich kann man leicht nachweisen, dass I die Hin- und Her-Eigenschaft besitzt (Details: *Übung*).
Somit gilt also $I : \mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$. Nach Korollar 3.59 hat also Duplicator eine Gewinnstrategie in $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$. Damit ist Behauptung 2 bewiesen. □

Insgesamt haben wir nun für jedes $k \in \mathbb{N}_{\geqslant 1}$ Graphen $\mathfrak{A} := H_{k+1,k} \in \mathsf{Hamilton}$ und $\mathfrak{B} := H_{k+2,k} \in \mathsf{SConn} \setminus \mathsf{Hamilton}$ gefunden, so dass Duplicator das k-Pebble-Spiel $\mathcal{G}^k_\infty(\mathfrak{A}, \mathfrak{B})$ gewinnt. Dies schließt den Beweis von Theorem 3.62 ab. □

3.4 Interpretationen und Logische Reduktionen

3.63 Definition. Sei σ eine Signatur. Mit σ-STRUKTUREN bezeichnen wir die Klasse aller σ-Strukturen.

Ziel dieses Abschnitts ist, die Methode der *logischen Reduktionen* und *Interpretationen*, die in Bemerkung 3.21 bereits skizziert wurde, zu präzisieren. Wir wollen nun also einen Begriff für "logische Reduktionen" entwickeln, der analog ist zum Begriff der Polynomialzeit-Reduktionen.

Statt Problemen $A \subseteq \Sigma_1^*$, $B \subseteq \Sigma_2^*$ und einer Polynomialzeit-berechenbaren Reduktion $f : \Sigma_1^* \to \Sigma_2^*$ mit $w \in A \iff f(w) \in B$ geht es nun darum, eine Klasse C von τ-Strukturen, eine Klasse D von σ-Strukturen und eine "logisch definierbare" Reduktion $I : \tau$-STRUKTUREN $\to \sigma$-STRUKTUREN zu betrachten, so dass für alle τ-Strukturen $\mathfrak{A}$ gilt: $\mathfrak{A} \in \mathsf{C} \iff I(\mathfrak{A}) \in \mathsf{D}$. Analog zur Eigenschaft von Polynomialzeit-Reduktionen

> "Falls $f : A \leqslant_p B$ und $A \notin$ PTIME, so auch $B \notin$ PTIME."

soll für logische Reduktionen gelten:

> "Falls $I : \mathsf{C} \leqslant \mathsf{D}$ und C nicht FO-definierbar, so ist auch D nicht FO-definierbar."

Solche logischen Reduktionen werden durch den folgenden Begriff realisiert:

3.64 Definition (Interpretation von σ in τ)**.** Sei $\mathscr{L}$ eine Logik, σ und τ Signaturen, $\sigma = \{R_1, .., R_m\}$, wobei R_i ein r_i-stelliges Relationssymbol sei (für $1 \leqslant i \leqslant m$).[9] Sei $k \in \mathbb{N}_{\geqslant 1}$.

(a) Eine (einfache, k-dimensionale) *$\mathscr{L}$-Interpretation* I von σ in τ ist eine Sequenz

$$\big(\, \varphi_{\text{Univ}}(\vec{x}),\ \varphi_{R_1}(\vec{x}_1, .., \vec{x}_{r_1}),\ \ldots,\ \varphi_{R_m}(\vec{x}_1, .., \vec{x}_{r_m}) \,\big)$$

von $\mathscr{L}[\tau]$-Formeln, wobei $\vec{x}, \vec{x}_1, .., \vec{x}_{r_i}$ jeweils Tupel aus k verschiedenen Variablen erster Stufe sind (d.h. $\vec{x} = x_1, .., x_k$ und $\vec{x}_j = x_{j,1}, .., x_{j,k}$).

[9] Wir erlauben der Einfachheit halber in σ keine Konstantensymbole; der Begriff der Interpretation kann aber leicht modifiziert werden für Signaturen σ, die auch Konstantensymbole enthalten, indem man jedes Konstantensymbol c wie ein 1-stelliges Relationssymbol C behandelt.

(b) Eine Interpretation I von σ in τ definiert eine Abbildung

$$I \ : \ \tau\text{-STRUKTUREN} \ \rightarrow \ \sigma\text{-STRUKTUREN},$$

die jeder τ-Struktur $\mathfrak{A}$ die folgendermaßen definierte σ-Struktur $I(\mathfrak{A})$ zuordnet:

- Das *Universum* U von $I(\mathfrak{A})$ ist die Menge

$$U \ := \ \varphi_{\text{Univ}}(\mathfrak{A}) \ = \ \{\vec{a} \in A^k \ : \ \mathfrak{A} \models \varphi_{\text{Univ}}[\vec{a}]\}.$$

- Für jedes $R_i \in \sigma$ ist $R_i^{I(\mathfrak{A})}$ die r_i-stellige Relation

$$R_i^{I(\mathfrak{A})} \ := \ \varphi_{R_i}(\mathfrak{A}) \cap U^{r_i} \ = \ \{(\vec{a}_1, .., \vec{a}_{r_i}) \in U^{r_i} \ : \ \mathfrak{A} \models \varphi_{R_i}[\vec{a}_1, .., \vec{a}_{r_i}]\}.$$

(c) Sei $\mathfrak{A}$ eine τ-Struktur, $\mathfrak{B}$ eine σ-Struktur und I eine Interpretation von σ in τ. Wir sagen *I interpretiert $\mathfrak{B}$ in $\mathfrak{A}$*, falls $\mathfrak{B} \cong I(\mathfrak{A})$ (d.h. $\mathfrak{B}$ ist isomorph zu $I(\mathfrak{A})$).

3.65 Beispiele.
Sei $\tau := \{<\}$ die Signatur für lineare Ordnungen und S die Klasse aller endlichen linearen Ordnungen $\mathfrak{A} = (A, <^{\mathfrak{A}})$.

(a) Sei $\sigma_1 := \{E, S, T\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E und zwei 1-stelligen Relationssybolen S und T besteht.
Wir definieren eine FO-Interpretation I_1 von σ_1 in τ, die jeder endlichen linearen Ordnung $\mathfrak{A} = (A, <^{\mathfrak{A}})$ eine σ-Strukutur $I_1(\mathfrak{A}) = G = (V, E^G, S^G, T^G)$ zuordnet, so dass gilt:

$$|A| \text{ ist ungerade} \iff \text{im Graphen } G \text{ gibt es einen Pfad von einem Knoten in } S^G \text{ zu einem Knoten in } T^G.$$

Idee: Ist $A = \{0, .., n\}$, so $V := A$, in E^G gibt es eine Kante von Knoten i zu Knoten $i{+}2$ (für alle $i \leqslant n{-}2$), S^G besteht aus dem kleinsten Element in A und T^G aus dem größten Element in A.

Formal ist die FO-Interpretation I_1 von σ_1 in τ folgendermaßen definiert:

$$I_1 \ = \ \big(\varphi_{\text{Univ}}(x), \ \varphi_E(x,y), \ \varphi_S(x), \ \varphi_T(x)\big)$$

mit

$$\begin{aligned}
\varphi_{\text{Univ}}(x) &:= x{=}x \\
\varphi_E(x,y) &:= \exists z \Big(x < z \wedge z < y \wedge \forall u \big((x < u \wedge u < y) \rightarrow u = z \big) \Big) \\
\varphi_S(x) &:= \neg \exists y \ y < x \\
\varphi_T(x) &:= \neg \exists y \ x < y.
\end{aligned}$$

(b) Sei $\sigma_2 := \{E\}$ die Signatur für Graphen. Wir definieren eine FO-Interpretation I_2 von σ_2 in τ, die jeder endlichen linearen Ordnung $\mathfrak{A} = (A, <^{\mathfrak{A}})$ einen ungerichteten[10] Graphen $I_2(\mathfrak{A}) = G = (V, E^G)$ zuordnet, so dass gilt

$$|A| \text{ ist gerade} \iff G \text{ ist zusammenhängend.}$$

Idee: Ist $A = \{0, \ldots, n\}$, so ist $V := A$ und in G gibt es eine Kante zwischen den Knoten

(1) i und $i{+}2$, für alle $i \leqslant n{-}2$,

(2) n und 0,

(3) $n{-}1$ und 1.

Dann gilt: Ist n gerade (also $|A|$ ungerade), so zerfällt G in zwei disjunkte Kreise $C_1 = \{0, 2, 4, \ldots, n\}$ und $C_2 = \{1, 3, 5, \ldots, n{-}1\}$. Ist n ungerade (also $|A|$ gerade), so besteht G aus einem Kreis $C = \{0, 2, 4, \ldots, n{-}1, 1, 3, 5, \ldots, n\}$.

Formal ist die FO-Interpretation I_2 von σ_2 in τ folgendermaßen definiert:

$$I_2 \;=\; \big(\, \psi_{\text{Univ}}(x),\ \psi_E(x,y) \,\big)$$

mit

$$\begin{aligned}
\psi_{\text{Univ}}(x) \;&:=\; x = x \\
\psi_E(x,y) \;&:=\; \quad \varphi_E(x,y) \;\vee\; \varphi_E(y,x) && (1)\\
&\quad\ \vee \big((\text{“}x{=}\textit{min}\text{”} \wedge \text{“}y{=}\textit{max}\text{”}) \;\vee\; (\text{“}y{=}\textit{min}\text{”} \wedge \text{“}x{=}\textit{max}\text{”})\big) && (2)\\
&\quad\ \vee \big((\text{“}x{=}\textit{min}{+}1\text{”} \wedge \text{“}y{=}\textit{max}{-}1\text{”}) \;\vee\; (\text{“}y{=}\textit{min}{+}1\text{”} \wedge \text{“}x{=}\textit{max}{-}1\text{”})\big) && (3)
\end{aligned}$$

Dabei ist $\varphi_E(x,y)$ die Formel aus (a) und

$$\begin{aligned}
\text{“}x{=}\textit{min}\text{”} \;&:=\; \neg\, \exists z\ z < x\,, \\
\text{“}y{=}\textit{max}\text{”} \;&:=\; \neg\, \exists z\ y < z\,, \\
\text{“}x{=}\textit{min}{+}1\text{”} \;&:=\; \exists x' \big(\, x' < x \wedge \forall z\, (z < x \to z = x') \,\big)\,, \\
\text{“}y{=}\textit{max}{-}1\text{”} \;&:=\; \exists y' \big(\, y < y' \wedge \forall z\, (y < z \to z = y') \,\big)\,.
\end{aligned}$$

3.66 Definition ($\mathscr{L}$-Reduktion)**.** Sei $\mathscr{L}$ eine Logik und sei $k \in \mathbb{N}_{\geqslant 1}$. Seien σ und τ Signaturen mit $\sigma = \{R_1, \ldots, R_m\}$. Sei S_1 eine Klasse von τ-Strukturen und S_2 eine Klasse von σ-Strukturen und sei $\mathsf{C} \subseteq \mathsf{S}_1$ und $\mathsf{D} \subseteq \mathsf{S}_2$.
Eine (einfache, k-dimensionale) *$\mathscr{L}$-Reduktion von* $\mathsf{C} \subseteq \mathsf{S}_1$ *auf* $\mathsf{D} \subseteq \mathsf{S}_2$ ist eine (einfache, k-dimensionale) $\mathscr{L}$-Interpretation I von σ in τ, so dass für alle $\mathfrak{A} \in \mathsf{S}_1$ gilt:

[10] Ein Graph $G = (V, E)$ heißt *ungerichtet*, falls für alle $v, w \in V$ gilt: $(v, v) \notin E$ und falls $(v, w) \in E$, so auch $(w, v) \in E$.

(1.) $I(\mathfrak{A}) \in \mathsf{S}_2$ und

(2.) $\mathfrak{A} \in \mathsf{C} \iff I(\mathfrak{A}) \in \mathsf{D}$.

3.67 Beispiel. Die Interpretation I_2 aus Beispiel 3.65 (b) liefert eine 1-dimensionale FO-Reduktion von $\mathsf{Even}_< \subseteq \mathsf{S}_<$ auf $\mathsf{Conn} \subseteq \mathsf{UGraphs}$, wobei

$\mathsf{Even}_<$	die Klasse aller endlichen linearen Ordnungen gerader Länge,
$\mathsf{S}_<$	die Klasse aller endlichen linearen Ordnungen,
Conn	die Klasse aller endlichen ungerichteten zusammenhängenden Graphen,
$\mathsf{UGraphs}$	die Klasse aller endlichen ungerichteten Graphen ist.

Im Folgenden wird nun der Zusammenhang zwischen $\mathscr{L}$-Reduktionen (bzw. $\mathscr{L}$-Interpretationen) und der $\mathscr{L}$-Definierbarkeit von Problemen hergestellt.

3.68 Definition. Sei $k \in \mathbb{N}_{\geqslant 1}$, σ und τ Signaturen mit $\sigma = \{R_1, .., R_m\}$ und $\mathscr{L}$ eine der Logiken FO, SO, LFP, IFP, PFP, TC, DTC. Eine k-dimensionale $\mathscr{L}$-Interpretation

$$I \;=\; \big(\, \varphi_{\text{Univ}}(\vec{x}),\; \varphi_{R_1}(\vec{x}_1, .., \vec{x}_{r_1}),\; \ldots,\; \varphi_{R_m}(\vec{x}_1, .., \vec{x}_{r_m}) \,\big)$$

von σ in τ definiert eine Abbildung

$$\cdot^I \;:\; \mathscr{L}[\sigma] \;\to\; \mathscr{L}[\tau]\,,$$

die jeder $\mathscr{L}[\sigma]$-Formel ψ die folgendermaßen per Induktion nach dem Formelaufbau definierte $\mathscr{L}[\tau]$-Formel ψ^I zuordnet:

(A1) Ist ψ von der Form $R_i(y_1, .., y_{r_i})$ für $R_i \in \sigma$, so $\psi^I := \varphi_{R_i}(\vec{y}_1, .., \vec{y}_{r_i})$, wobei $\vec{y}_j = y_{j,1}, .., y_{j,k}$, für alle $j \in \{1, .., r_i\}$.

(A2) Ist ψ von der Form $y = z$ für $y, z \in \text{Var}_1$, so $\psi^I := \bigwedge_{j=1}^{k}(y_j = z_j)$.

(A3) Ist ψ von der Form $X(y_1, .., y_r)$, für eine r-stellige Relationsvariable $X \in \text{Var}_2$, so $\psi^I := \hat{X}(\vec{y}_1, .., \vec{y}_r)$, wobei $\hat{X}$ eine $(r{\cdot}k)$-stellige Relationsvariable und $\vec{y}_j := y_{j,1}, .., y_{j,k}$, für alle $j \in \{1, .., r\}$.

(BC) Ist ψ von der Form $\neg\psi_1$ bzw. von der Form $(\psi_1 * \psi_2)$ mit $* \in \{\wedge, \vee, \to, \leftrightarrow\}$, so ist $\psi^I := \neg\, \psi_1^I$ bzw. von der Form $(\psi_1^I * \psi_2^I)$.

(Q1) Ist ψ von der Form $\exists y\, \psi_1$ für eine Variable $y \in \text{Var}_1$, so ist

$$\psi^I \;:=\; \exists y_1 \cdots \exists y_k \,\big(\, \varphi_{\text{Univ}}(y_1, .., y_k) \;\wedge\; \psi_1^I \,\big).$$

Ist ψ von der Form $\forall y\, \psi_1$, so ist

$$\psi^I \;:=\; \forall y_1 \cdots \forall y_k \,\big(\, \varphi_{\text{Univ}}(y_1, .., y_k) \;\to\; \psi_1^I \,\big).$$

(Q2) Ist ψ von der Form $\exists X\,\psi_1$ bzw. $\forall X\,\psi_1$ für eine r-stellige Relationsvariable X, so ist, für eine $(r{\cdot}k)$-stellige Relationsvariable $\hat{X}$,

$$\psi^I \;:=\; \exists\hat{X}\Big(\forall\vec{z}_1\cdots\forall\vec{z}_r\,\big(\,\hat{X}(\vec{z}_1,..,\vec{z}_r)\to\bigwedge_{j=1}^{k}\varphi_{\text{Univ}}(\vec{z}_j)\,\big)\;\wedge\;\psi_1^I\Big)$$

bzw.

$$\psi^I \;:=\; \forall\hat{X}\Big(\forall\vec{z}_1\cdots\forall\vec{z}_r\,\big(\,\hat{X}(\vec{z}_1,..,\vec{z}_r)\to\bigwedge_{j=1}^{k}\varphi_{\text{Univ}}(\vec{z}_j)\,\big)\;\to\;\psi_1^I\Big).$$

(FP) Ist ψ von der Form $[\mathbf{fp}_{R,x_1,..,x_r}\,\psi_1(R,x_1,..,x_r)](t_1,..,t_r)$, für $\mathbf{fp}\in\{\mathbf{lfp},\mathbf{ifp},\mathbf{pfp}\}$ und eine r-stellige Relationsvariable R, so

$$\psi^I \;:=\; \Big[\mathbf{fp}_{\hat{R},\vec{x}_1,..,\vec{x}_r}\;\psi_1^I(\hat{R},\vec{x}_1,..,\vec{x}_r)\;\wedge\;\bigwedge_{j=1}^{r}\varphi_{\text{Univ}}(\vec{x}_j)\Big](\vec{t}_1,..,\vec{t}_r)\,,$$

wobei $\hat{R}$ eine $(r{\cdot}k)$-stellige Relationsvariable ist.

((D)TC) Ist ψ von der Form $[(\mathbf{d})\mathbf{tc}_{x_1,..,x_r,y_1,..,y_r}\,\psi_1](s_1,..,s_r,t_1,..,t_r)$, für $(\mathbf{d})\mathbf{tc}\in\{\mathbf{tc},\mathbf{dtc}\}$ und $r\geqslant 1$, so

$$\psi^I \;:=\; \Big[(\mathbf{d})\mathbf{tc}_{\vec{x}_1,..,\vec{x}_r,\vec{y}_1,..,\vec{y}_r}\;\psi_1^I\wedge\bigwedge_{j=1}^{r}\varphi_{\text{Univ}}(\vec{x}_j)\wedge\bigwedge_{j=1}^{r}\varphi_{\text{Univ}}(\vec{y}_j)\Big](\vec{s}_1,..,\vec{s}_r,\vec{t}_1,..,\vec{t}_r)\,.$$

Obige Definition ist gerade so gewählt, dass gilt:

3.69 Lemma. *Sei $\mathscr{L}$ eine der Logiken* FO, SO, LFP, IFP, PFP, TC, DTC. *Seien σ und τ Signaturen mit $\sigma=\{R_1,..,R_m\}$, und sei I eine $\mathscr{L}$-Interpretation von σ in τ. Dann gilt für jede τ-Struktur $\mathfrak{A}$ und jeden $\mathscr{L}[\sigma]$-Satz ψ:*

$$\mathfrak{A}\;\models\;\psi^I\quad\Longleftrightarrow\quad I(\mathfrak{A})\;\models\;\psi\,.$$

Beweis: *Übung.* □

3.70 Korollar (Reduktionslemma)**.** *Sei $\mathscr{L}$ eine der Logiken* FO, SO, LFP, IFP, PFP, TC, DTC. *Seien σ und τ Signaturen mit $\sigma=\{R_1,..,R_m\}$. Sei S_1 eine Klasse von τ-Strukturen, S_2 eine Klasse von σ-Strukturen und sei I eine $\mathscr{L}$-Reduktion von $\mathsf{C}\subseteq\mathsf{S}_1$ auf $\mathsf{D}\subseteq\mathsf{S}_2$. Dann gilt:*

(a) Falls D $\mathscr{L}$-definierbar in S_2, dann ist auch C $\mathscr{L}$-definierbar in S_1.

(b) Falls C nicht $\mathscr{L}$-definierbar in S_1, dann ist auch D nicht $\mathscr{L}$-definierbar in S_2.

Beweis: (b) ist nur eine andere (äquivalente) Formulierung der Aussage von (a). Zum Beweis von (a) sei ψ ein $\mathscr{L}[\sigma]$-Satz, der D in S_2 definiert, d.h. es gilt für alle σ-Strukturen $\mathfrak{B} \in \mathsf{S}_2$:

$$(*) \qquad \mathfrak{B} \in \mathsf{D} \iff \mathfrak{B} \models \psi.$$

Da I eine $\mathscr{L}$-Reduktion von $\mathsf{C} \subseteq \mathsf{S}_1$ auf $\mathsf{D} \subseteq \mathsf{S}_2$ ist, gilt für alle τ-Strukturen $\mathfrak{A} \in \mathsf{S}_1$, dass $I(\mathfrak{A}) \in \mathsf{S}_2$ und

$$\mathfrak{A} \in \mathsf{C} \iff I(\mathfrak{A}) \in \mathsf{D} \overset{(*)}{\iff} I(\mathfrak{A}) \models \psi \overset{\text{Lemma 3.69}}{\iff} \mathfrak{A} \models \psi^I.$$

Der $\mathscr{L}[\tau]$-Satz ψ^I ist also eine $\mathscr{L}$-Definition von C in S_1. □

Hat man gezeigt, dass ein Problem $\mathsf{C} \subseteq \mathsf{S}_1$ *nicht* $\mathscr{L}$-definierbar ist, so kann man unter Verwendung von Korollar 3.70 durch Angabe einer $\mathscr{L}$-Reduktion von $\mathsf{C} \subseteq \mathsf{S}_1$ auf $\mathsf{D} \subseteq \mathsf{S}_2$ nachweisen, dass auch das Problem $\mathsf{D} \subseteq \mathsf{S}_2$ *nicht* $\mathscr{L}$-definierbar ist.

3.71 Beispiel. Aus Beispiel 3.67 und Korollar 3.70 folgt:
Falls $\mathsf{Even}_<$ *nicht* FO-definierbar in $\mathsf{S}_<$, so ist auch Conn *nicht* FO-definierbar in $\mathsf{UGraphs}$.
D.h.: Falls man in der Logik erster Stufe nicht beschreiben kann, dass die Länge einer linearen Ordnung *gerade* ist, dann kann man auch nicht beschreiben, dass ein Graph *zusammenhängend* ist.

Abschließend zeigen wir, wie man *jede* σ-Struktur (für jede beliebige Signatur σ) durch einen *Graphen* (also eine $\{E\}$-Struktur) interpretieren kann. Dies kann man dann u.a. dazu benutzen, um zu folgern, dass der Satz von Trakhtenbrot (siehe Theorem 1.62 und Bemerkung 1.63) tatsächlich auch für die Signatur $\{E\}$ gilt, die aus nur einem Relationssymbol der Stelligkeit 2 besteht.

3.72 Satz. *Sei $\sigma = \{R_1, \ldots, R_m\}$ eine beliebige relationale Signatur und sei $\tau := \{E\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E besteht. Dann gibt es eine* FO*-Interpretation I von $\{E\}$ in σ und eine* FO*-Interpretation J von σ in $\{E\}$, so dass für alle σ-Strukturen $\mathfrak{A}$ mit $|A| \geqslant 2$ gilt:*

$$I(\mathfrak{A}) \text{ ist ein ungerichteter Graph} \quad \text{und} \quad J\big(I(\mathfrak{A})\big) \cong \mathfrak{A}.$$

Beweis: *Vorbemerkung:* In den folgenden Abbildungen zeichnen wir

um anzudeuten, dass x und y zwei verschiedene Knoten eines Graphen G sind und $(x, y) \in E^G$ und $(y, x) \in E^G$. Für eine Zahl $i \geqslant 3$ heißt ein Knoten x von G *i-etikettiert*, falls es in G einen induzierten Teilgraphen der Form

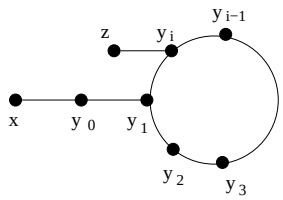


Abbildung 3.1: Das i-Etikett eines Knotens x

gibt (die Skizze bedeutet dabei, dass sämtliche Knoten verschieden sind, zwischen ihnen nur die eingezeichneten Kanten verlaufen und es von den Knoten $y_0, \ldots, y_i, z$ keine Kante zu irgendwelchen anderen Knoten des Graphen gibt).

Schritt 1: Wir zeigen zunächst, wie man jede σ-Struktur $\mathfrak{A}$ durch einen ungerichteten Graphen $G := \mathcal{I}(\mathfrak{A})$ repräsentieren kann.
Seien $r_1, \ldots, r_m \in \mathbb{N}_{\geqslant 1}$ die Stelligkeiten der Relationssymbole $R_1, \ldots, R_m$ aus σ. Der zu einer σ-Struktur $\mathfrak{A}$ gehörige Graph $G := \mathcal{I}(\mathfrak{A})$ ist folgendermaßen aufgebaut: Für jedes Element $a \in A$ gibt es in G einen 5-etikettierten Knoten v_a (d.h. für jedes einzelne Element a in A gibt es 8 Knoten in G, nämlich einen für a selbst und 7 weitere für das zugehörige 5-Etikett).
Außerdem gibt es für jedes $j \in \{1, \ldots, m\}$ und jedes r_j-Tupel $\vec{a} = (a_1, \ldots, a_{r_j}) \in R_j^{\mathfrak{A}}$ einen $5{+}j$-etikettierten Knoten $v_{j,\vec{a}}$, von dem aus es zusätzlich für jede Position i im r_j-Tupel (also für jedes $i \in \{1, \ldots, r_j\}$) einen Pfad der Länge $i{+}1$ von $v_{j,\vec{a}}$ zu v_{a_i} gibt. D.h. es gibt zusätzliche Knoten $w_{j,\vec{a},i,1}, w_{j,\vec{a},i,2}, \ldots, w_{j,\vec{a},i,i}$, so dass

$$v_{j,\vec{a}} \text{ — } w_{j,\vec{a},i,1} \text{ — } w_{j,\vec{a},i,2} \text{ — } \cdots \text{ — } w_{j,\vec{a},i,i} \text{ — } v_{a_i} \qquad (*)$$

einen Pfad in G bildet und die Knoten $w_{j,\vec{a},k,1}, w_{j,\vec{a},k,2}, \ldots, w_{j,\vec{a},k,k}$ mit keinem anderen Knoten aus G verbunden sind.
Damit sind die Knoten und Kanten des Graphen $G = \mathcal{I}(\mathfrak{A})$ vollständig beschrieben.

Schritt 2: Wir teilen nun die Knoten von $G := \mathcal{I}(\mathfrak{A})$ in verschiedene *Typen* ein, die man jeweils durch eine FO$[E]$-Formel beschreiben kann.
Dazu definieren wir die FO$[E]$-Formel

$$\begin{aligned} \rho_E(x,y) \quad &:= \quad E(x,y) \,\wedge\, E(y,x) \,\wedge\, \neg x{=}y \\ &\text{und} \\ \rho_{\neg E}(x,y) \quad &:= \quad \neg E(x,y) \,\wedge\, \neg E(y,x) \,\wedge\, \neg x{=}y\,. \end{aligned}$$

Für jedes $i \geqslant 3$ sei *Etikett*$_i(x, y_0, \ldots, y_i, z)$ die folgende FO$[E]$-Formel, die besagt, dass die

Knoten $x, y_0, \ldots, y_i, z$ ein i-Etikett für Knoten x bilden.

Etikett$_i(x, y_0, \ldots, y_i, z)$ $:=$

$$\rho_E(x,y_0) \wedge \rho_E(y_0,y_1) \wedge \bigwedge_{j=1}^{i-1} \rho_E(y_j,y_{j+1}) \wedge \rho_E(y_i,y_1) \wedge \rho_E(y_i,z) \wedge$$

$$\Big(\rho_{\neg E}(x,z) \wedge \bigwedge_{j=1}^{i} \rho_{\neg E}(x,y_j)\Big) \wedge \bigwedge_{j'=0}^{i-1} \Big(\rho_{\neg E}(y_{j'},z) \wedge \bigwedge_{j=j'+2}^{i} \rho_{\neg E}(y_{j'},y_j)\Big) \wedge$$

$$\forall u \forall v \Big(\big(u = z \vee \bigvee_{j=0}^{i} u = y_j \big) \wedge \big(v \neq x \wedge v \neq z \wedge \bigwedge_{j=0}^{i} v \neq y_j \big) \Big) \rightarrow \Big(\neg E(u,v) \wedge \neg E(v,u) \Big) .$$

Wir sagen, dass ein Knoten v *vom Typ* (i, x) ist, falls er der Knoten x eines i-Etiketts ist. Analog heißt ein Konten v *vom Typ* (i, y_j) (bzw. (i, z)), falls er der Knoten y_j (bzw. der Knoten z) eines i-Etiketts ist. Unter Verwendung der Formel *Etikett*$_i(x, y_0, \ldots, y_i, z)$ findet man für jedes $i \geqslant 3$ und jedes $u \in \{x, y_0, \ldots, y_i, z\}$ leicht eine FO$[E]$-Formel

$$\alpha_{(i,u)}(v),$$

die besagt, dass Konten v vom Typ (i, u) ist.

Knoten, die auf Pfaden aus $(*)$ liegen, folgende Typen zu: Ein Knoten v heißt *vom Typ* (j, i, k), falls er der k-te Knoten in einem Pfad der Länge $i{+}1$ ist, der von einem $(5{+}j)$-etikettierten Knoten zu einem 5-etikettierten Knoten verläuft (d.h. falls er der Knoten $w_{j,\vec{a},i,k}$ in einem Pfad $(*)$ ist). Analog zu den Formeln $\alpha_{(i,u)}(v)$ kann man leicht für alle Zahlen i, j, k eine FO$[E]$-Formel

$$\alpha_{(j,i,k)}(v)$$

konstruieren, die besagt, dass Knoten v vom Typ (j, i, k) ist.

Jeder Knoten von G ist von genau einem der folgenden Typen:

- (i, u), für ein $i \in \{5, \ldots, 5{+}m\}$ und ein $u \in \{x, y_0, \ldots, y_i, z\}$,
- (j, i, k), für ein $j \in \{1, \ldots, m\}$, $i \in \{1, \ldots, r_j\}$, $k \in \{1, \ldots, i\}$.

Schritt 3: Die (1-dimensionale) Interpretation

$$J := \Big(\psi_{\text{Univ}}(v), \big(\psi_{R_j}(v_1, \ldots, v_{r_j})\big)_{j=1,\ldots,m} \Big)$$

von σ in $\{E\}$ ist folgendermaßen definiert:

$$\begin{aligned} \psi_{\text{Univ}}(v) &:= \alpha_{(5,x)}(v) \\ \psi_{R_j}(v_1, \ldots, v_{r_j}) &:= \exists u \Big(\alpha_{(5+j,x)}(u) \wedge \bigwedge_{i=1}^{r_j} \text{“es gibt einen Pfad der Länge } i{+}1 \text{ von } u \text{ nach } v_i\text{”} \Big) \end{aligned}$$

J ist so definiert, dass für alle σ-Strukturen $\mathfrak{A}$ gilt:

$(**)$ Ist $G := \mathcal{I}(\mathfrak{A})$, so ist $J(G) \cong \mathfrak{A}$.

Schritt 4: Wir zeigen nun noch, dass die Abbildung $\mathcal{I}$ durch eine FO$[\sigma]$-Interpretation

$$I := \Big(\varphi_{\text{Univ}}(\vec{v}),\ \varphi_E(\vec{v},\vec{w})\Big)$$

realisiert werden kann.

Sei dazu $r := max\{r_1,..,r_m\}$ die maximale Stelligkeit eines Relationssymbols in σ und sei $p \in \mathbb{N}$ die Anzahl der am Ende von Schritt 2 genannten möglichen Typen (i,u) und (j,i,k), und sei $t_1,..,t_p$ eine Auflistung all dieser Typen.
Jeder Knoten v von $G = \mathcal{I}(\mathfrak{A})$ kann durch ein $(r+p+1)$-Tupel

$$(\vec{a},\vec{b},c) \ \in \ A^{r+p+1}$$

wie folgt repräsentiert werden: $\vec{a} = a_1,..,a_r$ gibt das Element $a_1 \in A$ oder das Tupel $(a_1,..,a_{r_j}) \in R_j^{\mathfrak{A}}$ an, für das der Knoten v geschaffen wurde. Die Sequenz $\vec{b},c = b_1,..,b_p,c$ gibt wie folgt an, von welchem Typ der Knoten v ist:

$$v \text{ ist vom Typ } t_j \iff c = b_j \text{ (und } c \neq b_i \text{ für alle } i \neq j)$$

(genau dafür brauchen wir die Voraussetzung, dass $|A| \geqslant 2$ ist). Man beachte, dass es bei dieser Repräsentation für jeden Knoten v *mehrere* Tupel $(\vec{a},\vec{b},c)$ geben kann, die v repräsentieren. Es ist nicht schwierig (aber einigermaßen aufwendig), FO$[\sigma]$-Formeln $\varphi_{\text{Univ}}(\vec{x})$ und $\varphi_E(\vec{x},\vec{y})$ anzugeben, die besagen, dass $\vec{x}$ ein Repräsentant eines Knotens von $G := \mathcal{I}(\mathfrak{A})$ ist bzw. dass $\vec{x}$ und $\vec{y}$ Repräsentanten von Knoten sind, zwischen denen in G eine Kante verläuft. Insgesamt erhalten wir dadurch eine $(r+p+1)$-dimensionale Interpretation I von $\{E\}$ in σ, so dass für alle σ-Strukturen $\mathfrak{A}$ gilt: $\mathfrak{A} \cong J\big(I(\mathfrak{A})\big)$. □

4 Der Satz von Gaifman

In diesem Kapitel werden wir den Satz von Gaifman beweisen, der ein tieferes Verständnis dafür liefert, welche Aussagen durch Formeln der Logik erster Stufe getroffen werden können. In gewisser Weise besagt der Satz von Gaifman, dass die Logik erster Stufe nur "lokale" Eigenschaften von Strukturen definieren kann.

Der Einfachheit halber werden wir in diesem Kapitel nur *relationale Signaturen* betrachten, d.h. Signaturen, die ausschließlich aus Relationssymbolen bestehen.

4.1 Formulierung und Beweis des Satzes von Gaifman

Bevor wir die exakte Formulierung des Satzes von Gaifman angeben können, benötigen wir zunächst noch einige Notationen und Begriffe.

Zur Erinnerung: In Definition 3.22 wurden bereits der *Gaifman-Graph* $\mathcal{G}(\mathfrak{A})$ einer Struktur $\mathfrak{A}$ eingeführt sowie die *Distanzfunktion* $\text{Dist}^{\mathfrak{A}}(a,b)$, die die Distanz zwischen zwei Knoten a und b im Gaifman-Graph $\mathcal{G}(\mathfrak{A})$ bezeichnet.

Außerdem wurden die r*-Umgebung* (oder r*-Nachbarschaft*)

$$N_r^{\mathfrak{A}}(a) \;:=\; \{b \in A \,:\, \text{Dist}^{\mathfrak{A}}(a,b) \leqslant r\}$$

sowie die durch die r-Nachbarschaft induzierte Substruktur

$$\mathcal{N}_r^{\mathfrak{A}}(a) \;:=\; \Big(N_r^{\mathfrak{A}}(a),\; \big(R^{\mathfrak{A}} \cap N_r^{\mathfrak{A}}(a)^{ar(R)}\big)_{R\in\sigma}\Big)$$

eingeführt. Wir verallgemeinern diese Begriffe nun auf die naheliegende Weise auch für Tupel $\vec{a} = a_1, \ldots, a_k$.

4.1 Definition.
Sei σ eine relationale Signatur, $\mathfrak{A}$ eine σ-Struktur, $k \in \mathbb{N}_{\geqslant 1}$, $\vec{a} = a_1, \ldots, a_k \in A$ und $r \in \mathbb{N}$.

(a) Für jedes $b \in A$ ist $\text{Dist}^{\mathfrak{A}}(\vec{a}, b) := \min\{\text{Dist}^{\mathfrak{A}}(a_i, b) \,:\, 1 \leqslant i \leqslant k\}$.

(b) $N_r^{\mathfrak{A}}(\vec{a}) \;:=\; \bigcup_{i=1}^{k} N_r^{\mathfrak{A}}(a_i) \;=\; \{b \in A \,:\, \text{Dist}^{\mathfrak{A}}(\vec{a}, b) \leqslant r\}$.

(c) $\mathcal{N}_r^{\mathfrak{A}}(\vec{a}) \;:=\; \Big(N_r^{\mathfrak{A}}(\vec{a}),\; \big(R^{\mathfrak{A}} \cap N_r^{\mathfrak{A}}(\vec{a})^{ar(R)}\big)_{R\in\sigma}\Big)$.

Für jede Zahl r kann man leicht eine FO-Formel finden, die ausdrückt, dass die Distanz zwischen zwei Knoten höchstens r ist:

4.2 Lemma. *Sei σ eine relationale Signatur und sei $r \in \mathbb{N}$. Dann gibt es eine* FO$[\sigma]$*-Formel $\mathit{dist}_{\leqslant r}(x,y)$ so dass für alle σ-Strukturen $\mathfrak{A}$ und alle $a, b \in A$ gilt:*

$$\mathfrak{A} \models \mathit{dist}_{\leqslant r}[a,b] \iff \mathit{Dist}^{\mathfrak{A}}(a,b) \leqslant r.$$

Analog gibt es für jede Zahl k und das Variablentupel $\vec{x} = x_1, .., x_k$ eine FO$[\sigma]$*-Formel $\mathit{dist}_{\leqslant r}(\vec{x}, y)$ so dass für alle σ-Strukturen $\mathfrak{A}$, alle $\vec{a} = a_1, .., a_k \in A$ und alle $b \in A$ gilt: $\mathfrak{A} \models \mathit{dist}_{\leqslant r}[\vec{a}, b] \iff \mathit{Dist}^{\mathfrak{A}}(\vec{a}, b) \leqslant r$.*

Beweis: Per Induktion nach r.
Für den Induktionsanfang setzen wir $\mathit{dist}_{\leqslant 0}(x,y) := (x = y)$ und

$$\mathit{dist}_{\leqslant 1}(x,y) := \quad x = y \ \vee \bigvee_{R \in \sigma} \exists u_1 \cdots \exists u_{ar(R)} \Big(R(u_1, .., u_{ar(R)}) \ \wedge \bigvee_{1 \leqslant i,j \leqslant ar(R)} \big(u_i = x \ \wedge \ u_j = y \big) \Big) \Big).$$

Für den Induktionsschritt von r nach $r+1$ setzen wir

$$\mathit{dist}_{\leqslant r+1}(x,y) := \quad \mathit{dist}_{\leqslant r}(x,y) \ \vee \ \exists z \Big(\mathit{dist}_{\leqslant r}(x,z) \ \wedge \mathit{dist}_{\leqslant 1}(z,y) \Big).$$

Für $\vec{x} = x_1, .., x_k$ setzen wir schließlich $\mathit{dist}_{\leqslant r}(\vec{x}, y) := \bigvee_{i=1}^{k} \mathit{dist}_{\leqslant r}(x_i, y)$. □

Zur besseren Lesbarkeit von Formeln schreiben wir im Folgenden oft

$$\mathit{dist}(\vec{x}, y) \leqslant r \qquad \text{bzw.} \qquad \mathit{dist}(\vec{x}, y) > r$$

and Stelle von $\mathit{dist}_{\leqslant r}(\vec{x}, y)$ bzw. $\neg \mathit{dist}_{\leqslant r}(\vec{x}, y)$.

Als nächstes führen wir einige Begriffe zur *Lokalität* von Formeln ein. Informell gesprochen ist eine Formel $\varphi(\vec{x})$ genau dann lokal, wenn sie nur über eine r-Nachbarschaft von $\vec{x}$ "spricht".

4.3 Definition (lokale Formeln)**.**
Sei σ eine relationale Signatur und sei $\psi(\vec{x})$ eine FO$[\sigma]$-Formel mit $\mathit{frei}(\psi) = \{\vec{x}\} \neq \varnothing$.

(a) Sei $r \in \mathbb{N}$. Die Formel ψ heißt *r-lokal um $\vec{x}$*, falls für alle σ-Strukturen $\mathfrak{A}$ und alle $\vec{a} \in A$ gilt: $\mathfrak{A} \models \psi[\vec{a}] \iff \mathcal{N}_r^{\mathfrak{A}}(\vec{a}) \models \psi[\vec{a}]$.

(b) Die Formel ψ heißt *lokal um $\vec{x}$*, falls es eine Zahl $r \in \mathbb{N}$ gibt, so dass ψ r-lokal um $\vec{x}$ ist.

4.4 Beispiel.
Für jedes $r \in \mathbb{N}$ ist die Formel $\mathit{dist}(x,y) > 2{\cdot}r$ aus Lemma 4.2 r-lokal um x, y.

Man sieht leicht, dass eine Formel, die r-lokal um $\vec{x}$ ist, auch r'-lokal um $\vec{x}$ ist, für jedes $r' \geqslant r$.

Eine Möglichkeit, r-Lokalität zu erzwingen, besteht natürlich darin, explizit sämtliche Quantoren einer Formel auf die r-Nachbarschaft von $\vec{x}$ einzuschränken. Dies wird in der folgenden Definition formalisiert.

4.5 Definition (r-Relativierung)**.** Sei σ eine relationale Signatur, sei $\psi(\vec{x})$ eine FO$[\sigma]$-Formel mit $\varnothing \neq \{\vec{x}\} = \mathit{frei}(\psi)$, und sei $r \in \mathbb{N}$. Die *r-Relativierung von ψ um $\vec{x}$* ist die Formel

$$\psi^{N_r(\vec{x})}(\vec{x}),$$

die aus $\psi(\vec{x})$ entsteht, indem zunächst alle gebundenen Variablen so umbenannt werden, dass sie verschieden von $\vec{x}$ sind und danach jede Teilformel der Form

- $\exists z\ \varphi$ durch $\exists z \big(\mathit{dist}(\vec{x}, z) \leqslant r\ \wedge\ \varphi\big)$
- $\forall z\ \varphi$ durch $\forall z \big(\mathit{dist}(\vec{x}, z) \leqslant r\ \rightarrow\ \varphi\big)$

ersetzt wird.

Offensichtlich gilt:

4.6 Lemma ("r-Relativierungen sind r-lokal")**.** *Sei σ eine relationale Signatur, sei $\psi(\vec{x})$ eine* FO$[\sigma]$*-Formel mit frei$(\psi) = \{\vec{x}\} \neq \varnothing$, und sei $r \in \mathbb{N}$. Dann gilt: Die r-Relativierung $\psi^{N_r(\vec{x})}(\vec{x})$ ist r-lokal um $\vec{x}$, und es gilt für alle σ-Strukturen $\mathfrak{A}$ und alle $\vec{a} \in A$, dass*

$$\mathfrak{A} \models \psi^{N_r(\vec{x})}[\vec{a}] \quad \Longleftrightarrow \quad \mathcal{N}_r^{\mathfrak{A}}(\vec{a}) \models \psi^{N_r(\vec{x})}[\vec{a}] \quad \Longleftrightarrow \quad \mathcal{N}_r^{\mathfrak{A}}(\vec{a}) \models \psi[\vec{a}].$$

Die obige Definition 4.3 von *lokalen Formeln* dreht sich nur um Formeln, die freie Variablen besitzen. Als nächstes führen wir auch einen Lokalitäts-Begriff für *Sätze* ein, d.h. für Formeln, die keine freien Variablen besitzen.

4.7 Definition (basis-lokale Sätze)**.** Sei σ eine relationale Signatur und seien $\ell, r \in \mathbb{N}$. Ein FO$[\sigma]$-Satz χ heißt *basis-lokal* (mit Parametern ℓ, r), falls χ von der Form

$$\exists x_1 \cdots \exists x_\ell \Big(\Big(\bigwedge_{1 \leqslant i < j \leqslant \ell} \mathit{dist}(x_i, x_j) > 2{\cdot}r \Big) \ \wedge\ \Big(\bigwedge_{i=1}^{\ell} \psi(x_i) \Big) \Big)$$

ist, wobei $\psi(x)$ eine FO$[\sigma]$-Formel ist, die r-lokal um x ist.

Ein basis-lokaler Satz χ besagt also, dass es mindestens ℓ Elemente $x_1, \ldots, x_\ell$ gibt, deren r-Nachbarschaften paarweise disjunkt sind und sämtlich die Formel ψ erfüllen.

4.8 Definition (Gaifman-Normalform)**.** Sei σ eine relationale Signatur.

(a) Ein FO[σ]-Satz φ ist in *Gaifman-Normalform*, falls φ eine Boolesche Kombination[1] von basis-lokalen FO[σ]-Sätzen ist.

(b) Eine FO[σ]-Formel $\varphi(\vec{x})$ mit *frei*$(\varphi) = \{\vec{x}\} \neq \varnothing$ ist in *Gaifman-Normalform*, falls $\varphi(\vec{x})$ eine Boolesche Kombination von basis-lokalen FO[σ]-Sätzen ist und von FO[σ]-Formeln, die lokal um $\vec{x}$ sind.

Wir können nun den Satz von Gaifman angeben und beweisen:

4.9 Theorem (Satz von Gaifman, 1981)**.** *Sei σ eine relationale Signatur.*
Jede FO[σ]*-Formel ist äquivalent*[2] *zu einer* FO[σ]*-Formel in Gaifman-Normalform.*

Außerdem gibt es einen Algorithmus, der bei Eingabe einer FO[σ]*-Formel eine äquivalente* FO[σ]*-Formel in Gaifman-Normalform berechnet.*

Beweis: Wir geben hier den Originalbeweis von Gaifman an, der per Induktion nach dem Aufbau von FO[σ]-Formeln vorgeht.

Induktionsanfang: φ ist eine atomare σ-Formel, d.h. von der Form $x_1 = x_2$ oder von der Form $R(x_1, .., x_k)$ mit $k := ar(R)$.
Offensichtlich ist φ dann 0-lokal um $\vec{x}$ und daher insbesondere in Gaifman-Normalform.

Induktionsschritt:

Fall 1: φ ist von der Form $\neg\varphi'$.
Gemäß Induktionsannahme ist φ' äquivalent zu einer Formel $\tilde{\varphi}'$ in Gaifman-Normalform. Offensichtlich ist dann $\neg\tilde{\varphi}'$ eine zu φ äquivalente Formel in Gaifman-Normalform.

Fall 2: φ ist von der Form $(\varphi_1 * \varphi_2)$ für ein $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
Analog zu Fall 1.

Fall 3: φ ist von der Form $\forall y\, \varphi'$.
Dann ist φ äquivalent zur Formel $\neg\exists y\, \neg\varphi'$. Daher kann Fall 3 durch eine Kombination von Fall 1 und dem folgenden Fall 4 gelöst werden.

Fall 4: φ ist von der Form $\exists y\, \varphi'$.
Gemäß Induktionsannahme ist φ' äquivalent zu einer Formel $\tilde{\varphi}'$ in Gaifman-Normalform. Gemäß Definition 4.8 ist $\tilde{\varphi}'$ eine Boolesche Kombination von basis-lokalen Sätzen und von Formeln, die lokal um $\vec{x}, y$ sind, wobei $\vec{x}, y$ die freien Variablen in φ' bezeichnen.

O.B.d.A. ist diese Boolesche Kombination in *disjunktiver Normalform*, so dass $\tilde{\varphi}'$ von der Form

$$\bigvee_{i \in I} \Big(\chi_i \ \wedge \ \psi_i(\vec{x}, y) \Big)$$

[1] Ist M eine Formelmenge, so ist die Menge BC(M) aller *Booleschen Kombinationen* von Formeln aus M die kleinste Menge mit $M \subseteq$ BC(M), für die gilt: Ist $\varphi \in$ BC(M), so ist auch $\neg\varphi \in$ BC(M); und sind $\varphi_1, \varphi_2 \in$ BC(M) und ist $*$ eins der Symbole $\wedge, \vee, \rightarrow, \leftrightarrow$, so ist auch $(\varphi_1 * \varphi_2) \in$ BC(M).

[2] auf der Klasse aller σ-Strukturen

ist, wobei I eine geeignete endliche Indexmenge und, für jedes $i \in I$, χ_i eine Boolesche Kombination von basis-lokalen Sätzen, und $\psi_i(\vec{x}, y)$ eine lokale Formel um $\vec{x}, y$ ist (beachte dazu: Boolesche Kombinationen von lokalen Formeln sind selbst wieder lokal).

Wir werden im Folgenden oft das Symbol $\equiv$ benutzen, um auszudrücken, dass zwei Formeln äquivalent (über der Klasse aller σ-Strukturen) sind.

Insbesondere gilt für die Formel $\varphi := \exists y\, \varphi'$, dass

$$\varphi \;\equiv\; \exists y \bigvee_{i\in I} \Big(\chi_i \wedge \psi_i(\vec{x}, y)\Big) \;\equiv\; \bigvee_{i\in I} \exists y \,\Big(\chi_i \wedge \psi_i(\vec{x}, y)\Big) \;\equiv\; \bigvee_{i\in I} \Big(\chi_i \;\wedge\; \exists y\, \psi_i(\vec{x}, y)\Big)$$

(die letzte Äquivalenz gilt, da χ_i ein *Satz* ist und daher insbesondere $y \notin \mathit{frei}(\chi_i)$ ist).

Zum Abschluss von Fall 4 genügt es daher, im Folgenden zu zeigen, dass für jedes $i \in I$ die Formel $\exists y\, \psi_i(\vec{x}, y)$ äquivalent zu einer Formel in Gaifman-Normalform ist. Sei also $i \in I$ beliebig. Wir schreiben im Folgenden kurz $\psi(\vec{x}, y)$, um die Formel $\psi_i(\vec{x}, y)$ zu bezeichnen. Wir wissen bereits, dass $\psi(\vec{x}, y)$ lokal um $\vec{x}, y$ ist, d.h. es gibt eine Zahl $r \in \mathbb{N}$, so dass $\psi(\vec{x}, y)$ r-lokal um $\vec{x}, y$ ist.

Falls $\vec{x}$ das *leere* Tupel ist, so ist $\psi(\vec{x}, y)$ einfach die Formel $\psi(y)$, die r-lokal um y ist. Gemäß Definition 4.7 ist die Formel $\exists y\, \psi(y)$ daher ein basis-lokaler Satz und damit insbesondere in Gaifman-Normalform.

Falls $\vec{x}$ ein *nicht-leeres* Tupel ist, so gilt offensichtlicherweise: $\exists y\, \psi(\vec{x}, y) \;\equiv$

$$\underbrace{\exists y \Big(\mathit{dist}(\vec{x}, y) \leqslant 2r{+}1 \;\wedge\; \psi(\vec{x}, y)\Big)}_{=:\;\vartheta_1(\vec{x})} \;\vee\; \underbrace{\exists y \Big(\mathit{dist}(\vec{x}, y) > 2r{+}1 \;\wedge\; \psi(\vec{x}, y)\Big)}_{=:\;\vartheta_2(\vec{x})}.$$

Zum Abschluss von Fall 4 genügt es daher, im Folgenden zu zeigen, dass jede der beiden Formeln $\vartheta_1(\vec{x})$ und $\vartheta_2(\vec{x})$ äquivalent zu einer Formel in Gaifman-Normalform ist.

Wir betrachten zunächst die Formel $\vartheta_1(\vec{x})$ und zeigen, dass $\vartheta_1(\vec{x})$ lokal um $\vec{x}$ ist:
Sei dazu $\mathfrak{A}$ eine beliebige σ-Struktur und sei $\vec{a} \in A$ eine Belegung für $\vec{x}$. Da die Formel $\psi(\vec{x}, y)$ r-lokal um $\vec{x}, y$ ist, wissen wir, dass folgendes gilt:

$$\mathfrak{A} \models \vartheta_1[\vec{a}] \quad\Longleftrightarrow\quad \text{es gibt ein } b \in N^{\mathfrak{A}}_{2r+1}(\vec{a}), \text{ so dass } \mathcal{N}^{\mathfrak{A}}_r(\vec{a}, b) \models \psi[\vec{a}, b].$$

Ferner wissen wir, dass $N^{\mathfrak{A}}_r(\vec{a}, b) = N^{\mathfrak{A}}_r(\vec{a}) \cup N^{\mathfrak{A}}_r(b)$.

Für $b \in N^{\mathfrak{A}}_{2r+1}(\vec{a})$ ist außerdem $N^{\mathfrak{A}}_r(b) \subseteq N^{\mathfrak{A}}_{3r+1}(\vec{a})$, und daher ist $N^{\mathfrak{A}}_r(\vec{a}, b) \subseteq N^{\mathfrak{A}}_{3r+1}(\vec{a})$.

Daraus folgt unmittelbar, dass die Formel $\vartheta_1(\vec{x})$ $(3r{+}1)$-lokal um $\vec{x}$ ist. Insbesondere ist $\vartheta_1(\vec{x})$ also in Gaifman-Normalform.

Wir betrachten nun die Formel $\vartheta_2(\vec{x})$:
Sei dazu wieder $\mathfrak{A}$ eine beliebige σ-Struktur und sei $\vec{a} \in A$ eine Belegung für $\vec{x}$. Für jedes $b \in A$ mit $\mathrm{Dist}^{\mathfrak{A}}(\vec{a}, b) > 2r{+}1$ gilt insbesondere:

$N^{\mathfrak{A}}_r(\vec{a}) \cap N^{\mathfrak{A}}_r(b) = \varnothing$ und für alle $u \in N^{\mathfrak{A}}_r(\vec{a})$ und $v \in N^{\mathfrak{A}}_r(b)$ ist $\mathrm{Dist}^{\mathfrak{A}}(u, v) > 1$. Daher ist die Struktur $\mathcal{N}^{\mathfrak{A}}_r(\vec{a}, b)$ die disjunkte Vereinigung der Strukturen $\mathcal{N}^{\mathfrak{A}}_r(\vec{a})$ und $\mathcal{N}^{\mathfrak{A}}_r(b)$, kurz: $\mathcal{N}^{\mathfrak{A}}_r(\vec{a}, b) = \mathcal{N}^{\mathfrak{A}}_r(\vec{a}) \sqcup \mathcal{N}^{\mathfrak{A}}_r(b)$.

Da $\psi(\vec{x}, y)$ r-lokal um $\vec{x}, y$ ist, hängt die Gültigkeit von $\psi[\vec{a}, b]$ in $\mathfrak{A}$ nur von $\mathcal{N}_r^{\mathfrak{A}}(\vec{a}) \sqcup \mathcal{N}_r^{\mathfrak{A}}(b)$ ab.

Eingeschränkt auf σ-Strukturen $\mathfrak{A}$ mit Belegungen $\vec{a}$ und b so dass $\text{Dist}^{\mathfrak{A}}(\vec{a}, b) > 2r+1$ ist, ist daher die Formel $\psi(\vec{x}, y)$ äquivalent zu einer Booleschen Kombination von

(i) Formeln, die r-lokal um $\vec{x}$ sind, und

(ii) Formeln, die r-lokal um y sind.

O.B.d.A. ist diese Boolesche Kombination in *disjunktiver Normalform*, so dass $\vartheta_2(\vec{x})$ äquivalent (auf der Klasse aller σ-Strukturen) ist zu einer Formel der Form

$$\exists y \Big(\mathit{dist}(\vec{x}, y) > 2r+1 \ \wedge \bigvee_{j \in J} \big(\gamma_j(\vec{x}) \ \wedge \ \delta_j(y) \big) \Big), \tag{4.1}$$

wobei J eine geeignete endliche Indexmenge und, für jedes $j \in J$, $\gamma_j(\vec{x})$ eine r-lokale Formel um $\vec{x}$ und $\delta_j(y)$ eine r-lokale Formel um y ist.

Offensichtlich ist die Formel aus (4.1) äquivalent zu

$$\bigvee_{j \in J} \ \exists y \Big(\ \mathit{dist}(\vec{x}, y) > 2r+1 \ \ \wedge \ \ \gamma_j(\vec{x}) \ \ \wedge \ \ \delta_j(y) \ \Big). \tag{4.2}$$

Da die Variable y nicht frei in der Formel $\gamma_j(\vec{x})$ vorkommt, ist die Formel aus (4.2) wiederum äquivalent zu

$$\bigvee_{j \in J} \ \Big(\ \gamma_j(\vec{x}) \ \ \wedge \ \ \exists y \ \big(\ \mathit{dist}(\vec{x}, y) > 2r+1 \ \ \wedge \ \ \delta_j(y) \ \big) \ \Big).$$

Wir wissen bereits, dass jede der Formeln $\gamma_j(\vec{x})$ lokal um $\vec{x}$ und damit insbesondere in Gaifman-Normalform ist. Zum Nachweis, dass die Formel $\vartheta_2(\vec{x})$ äquivalent zu einer Formel in Gaifman-Normalform ist, genügt es daher, im Folgenden zu zeigen, dass für jedes $j \in J$ die Formel $\exists y \big(\mathit{dist}(\vec{x}, y) > 2r+1 \wedge \delta_j(y) \big)$ äquivalent zu einer Formel in Gaifman-Normalform ist. Sei also $j \in J$ beliebig. Wir schreiben im Folgenden kurz $\delta(y)$, um die Formel $\delta_j(y)$ zu bezeichnen, und wir setzen

$$\mu(\vec{x}) \ \ := \ \ \exists y \big(\mathit{dist}(\vec{x}, y) > 2r+1 \wedge \delta(y) \big).$$

Wir wissen bereits, dass $\delta(y)$ r-lokal um y ist.

Um zeigen zu können, dass $\mu(\vec{x})$ äquivalent zu einer Formel in Gaifman-Normalform ist, werden wir nun zunächst Formeln konstruieren, die "zählen", wie viele paarweise weit voneinander entfernte Elemente es gibt, deren r-Nachbarschaften die Formel $\delta(y)$ erfüllen.

Sei dazu $k \geqslant 1$ die Länge des Tupels $\vec{x}$, d.h. $\vec{x} = x_1, .., x_k$. Für jedes $\ell \in \{1, .., k+1\}$ sei χ_ℓ der basis-lokale Satz

$$\chi_\ell \ \ := \ \ \exists z_1 \cdots \exists z_\ell \ \Big(\ \underbrace{\Big(\bigwedge_{1 \leqslant i < j \leqslant \ell} \mathit{dist}(z_i, z_j) > 2 \cdot (2r+1) \Big) \ \wedge \ \Big(\bigwedge_{i=1}^{\ell} \delta(z_i) \Big)}_{=: \ \alpha_\ell(z_1, .., z_\ell)} \ \Big), \tag{4.3}$$

der besagt, dass es mindestens ℓ paarweise weit voneinander entfernte Elemente gibt, deren r-Umgebungen allesamt die Formel δ erfüllen.
Offensichtlich ist $\exists y\,\delta(y)$ äquivalent zu χ_1 und auch äquivalent zu

$$(\chi_1 \wedge \neg\chi_2) \vee (\chi_2 \wedge \neg\chi_3) \vee \cdots \vee (\chi_k \wedge \neg\chi_{k+1}) \vee \chi_{k+1}\,.$$

Daher gilt auch: $\mu(\vec{x}) \equiv \big(\mu(\vec{x}) \wedge \exists y\,\delta(y)\big) \equiv$

$$(\mu(\vec{x}) \wedge \chi_1 \wedge \neg\chi_2) \vee (\mu(\vec{x}) \wedge \chi_2 \wedge \neg\chi_3) \vee \cdots \vee (\mu(\vec{x}) \wedge \chi_k \wedge \neg\chi_{k+1}) \vee (\mu(\vec{x}) \wedge \chi_{k+1})\,.$$

Zum Abschluss des Beweises von Theorem 4.9 genügt es daher, im Folgenden zu zeigen, dass die Formel $(\mu(\vec{x}) \wedge \chi_{k+1})$ sowie die Formel $(\mu(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1})$, für jedes $\ell \in \{1,\ldots,k\}$, äquivalent zu einer Formel in Gaifman-Normalform ist.

Wir betrachten zunächst die Formel $(\mu(\vec{x}) \wedge \chi_{k+1})$ und zeigen, dass diese Formel äquivalent ist zur Formel χ_{k+1}.
Sei dazu $\mathfrak{A}$ eine beliebige σ-Struktur und sei $\vec{a} = a_1,\ldots,a_k \in A$ eine Belegung für $\vec{x} = x_1,\ldots,x_k$. Falls $\mathfrak{A} \models (\mu[\vec{a}] \wedge \chi_{k+1})$, so gilt natürlich insbesondere auch $\mathfrak{A} \models \chi_{k+1}$.

Umgekehrt, falls $\mathfrak{A} \models \chi_{k+1}$, so gibt es (mindestens) $k{+}1$ Elemente $c_1,\ldots,c_{k+1}$ in A, die paarweise Abstand $> 2{\cdot}(2r{+}1)$ voneinander haben, so dass $\mathcal{N}_r^{\mathfrak{A}}(c_i) \models \delta[c_i]$ für alle $i \in \{1,\ldots,k{+}1\}$.
Da das Tupel $\vec{a} = a_1,\ldots,a_k$ aus höchstens k verschiedenen Elementen besteht, muss es ein $i \in \{1,\ldots,k{+}1\}$ geben, so dass $\mathrm{Dist}^{\mathfrak{A}}(\vec{a},c_i) > 2r{+}1$ ist, denn: Angenommen nicht, dann gibt es ein $j \in \{1,\ldots,k\}$ und zwei Indices i,i' mit $1 \leqslant i < i' \leqslant k{+}1$, so dass $\mathrm{Dist}^{\mathfrak{A}}(a_j,c_i) \leqslant 2r{+}1$ und $\mathrm{Dist}^{\mathfrak{A}}(a_j,c_{i'}) \leqslant 2r{+}1$. Dann ist aber $\mathrm{Dist}^{\mathfrak{A}}(c_i,c_{i'}) \leqslant 2{\cdot}(2r{+}1)$, was im Widerspruch zur Wahl der Elemente $c_1,\ldots,c_{k+1}$ steht.
Somit gibt es ein $i \in \{1,\ldots,k{+}1\}$, so dass $\mathrm{Dist}^{\mathfrak{A}}(\vec{a},c_i) > 2r{+}1$. Dieses c_i bezeugt, dass $\mathfrak{A}$ die Formel $\mu[\vec{a}]$ erfüllt. Somit gilt $\mathfrak{A} \models (\mu[\vec{a}] \wedge \chi_{k+1})$.

Insgesamt haben wir also gezeigt, dass $(\mu(\vec{x}) \wedge \chi_{k+1})$ äquivalent zum basis-lokalen Satz χ_{k+1} ist, der insbesondere in Gaifman-Normalform ist.

Wir betrachten nun die Formel $(\mu(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1})$ für ein $\ell \in \{1,\ldots,k\}$.
Sei dazu $\alpha_\ell(z_1,\ldots,z_\ell)$ die Formel aus (4.3), und sei

$$\kappa_\ell(\vec{x}) \;:=\; \exists z_1 \cdots \exists z_\ell \Big(\big(\bigwedge_{i=1}^{\ell} \mathit{dist}(\vec{x},z_i) \leqslant 2r{+}1 \big) \wedge \alpha_\ell(z_1,\ldots,z_\ell) \Big)$$

eine Formel, die besagt, dass es in der $2r{+}1$-Nachbarschaft von $\vec{x}$ mindestens ℓ verschiedene Elemente vom paarweisen Abstand $> 2{\cdot}(2r{+}1)$ gibt, deren r-Nachbarschaften allesamt die Formel δ erfüllen. Man sieht leicht, dass die Formel

$$\kappa_\ell(\vec{x}) \quad 2{\cdot}(2r{+}1)\text{-lokal um } \vec{x} \text{ ist}$$

(beachte dazu: die Teilformel $\mathit{dist}(z_i,z_j) > 2{\cdot}(2r{+}1)$ in $\alpha_\ell(z_1,\ldots,z_\ell)$ ist laut Beispiel 4.4 $(2r{+}1)$-lokal um z_i,z_j).

Außerdem ist natürlich die Formel $(\mu(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1})$ äquivalent zur Formel

$$\underbrace{\Big(\kappa_\ell(\vec{x}) \wedge \mu(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1}\Big)}_{=:\text{ (I)}} \vee \underbrace{\Big(\neg\kappa_\ell(\vec{x}) \wedge \mu(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1}\Big)}_{=:\text{ (II)}}$$

Wir schauen uns zunächst die Formel (II) an und zeigen, dass diese äquivalent zu einer Formel in Gaifman-Normalform ist:

$\chi_\ell \wedge \neg\kappa_\ell(\vec{x})$ impliziert natürlich, dass es mindestens ein Element außerhalb der $2r{+}1$-Umgebung von $\vec{x}$ geben muss, dessen r-Nachbarschaft die Formel δ erfüllt. Somit impliziert $\chi_\ell \wedge \neg\kappa_\ell(\vec{x})$, dass auch $\mu(\vec{x})$ gilt. Daher ist die Formel (II) äquivalent zur Formel $(\neg\kappa_\ell(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1})$, die offensichtlich in Gaifman-Normalform ist.

Wir zeigen nun noch, dass auch die Formel (I) äquivalent zu einer Formel in Gaifman-Normalform ist:

$\kappa_\ell(\vec{x}) \wedge \neg\chi_{\ell+1}$ impliziert, dass es ℓ verschiedene Elemente $z_1, \ldots, z_\ell$ in der $2r{+}1$-Nachbarschaft von $\vec{x}$ gibt, die paarweisen Abstand $> 2{\cdot}(2r{+}1)$ voneinander haben, deren r-Nachbarschaften allesamt die Formel δ erfüllen, und für die gilt: *Jedes* Element u, dessen r-Umgebung die Formel δ erfüllt, hat Abstand $\leqslant 2{\cdot}(2r{+}1)$ zu mindestens einem der $z_1, \ldots, z_\ell$ (denn sonst würde $\chi_{\ell+1}$ an Stelle von $\neg\chi_{\ell+1}$ gelten).

Dies wiederum bedeutet aber, dass *jedes* Element u, dessen r-Umgebung die Formel δ erfüllt, in der $3{\cdot}(2r{+}1)$-Nachbarschaft von $\vec{x}$ liegt.

Somit impliziert $\kappa_\ell(\vec{x}) \wedge \neg\chi_{\ell+1}$, dass $\mu(\vec{x})$ äquivalent ist zur Formel

$$\tilde{\mu}(\vec{x}) \;:=\; \exists y \Big(\mathit{dist}(\vec{x}, y) \leqslant 3{\cdot}(2r{+}1) \;\wedge\; \mathit{dist}(\vec{x}, y) > 2r{+}1 \;\wedge\; \delta(y) \Big).$$

Daher ist die Formel (I) äquivalent zu

$$\Big(\kappa_\ell(\vec{x}) \wedge \tilde{\mu}(\vec{x}) \wedge \chi_\ell \wedge \neg\chi_{\ell+1}\Big),$$

und diese Formel ist in Gaifman-Normalform, da die Formel $\tilde{\mu}(\vec{x})$ $(3{\cdot}(2r{+}1) + r)$-lokal um $\vec{x}$, also $(7r{+}3)$-lokal um $\vec{x}$ ist.

Damit ist nun (endlich) Fall 4 des Beweises von Theorem 4.9 abgeschlossen. Insgesamt haben wir per Induktion nach dem Aufbau von $\text{FO}[\sigma]$-Formeln gezeigt, dass jede $\text{FO}[\sigma]$-Formel äquivalent zu einer Formel in Gaifman-Normalform ist.

Außerdem führt der obige Beweis unmittelbar zu einem Algorithmus, der bei Eingabe einer Formel φ eine zu φ äquivalente Formel in Gaifman-Normalform berechnet. Dies schließt den Beweis von Theorem 4.9 ab. □

Ein alternativer, etwas kürzerer Beweis, der nicht per Induktion nach dem Formelaufbau sondern durch Verwendung von EF-Spielen bzw. Hin-und-Her-Systemen vorgeht, wird in Gaifmans Originalarbeit skizziert und im Buch von Ebbinghaus und Flum ausführlich dargestellt. Dieser "modelltheoretische" Beweis liefert allerdings nicht unmittelbar einen Algorithmus, der bei Eingabe einer Formel eine äquivalente Formel in Gaifman-Normalform berechnet.

4.2 Anwendungen des Satzes von Gaifman

Der Satz von Gaifman (Theorem 4.9) zeigt, dass Formeln der Logik erster Stufe nur sehr eingeschränkte Ausdrucksstärke besitzen. Ähnlich wie der Satz von Hanf (3.25) führt der Satz von Gaifman zu einem *Lokalitätsbegriff*, mit dessen Hilfe man zeigen kann, dass viele Eigenschaften nicht FO-definierbar sind. Der Satz von Gaifman führt daher unmittelbar zu Nicht-Ausdrückbarkeits-Resultaten für die Logik erster Stufe; diese werden in Abschnitt 4.2.1 vorgestellt.

Der Satz von Gaifman ist aber nicht nur für Nicht-Ausdrückbarkeits-Resultate nützlich, sondern auch für so genannte *algorithmische Meta-Theoreme*, die besagen, dass Formeln der Logik erster Stufe auf bestimmten Klassen von Strukturen sehr effizient ausgewertet werden können. Auf solche Resultate werden wir kurz in Abschnitt 4.2.2 eingehen.

4.2.1 Die Gaifman-Lokalität der Logik erster Stufe

Bisher haben wir uns meistens mit der Frage beschäftigt, welche *Klassen von Strukturen* durch *Sätze* der Logik erster Stufe (oder einer geeigneten anderen Logik) definiert werden können. In Verbindung mit den in Kapitel 0.1 skizzierten Anwendungen von Logiken als *Datenbank-Anfragesprachen* ist es jedoch oft auch interessant, Formeln zu untersuchen die freie Variablen besitzen. Der Begriff “Anfrage” lässt sich dabei folgendermaßen formalisieren.

4.10 Definition (Anfrage)**.** Sei σ eine relationale Signatur und sei $k \in \mathbb{N}_{\geqslant 1}$.

(a) Eine *k-stellige Anfrage* ist eine Abbildung Q, die jeder endlichen σ-Struktur $\mathfrak{A}$ eine k-stellige Relation $Q(\mathfrak{A}) \subseteq A^k$ zuordnet.

(b) Eine k-stellige Anfrage Q heißt FO*-definierbar*, falls es eine FO$[\sigma]$-Formel $\varphi(\vec{x})$ mit freien Variablen $\vec{x} = x_1, \ldots, x_k$ gibt, so dass für alle endlichen σ-Strukturen $\mathfrak{A}$ gilt:

$$Q(\mathfrak{A}) \;=\; \{\vec{a} \in A^k \,:\, \mathfrak{A} \models \varphi[\vec{a}]\}.$$

4.11 Beispiel. Sei $\sigma := \{E\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E besteht. Die 1-stellige Anfrage Isolierte-Punkte, die jedem Graphen $G = (V, E)$ die Menge

$$\textsf{Isolierte-Punkte}(G) \;:=\; \{\, v \in V \;\mid \text{ es gibt in } G \text{ keine Kante von oder zu } v \,\}$$

ist FO-definierbar durch die Formel $\varphi(x) := \neg\exists y\big(E(x,y) \vee E(y,x)\big)$.

Nun stellt sich natürlich die Frage, *welche* Anfragen FO-definierbar sind.

In Definition 3.27 haben wir bereits den Begriff der *Hanf-Lokalität* (als Eigenschaft von Strukturklassen) kennengelernt. Ein etwas anderer Lokalitätsbegriff, der sich nicht auf Strukturklassen, sondern auf Anfragen bezieht, ist die folgende *Gaifman-Lokalität*.

4.12 Definition (Gaifman-Lokalität)**.** Sei σ eine relationale Signatur und sei $k \in \mathbb{N}_{\geqslant 1}$. Eine k-stellige Anfrage Q heißt *Gaifman-lokal*, falls es ein $r \in \mathbb{N}$ gibt, so dass für alle endlichen σ-Strukturen $\mathfrak{A}$ und alle $\vec{a} \in A^k$ und $\vec{b} \in A^k$ gilt:

$$\text{Falls} \quad \left(\mathcal{N}_r^{\mathfrak{A}}(\vec{a}), \vec{a}\right) \;\cong\; \left(\mathcal{N}_r^{\mathfrak{A}}(\vec{b}), \vec{b}\right), \quad \text{so} \quad \Big(\, \vec{a} \in Q(\mathfrak{A}) \iff \vec{b} \in Q(\mathfrak{A}) \,\Big).$$

Aus dem Satz von Gaifman folgt unmittelbar, dass die Logik erster Stufe nur Gaifman-lokale Anfragen definieren kann:

4.13 Theorem (Gaifman-Lokalität von FO)**.** *Für jede relationale Signatur σ gilt:*
Alle FO$[\sigma]$*-definierbaren Anfragen sind Gaifman-lokal.*

Beweis: Sei $k \in \mathbb{N}$, sei $\vec{x} = x_1, \ldots, x_k$, sei $\varphi(\vec{x})$ eine FO$[\sigma]$-Formel, und sei Q die von $\varphi(\vec{x})$ definierte k-stellige Anfrage.

Gemäß dem Satz von Gaifman (Theorem 4.9) ist $\varphi(\vec{x})$ äquivalent zu einer Formel $\varphi'(\vec{x})$ in Gaifman-Normalform. D.h., $\varphi'(\vec{x})$ ist eine Boolesche Kombination von basis-lokalen Sätzen $\chi_1, \ldots, \chi_s$ und lokalen Formeln $\psi_1(\vec{x}), \ldots, \psi_t(\vec{x})$, für geeignete $s, t \in \mathbb{N}$.

Sei $r \in \mathbb{N}$ so dass jede der Formeln $\psi_i(\vec{x})$ r-lokal um $\vec{x}$ ist.
Betrachte nun eine beliebige σ-Struktur $\mathfrak{A}$. Seien $\vec{a} \in A^k$ und $\vec{b} \in A^k$ so dass $\left(\mathcal{N}_r^{\mathfrak{A}}(\vec{a}), \vec{a}\right) \cong \left(\mathcal{N}_r^{\mathfrak{A}}(\vec{b}), \vec{b}\right)$. Wir müssen zeigen, dass $\mathfrak{A} \models \varphi'[\vec{a}] \iff \mathfrak{A} \models \varphi'[\vec{b}]$. Offensichtlich genügt es, dafür folgendes zu zeigen:

(1) für alle $i \in \{1, \ldots, s\}$ gilt: $\mathfrak{A} \models \chi_i[\vec{a}] \iff \mathfrak{A} \models \chi_i[\vec{b}]$, und

(2) für alle $i \in \{1, \ldots, t\}$ gilt: $\mathfrak{A} \models \psi_i[\vec{a}] \iff \mathfrak{A} \models \psi_i[\vec{b}]$.

Punkt (1) gilt, weil χ_i ein *Satz* ist und daher nicht von $\vec{a}$ oder $\vec{b}$ abhängt.
Punkt (2) gilt, weil ψ_i r-lokal um $\vec{x}$ ist und daher nur von der r-Nachbarschaft um $\vec{a}$ bzw. $\vec{b}$ abhängt — und die Tupel $\vec{a}$ und $\vec{b}$ wurden gerade so gewählt, dass ihre r-Nachbarschaften isomorph sind. □

4.14 Bemerkung. Indem man zeigt, dass eine Anfrage Q *nicht Gaifman-lokal* ist, kann man (unter Verwendung von Theorem 4.13) zeigen, dass die Anfrage nicht FO-definierbar ist.

4.15 Beispiel. Die *Erreichbarkeits-Anfrage* E^*, die jedem endlichen Graphen $G = (V, E)$ die Relation

$$\mathsf{E}^*(G) \;:=\; \{\, (v, w) \in V \times V \mid \text{es gibt in } G \text{ einen Weg von } v \text{ nach } w \,\}$$

ist nicht Gaifman-lokal und daher laut Theorem 4.13 auch nicht FO-definierbar.

Beweis: Sei r eine beliebige natürliche Zahl. Sei G der Graph, der aus zwei disjunkten gerichteten Kreisen C und C' auf je $4r+4$ Knoten besteht. Ferner seien v und w zwei Knoten auf C vom Abstand $\text{Dist}^G(v, w) > 2r$, und sei w' ein beliebiger Knoten auf C'.

Offensichtlich ist dann $\big(\mathcal{N}_r^G(v,w), v, w\big)$ die disjunkte Vereinigung von zwei gerichteten Pfaden der Länge $2r+1$, so dass v und w die Mittelpunkte der beiden Pfade sind. Außerdem ist auch $\big(\mathcal{N}_r^G(v,w'), v, w'\big)$ die disjunkte Vereinigung von zwei gerichteten Pfaden der Länge $2r+1$, wobei v und w' die Mittelpunkte der beiden Pfade sind.
Somit gilt also $\big(\mathcal{N}_r^G(v,w), v, w\big) \;\cong\; \big(\mathcal{N}_r^G(v,w'), v, w'\big)$.
Andererseits gilt aber: $(v,w) \in \mathsf{E}^*(G)$ und $(v,w') \notin \mathsf{E}^*(G)$.
Somit ist die Erreichbarkeits-Anfrage E^* *nicht* Gaifman-lokal. □

4.2.2 Effiziente Auswertung von FO auf bestimmten Klassen von Strukturen

Aus den vorherigen Kapiteln wissen wir bereits, dass die kombinierte Komplexität des Auswertungsproblems für FO

AUSWERTUNGSPROBLEM FÜR FO AUF Fin
Eingabe: eine endliche Struktur $\mathfrak{A}$ und ein FO-Satz φ
Frage: Gilt $\mathfrak{A} \models \varphi$?

PSPACE-vollständig ist (Satz 2.51) und dass es einen Algorithmus gibt, der das Problem in Zeit $\mathcal{O}(k{\cdot}n^k)$ löst, wobei n die Größe (einer geeigneten Kodierung) der eingegebenen Struktur $\mathfrak{A}$ und k die Größe der eingegebenen Formel φ bezeichnet (der im Beweis von Lemma 1.59 skizzierte Algorithmus hält diese Zeitschranke ein). Auf der Klasse aller endlichen Strukturen ist das Auswertungsproblem für FO also i.A. schwierig.

Wenn man allerdings an Stelle der ganzen Klasse Fin nur *bestimmte* endliche Strukturen als Eingabe zulässt, kann das Auswertungsproblem für FO — unter Verwendung z.B. des Satzes von Gaifman — deutlich effizienter gelöst werden. Eine Art von Beispielen sind Klassen von Graphen von beschränktem Grad, die folgendermaßen definiert sind:

4.16 Definition (Klassen von Graphen von beschränktem Grad)**.**

(a) Sei G ein Graph. Der *Grad eines Knotens* v von G ist die Anzahl aller Kanten, die v als Ausgangspunkt oder Endpunkt haben.
Der *Grad von* G ist der maximale Grad eines Knotens von G.

(b) Sei C eine Klasse von Graphen und sei $d \in \mathbb{N}$.
C ist eine *Klasse von Graphen vom Grad* $\leqslant d$, falls jeder Graph $G \in \mathsf{C}$ Grad $\leqslant d$ hat.

(c) C ist eine *Klasse von Graphen von beschränktem Grad*, falls es eine Zahl $d \in \mathbb{N}$ gibt, so dass C eine Klasse von Graphen vom Grad $\leqslant d$ ist.

4.17 Theorem (Seese, 1996)**.**
Für jedes $d \in \mathbb{N}$ gibt es eine berechenbare Funktion $f_d : \mathbb{N} \to \mathbb{N}$, so dass das

AUSWERTUNGSPROBLEM FÜR FO AUF DER KLASSE ALLER ENDLICHEN GRAPHEN VOM GRAD $\leqslant d$	
Eingabe:	ein endlicher Graph G vom Grad $\leqslant d$ und ein FO$[E]$-Satz φ
Frage:	Gilt $G \models \varphi$?

in Zeit $f_d(k) \cdot n$ gelöst werden kann, wobei k die Größe der Formel φ und n die Größe (einer geeigneten Kodierung) des Graphen G bezeichnet.

Das heißt, für jeden festen FO*-Satz φ ist das Problem, zu gegebenem Graphen G vom Grad $\leqslant d$ zu entscheiden, ob $G \models \varphi$, in Linearzeit lösbar.*

Beweis: Seeses Originalbeweis argumentiert unter Verwendung des Satzes von Hanf. Wir werden hier im Folgenden einen auf dem Satz von Gaifman basierenden Beweis geben, der (im Gegensatz zu dem Hanf-basierten Ansatz) den Vorteil hat, dass er auch auf andere Klassen von Strukturen übertragen werden kann (siehe das folgende Theorem 4.18).

Sei $d \in \mathbb{N}$ fest, sei φ der gegebene FO$[E]$-Satz, und sei $G = (V^G, E^G)$ der gegebene Graph vom Grad $\leqslant d$. Wir schreiben n um die Größe (der Kodierung) von G zu bezeichnen.

Zunächst nutzen wir den Algorithmus aus Theorem 4.9, um φ in einen äquivalenten Satz φ' in *Gaifman-Normalform* zu übersetzen. Dann testen wir nacheinander für jeden einzelnen basis-lokalen Satz χ, der in φ' vorkommt, ob $G \models \chi$. Am Ende können wir dann die Resultate für die einzelnen basis-lokalen Sätze χ einfach kombinieren, um zu ermitteln, ob $G \models \varphi'$.

Sei im Folgenden χ ein basis-lokaler Satz der Form

$$\exists x_1 \cdots \exists x_\ell \, \Big(\Big(\bigwedge_{1 \leqslant i < j \leqslant \ell} \mathit{dist}(x_i, x_j) > 2{\cdot}r \Big) \, \wedge \, \Big(\bigwedge_{i=1}^{\ell} \psi(x_i) \Big) \Big) \,,$$

wobei $\ell, r \in \mathbb{N}$ und $\psi(x)$ r-lokal um x ist. Natürlich gilt $G \models \chi$ genau dann, wenn es mindestens ℓ Knoten vom paarweisen Abstand $> 2r$ in G gibt, so dass die r-Nachbarschaften dieser ℓ Knoten allesamt die r-lokale Formel ψ erfüllen. Um zu entscheiden, ob $G \models \chi$, gehen wir in 2 Schritten vor:

Schritt 1: Bestimme für jeden Knoten $v \in V^G$, ob $\mathcal{N}_r^G(v) \models \psi[v]$, und markiere diejenigen Knoten v, für die $\mathcal{N}_r^G(v) \models \psi[v]$ gilt, als *rot*.

Schritt 2: Entscheide, ob es in G ℓ *rote* Knoten vom paarweisen Abstand $> 2r$ gibt.

Zum Lösen von Schritt 1 beachte, dass für jeden Knoten v in einem Graphen G vom Grad $\leqslant d$ gilt:

$$|N_r^G(v)| \quad \leqslant \quad 1 + d + d^2 + \cdots + d^r \quad \leqslant \quad d^{r+1} \,,$$

d.h. die Größe von $N_r^G(v)$ hängt nicht von der Größe von G ab, sondern nur von den Zahlen d und r. Schritt 1 kann daher folgendermaßen gelöst werden:

Algorithmus 1:

1: **for** $v \in V^G$ **do**
2: berechne $\mathcal{N}_r^G(v)$
3: teste, ob $\mathcal{N}_r^G(v) \models \psi[v]$
4: **if** $\mathcal{N}_r^G(v) \models \psi[v]$ **then** markiere v *rot*
5: **endfor**

Jede der Zeilen 2–4 wird $|V^G|$-mal durchlaufen. Für jeden einzelnen Durchlauf durch Zeile 2 wird Zeit $\leqslant f_d^{(1)}(r)$ gebraucht (für eine geeignete Funktion $f_d^{(1)} : \mathbb{N} \to \mathbb{N}$), da $|N_r^G(v)| \leqslant d^{r+1}$ ist. Für jeden einzelnen Durchlauf durch Zeile 3 wird Zeit $\leqslant f_d^{(2)}(r, ||\psi||)$ gebraucht (für eine geeignete Funktion $f_d^{(2)} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$), da $|N_r^G(v)| \leqslant d^{r+1}$ ist. Für jeden einzelnen Durchlauf durch Zeile 4 wird Zeit $\mathcal{O}(1)$ gebraucht.

Insgesamt benötigt Algorithmus 1 also bei Eingabe eines Graphen G vom Grad $\leqslant d$ und einer r-lokalen Formel $\psi(x)$ also Zeit $\leqslant f_d^{(3)}(r, ||\psi||) \cdot n$, wobei $f_d^{(3)} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ eine geeignete Funktion ist.

Zum Lösen von Schritt 2 müssen wir nun noch ℓ rote Knoten vom paarweisen Abstand $> 2r$ finden. Dies wird durch den folgenden Algorithmus gewährleistet:

Algorithmus 2:

1: initialisiere R als die Menge aller *roten* Knoten in V^G
2: initialisiere $X := \varnothing$
% X ist die bisher gefundene Menge roter Knoten vom Abstand $> 2r$
3: **while** $R \neq \varnothing$ **do**
4: wähle ein beliebiges $v \in R$
5: $R := R \setminus N_{2r}^G(v)$
6: $X := X \cup \{v\}$
7: **endwhile**
8: **if** $|X| \geqslant \ell$ **then** akzeptiere G
9: **else**
10: $N_{4r}^G(X) := \bigcup_{x \in X} N_{4r}^G(x)$
11: nutze einen naiven Algorithmus zum Testen, ob es in $\mathcal{N}_{4r}^G(X)$ ℓ rote Knoten vom paarweisen Abstand $> 2r$ gibt.
12: **endif**

Dieser Algorithmus verwendet zunächst eine Greedy-Strategie, um aus den *roten* Knoten in G eine Menge X von roten Knoten mit paarweisem Abstand $> 2r$ auszuwählen. Dazu wird

in der Schleife in Zeilen 3–7 jeweils ein beliebiger Knoten v aus R ausgewählt und danach werden alle Knoten mit Abstand $\leqslant 2r$ von v aus R gelöscht.

Wurde auf diese Weise eine Menge X mit (mindestens) ℓ roten Knoten vom paarweisen Abstand $> 2r$ gefunden, so können wir anhalten und G akzeptieren.

Enthält die Menge X jedoch weniger als ℓ Knoten, so können wir daraus noch nicht schließen, dass wir G verwerfen können, denn wir könnten ja in der Schleife die Knoten v für X einfach ungeschickt ausgewählt haben. Was wir jedoch auf jeden Fall wissen ist, dass *jeder* rote Knoten von G in der $2r$-Nachbarschaft eines Elements aus X liegt. Daraus folgt auch, dass die $2r$-Nachbarschaft jedes *roten* Knotens in der $4r$-Nachbarschaft eines Elements aus X, also in der in Zeile 10 definierten Menge $N_{4r}^G(X)$ liegt. Da $|X| \leqslant \ell-1$ ist und da G ein Graph vom Grad $\leqslant d$ ist, wissen wir, dass

$$|N_{4r}^G(X)| \quad \leqslant \quad (\ell-1) \cdot d^{4r+1}.$$

Daher hängt die Zeit, die für Zeile 11 des Algorithmus benötigt wird, also nicht von der Größe von G sondern nur von einer Zahl $f_d^{(4)}(\ell, r)$ ab (für eine geeignete Funktion $f_d^{(4)} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$).

Insgesamt löst Algorithmus 2 damit die in Schritt 2 gestellte Aufgabe und benötigt dabei für die Zeilen 1–7 eine Laufzeit $\leqslant f_d^{(5)}(r) \cdot n$. Für die Zeilen 8–12 wird eine Laufzeit benötigt, die nur von d, ℓ und r, nicht aber von der Größe von G abhängt.

Insgesamt liefert die Hintereinander-Ausführung der Algorithmen 1 und 2 einen Algorithmus, der bei Eingabe eines basis-lokalen Satzes χ der Größe k und eines Graphen G vom Grad $\leqslant d$ in Zeit $f_d(k) \cdot n$ testet, ob $G \models \chi$ (wobei $f_d : \mathbb{N} \to \mathbb{N}$ eine geeignete Funktion ist). □

Ein weiteres Beispiel für eine Klasse von Strukturen, auf der das Auswertungsproblem für FO in Linearzeit Datenkomplexität gelöst werden kann, ist:

4.18 Theorem (Frick, Grohe, 2001)**.** *Es gibt eine berechenbare Funktion f, so dass das*

AUSWERTUNGSPROBLEM FÜR FO AUF PLANAREN GRAPHEN	
Eingabe:	ein planarer Graph G und ein FO-Satz φ
Frage:	Gilt $G \models \varphi$?

in Zeit $f(k) \cdot n$ gelöst werden kann, wobei k die Größe der Formel φ und n die Größe (einer geeigneten Kodierung) des Graphen G bezeichnet.

Das heißt, für jeden festen FO-Satz φ ist das Problem, zu gegebenem planaren Graphen G zu entscheiden, ob $G \models \varphi$, in Linearzeit lösbar.

Der Beweis von Theorem 4.18 basiert auf (a) dem Satz von Gaifman und (b) dem Konzept einer *Baumzerlegung* eines Graphen, das in dieser Vorlesung allerdings nicht genauer betrachtet wird. Wir werden hier daher keinen Beweis von Theorem 4.18 angeben.

Um die Aussage von Theorem 4.18 und Theorem 4.17 zu verdeutlichen, betrachten wir kurz ein konkretes Beispiel. Für jede feste Zahl $k \in \mathbb{N}$ kann das Problem

k-INDEPENDENT-SET	
Eingabe:	ein Graph G
Frage:	Gibt es in G eine unabhängige Menge der Größe k, d.h. eine Menge von k Knoten, zwischen denen es keine Kanten gibt?

durch einen Algorithmus gelöst werden, der Zeit $\Theta(n^k)$ benötigt (der Algorithmus testet einfach jede k-elementige Knotenmenge darauf, ob sie unabhängig ist). Andererseits ist zunächst völlig unklar, ob es einen Algorithmus geben kann, der das Problem in Linearzeit, also in Zeit $\mathcal{O}(n)$ löst. Da das k-INDEPENDENT-SET Problem jedoch durch den FO-Satz

$$\varphi \quad := \quad \exists x_1 \cdots \exists x_k \bigwedge_{1 \leqslant i < j \leqslant k} \Big(\neg x_i{=}x_j \ \wedge \ \neg E(x_i, x_j) \ \wedge \ \neg E(x_j, x_i) \Big)$$

definiert wird (in dem Sinn, dass für jeden Graphen G gilt: $G \models \varphi \iff G$ besitzt eine unabhängige Menge der Größe k), folgt aus Theorem 4.18 und Theorem 4.17, dass es Algorithmen gibt, die das k-INDEPENDENT-SET Problem in Zeit $\mathcal{O}(n)$ lösen, wenn man als Eingabe-Graphen ausschließlich *planare* Graphen oder Graphen von *beschränktem Grad* zulässt.

4.3 Untere Schranken für Formeln in Gaifman-Normalform

Insbesondere angesichts der in Abschnitt 4.2.2 dargestellten algorithmischen Anwendungen des Satzes von Gaifman ist es natürlich interessant zu wissen, wie effizient die laut Satz von Gaifman mögliche Umformung einer FO-Formel in eine äquivalente Formel in Gaifman-Normalform durchgeführt werden kann. Eine Analyse des Algorithmus, der sich aus dem Beweis von Theorem 4.9 ergibt, zeigt, dass die Laufzeit dieses Algorithmus verheerend ist, da sich die Laufzeit mit jedem Quantor, der in der Formel vorkommt, um mindestens eine Zweierpotenz erhöht. Für eine Formel mit k ineinander geschachtelten Quantoren ist die Laufzeit also größer als

$$\mathit{Turm}(k) \quad := \quad 2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}} \Big\} k.$$

4.19 Definition. Die Funktion $\mathit{Turm} : \mathbb{N} \to \mathbb{N}$ ist induktiv definiert durch $\mathit{Turm}(0) := 1$ und $\mathit{Turm}(k{+}1) := 2^{\mathit{Turm}(k)}$.

Die Tatsache, dass der eine Algorithmus zur Tranformation in Gaifman-Normalform, den wir bisher kennengelernt haben, eine verheerende Laufzeit besitzt, heißt natürlich noch nicht, dass die Tranformation prinzipiell nicht effizienter bewerkstelligt werden kann. Ziel dieses Abschnitts ist nun, zu zeigen, dass es in der Tat keinen wesentlich effizienteren Algorithmus geben kann, allein schon deshalb, weil es FO-Sätze gibt, deren kürzeste äquivalente Sätze in Gaifman-Normalform *extrem* lang sind. Dies wird durch das folgende Theorem präzisiert.

4.20 Theorem (Dawar, Grohe, Kreutzer, Schweikardt, 2007)**.** *Für jedes $h \in \mathbb{N}_{\geqslant 1}$ gibt es einen* FO$[E]$*-Satz φ_h der Größe $\mathcal{O}(h^4)$, so dass jeder zu φ_h äquivalente* FO$[E]$*-Satz in Gaifman-Normalform mindestens die Größe Turm(h) hat.*

Bevor wir den Beweis von Theorem 4.20 angeben können, müssen wir zunächst ein paar für den Beweis nützliche Werkzeuge einführen.

4.21 Definition (Baum-Kodierung natürlicher Zahlen)**.** Für natürliche Zahlen i, n schreiben wir bit(i, n), um das *i-te Bit der Binärdarstellung von n* zu bezeichnen. D.h., bit$(i, n) = 0$ falls $\lfloor \frac{n}{2^i} \rfloor$ gerade ist, und bit$(i, n) = 1$ falls $\lfloor \frac{n}{2^i} \rfloor$ ungerade ist. Induktiv definieren wir für jede natürliche Zahl n einen Baum $\mathcal{B}(n)$ wie folgt:

- $\mathcal{B}(0)$ ist der Baum, der aus genau einem Knoten besteht.
- Für $n \geqslant 1$ ist $\mathcal{B}(n)$ der Baum, der durch Erzeugen eines neuen Wurzelknotens entsteht, an den die Bäume $\mathcal{B}(i)$ angehängt werden, für alle i mit bit$(i, n) = 1$.

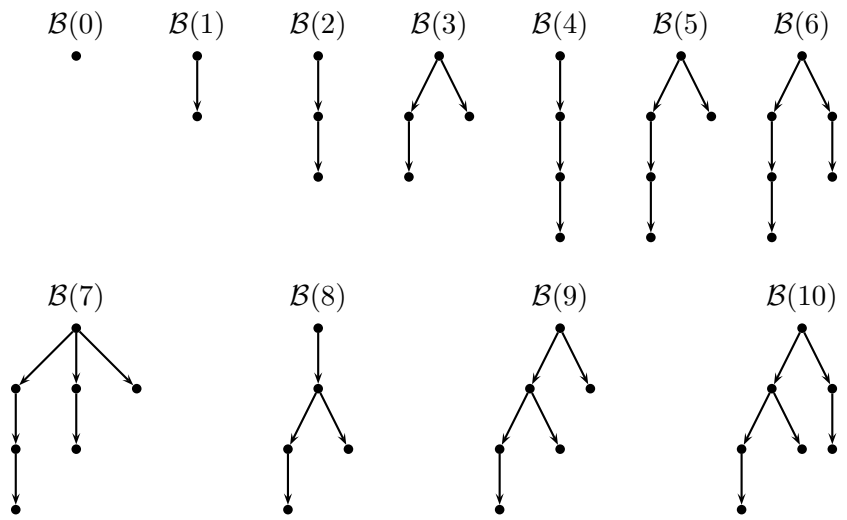


Abbildung 4.1: Die Bäume $\mathcal{B}(0)$ bis $\mathcal{B}(10)$

Für einen Baum $\mathcal{B}$ bezeichne *Höhe*$(\mathcal{B})$ die Länge des längsten gerichteten Pfades in $\mathcal{B}$. Man sieht leicht (Details: *Übung*), dass folgendes gilt:

$$\text{für alle } h, n \in \mathbb{N} \text{ ist} \quad \textit{Höhe}(\mathcal{B}(n)) \leqslant h \iff n < \textit{Turm}(h)\,. \tag{4.4}$$

Das nächste Lemma zeigt, dass Baum-Kodierungen natürlicher Zahlen durch relativ kurze Formeln der Logik erster Stufe verglichen werden können. In diesem Lemma und den folgenden Lemmas benutzen wir folgende Notation: Ist $\mathfrak{A} = (A, E^{\mathfrak{A}})$ ein gerichteter Graph und ist $a \in A$ ein Knoten in $\mathfrak{A}$, so bezeichnet A_a die Menge aller Knoten, die von a

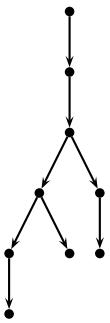


Abbildung 4.2: Der Baum $\mathcal{B}(2^{2^{10}})$

aus durch einen gerichteten Pfad in $\mathfrak{A}$ erreichbar sind (insbesondere gehört a zur Menge A_a). $\mathfrak{A}_a$ bezeichnet den induzierten Subgraph von $\mathfrak{A}$ mit Knotenmenge A_a, d.h. $\mathfrak{A}_a = (A_a, E^{\mathfrak{A}} \cap A_a^2)$.

4.22 Lemma. *Für jedes $h \in \mathbb{N}$ gibt es eine* FO$[E]$*-Formel $eq_h(x,y)$ der Länge $\mathcal{O}(h)$, so dass für alle Strukturen $\mathfrak{A} = (A, E^{\mathfrak{A}})$ und alle Knoten $a, b \in A$ gilt: Falls es Zahlen $m, n <$ Turm(h) gibt, so dass die Strukturen $\mathfrak{A}_a$ and $\mathfrak{A}_b$ isomorph zu $\mathcal{B}(m)$ und $\mathcal{B}(n)$ sind, so gilt:* $\quad \mathfrak{A} \models eq_h[a,b] \iff m = n$.

Beweis: Wir definieren die Formeln $eq_h(x,y)$ per Induktion nach h.
Wegen Turm$(0) = 1$ gilt: $m, n <$ Turm$(0) \iff m = n = 0$. Daher kann als $eq_0(x,y)$ einfach eine Formel gewählt werden, die *immer* erfüllt ist, z.B. $eq_0(x,y) := \forall z\ z{=}z$.

Für den Induktionsschritt sei $h \geqslant 1$. Wir nehmen an, dass die Formel $eq_{h-1}(x,y)$ bereits konstruiert ist. Um $eq_h(x,y)$ zu konstruieren, seien $\mathfrak{A} = (A, E^{\mathfrak{A}})$ und $a, b \in A$ gegeben, so dass es Zahlen $m, n <$ Turm(h) gibt, mit $\mathfrak{A}_a \cong \mathcal{B}(m)$ und $\mathfrak{A}_b \cong \mathcal{B}(n)$. Insbesondere sind dann die Teilbäume, deren Wurzeln die Kinder von a sind, isomorph zu $\mathcal{B}(m_1), \ldots, \mathcal{B}(m_k)$, wobei $m_1, .., m_k$ geeignete paarweise verschiedene Zahlen $<$ Turm$(h{-}1)$ sind. Analog sind die Teilbäume, deren Wurzeln die Kinder von b sind, isomorph zu $\mathcal{B}(n_1), \ldots, \mathcal{B}(n_\ell)$, wobei $n_1, .., n_\ell$ geeignete paarweise verschiedene Zahlen $<$ Turm$(h{-}1)$ sind.

Die Formel $eq_h(x,y)$ muss ausdrücken, dass $\{m_1, .., m_k\} = \{n_1, .., n_\ell\}$, d.h., dass es für jedes m_i ein n_j mit $m_i = n_j$ gibt und umgekehrt. Als ersten Ansatz wählen wir dazu

$$\begin{aligned} eq'_h(x,y) \;:=\; & \forall x' \Big(E(x,x') \rightarrow \exists y' \big(E(y,y') \wedge eq_{h-1}(x',y') \big) \Big) \wedge \\ & \forall y'' \Big(E(y,y'') \rightarrow \exists x'' \big(E(x,x'') \wedge eq_{h-1}(x'',y'') \big) \Big) . \end{aligned}$$

Diese Formel drückt das Gewünschte aus. Da in eq'_h die Formel eq_{h-1} zweimal vorkommt, ist die Formel eq'_h jedoch deutlich zu lang (denn wenn wir eq'_h auf diese Weise per Induktion nach h definieren, so erhalten wir Formeln, deren Länge exponentiell in h ist). Dieses exponentielle Wachstum der Formeln eq_h können wir aber durch den folgenden Trick ver-

hindern: Offensichtlich ist die Formel $eq'_h(x,y)$ äquivalent zur Formel

$$\Big(\big(\exists x'\ E(x,x')\big) \leftrightarrow \big(\exists y'\ E(y,y')\big)\Big) \wedge$$
$$\forall x' \Big(E(x,x') \rightarrow\ \exists y' \big(E(y,y')\ \wedge\ \forall y'' \Big(E(y,y'') \rightarrow\ \exists x'' \big(E(x,x'') \wedge \Big(eq_{h-1}(x',y') \wedge eq_{h-1}(x'',y'')\Big)\big)\big)\big)\Big).$$

Außerdem ist natürlich die Formel $\Big(eq_{h-1}(x',y') \wedge eq_{h-1}(x'',y'')\Big)$ äquivalent zur Formel

$$\forall u\, \forall v\, \Big(\big((u = x' \wedge v = y') \vee (u = x'' \wedge v = y'')\big) \rightarrow\ eq_{h-1}(u,v)\Big).$$

Insgesamt wählen wir daher $eq_h(x,y) :=$

$$\Big(\big(\exists x'\ E(x,x')\big) \leftrightarrow \big(\exists y'\ E(y,y')\big)\Big) \wedge$$
$$\forall x' \Big(E(x,x') \rightarrow\ \exists y' \big(E(y,y')\ \wedge\ \forall y'' \Big(E(y,y'') \rightarrow\ \exists x'' \big(E(x,x'') \wedge \forall u\, \forall v\, \Big(\big((u = x' \wedge v = y') \vee (u = x'' \wedge v = y'')\big) \rightarrow\ eq_{h-1}(u,v)\Big)\big)\big)\big)\Big)$$

und erhalten damit eine Formel, die das Gewünschte ausdrückt. Da außderdem in eq_h die Formel eq_{h-1} nur einmal vorkommt, erhält man per Induktion nach h, dass $||eq_h|| = \mathcal{O}(h)$. □

Unter Nutzung dieses Lemmas kann man leicht die beiden folgenden Lemmas beweisen, die Formeln bereitstellen, deren Länge polynomiell in h ist, und die Baum-Kodierungen identifizieren bzw. Arithmetik auf Baum-Kodierungen von Zahlen bis zur Größe $Turm(h)$ definieren können.

4.23 Lemma. *Für jedes $h \in \mathbb{N}$ gibt es eine* FO$[E]$*-Formel $encoding_h(x)$ der Länge $\mathcal{O}(h^2)$, so dass für alle Strukturen $\mathfrak{A} = (A, E^{\mathfrak{A}})$ und alle $a \in A$ gilt: $\mathfrak{A} \models encoding_h[a] \iff$ es gibt ein $i \in \{0, \ldots, Turm(h)-1\}$ so dass $\mathfrak{A}_a$ isomorph zu $\mathcal{B}(i)$ ist.*

Beweis: Wegen $Turm(0) = 1$ muss die Formel $encoding_0$ ausdrücken, dass $\mathfrak{A}_a$ isomorph zum Baum $\mathcal{B}(0)$ ist, der aus genau einem Knoten besteht. Daher können wir $encoding_0(x) := \neg\exists y\, E(x,y)$ setzen.

Für $h \geqslant 1$ wählen wir $encoding_h(x) :=$

$$\forall y\, \Big(E(x,y) \rightarrow encoding_{h-1}(y)\Big)\ \wedge$$
$$\forall y\, \forall y'\, \Big(\big(E(x,y) \wedge E(x,y') \wedge \neg y{=}y'\big) \rightarrow \neg eq_{h-1}(y,y')\Big).$$

Man sieht leicht, dass die Formel $encoding_h$ das Gewünschte ausdrückt. Hinsichtlich der Länge der Formel $encoding_h$ gibt es außerdem ein $c > 0$, so dass

$$\begin{array}{rcll} ||encoding_h|| & \leqslant & ||encoding_{h-1}|| + ||eq_{h-1}|| + c & \\ & \leqslant & ||encoding_{h-2}|| + ||eq_{h-2}|| + ||eq_{h-1}|| + 2{\cdot}c & \\ & \leqslant & \cdots & \\ & \leqslant & ||encoding_0|| + \sum_{h'=0}^{h-1} ||eq_{h'}|| + c{\cdot}h & = \mathcal{O}(h^2) \end{array}$$

(beachte dazu, dass laut Lemma 4.22 gilt: $||\mathit{eq}_{h'}|| = \mathcal{O}(h')$). □

4.24 Lemma. *Für jedes $h \in \mathbb{N}$ gibt es* FO$[E]$*-Formeln $\mathit{bit}_h(x,y)$ der Länge $\mathcal{O}(h)$, $\mathit{less}_h(x,y)$ der Länge $\mathcal{O}(h^2)$, $\mathit{min}(x)$ konstanter Länge (die nicht von h abhängt), $\mathit{succ}_h(x,y)$ der Länge $\mathcal{O}(h^3)$ und $\mathit{max}_h(x)$ der Länge $\mathcal{O}(h^4)$, so dass für alle Strukturen $\mathfrak{A} = (A, E^{\mathfrak{A}})$ und alle $a, b \in A$ gilt: Falls es Zahlen $m, n < \mathit{Turm}(h)$ gibt, so dass die Strukturen $\mathfrak{A}_a$ und $\mathfrak{A}_b$ isomorph zu $\mathcal{B}(m)$ und $\mathcal{B}(n)$ sind, dann gilt:*

(a) $\mathfrak{A} \models \mathit{bit}_h[a,b] \iff \mathrm{bit}(m,n) = 1$.

(b) $\mathfrak{A} \models \mathit{less}_h[a,b] \iff m < n$.

(c) $\mathfrak{A} \models \mathit{min}[a] \iff \mathfrak{A}_a$ *ist isomorph zu* $\mathcal{B}(0)$.

(d) $\mathfrak{A} \models \mathit{succ}_h[a,b] \iff m+1 = n$.

(e) $\mathfrak{A} \models \mathit{max}_h[a] \iff \mathfrak{A}_a$ *ist isomorph zu* $\mathcal{B}(\mathit{Turm}(h)-1)$.

Beweis: *(a):* Setze $\mathit{bit}_h(x,y) := \exists y' \big(E(y,y') \wedge \mathit{eq}_h(y',x)\big)$.

(b): Wir definieren die Formeln $\mathit{less}_h(x,y)$ per Induktion nach h.
Wegen $\mathit{Turm}(0) = 1$ gilt: $m, n < \mathit{Turm}(0) \iff m = n = 0$. Daher kann als $\mathit{less}_0(x,y)$ einfach eine Formel gewählt werden, die *nie* erfüllt wird, z.B. $\mathit{less}_0(x,y) := \exists x\, \neg x{=}x$.
Für $h \geqslant 1$ wählen wir $\mathit{less}_h(x,y) :=$

$$\begin{aligned}\exists y' \Big(& E(y,y') \wedge \\ & \forall x' \big(E(x,x') \to \neg \mathit{eq}_{h-1}(x',y')\big) \wedge \\ & \forall x'' \big((E(x,x'') \wedge \mathit{less}_{h-1}(y',x'')) \to \exists y''(E(y,y'') \wedge \mathit{eq}_{h-1}(y'',x''))\big) \Big).\end{aligned}$$

Anhand von Definition 4.21 sieht man leicht, dass die Formel less_h ausdrückt, dass es eine Zahl i gibt (die der Variablen y' in less_h entspricht), so dass $\mathrm{bit}(i,n) = 1$, $\mathrm{bit}(i,m) = 0$, und für jede Zahl j (die der Variablen x'' in less_h entspricht) mit $j > i$ und $\mathrm{bit}(j,m) = 1$ gilt, dass $\mathrm{bit}(j,n) = 1$. Daher drückt die Formel less_h aus, dass $m < n$ ist. Außerdem folgt auf die gleiche Art wie im Beweis von Lemma 4.23, dass $||\mathit{less}_h|| = \mathcal{O}(h^2)$.

(c): Da $\mathcal{B}(0)$ aus genau einem Knoten besteht, können wir $\mathit{min}(x) := \neg\exists y\, E(x,y)$ wählen.

(d): Wir definieren succ_h per Induktion nach h und nutzen dabei die Formeln eq_{h-1} und less_{h-1}. Für $h = 0$ kann als succ_0 natürlich eine Formel gewählt werden, die *nie* erfüllt ist.

Für $h \geqslant 1$ drücken die ersten drei Zeilen der folgenden Formel succ_h aus, dass es eine Zahl i gibt (die der Variablen y' entspricht), so dass i die kleinste Zahl mit $\mathrm{bit}(i,n) = 1$ ist, und dass für diese Zahl i gilt $\mathrm{bit}(i,m) = 0$. Die Zeilen 4 und 5 drücken aus, dass für jedes $j > i$ gilt: Falls $\mathrm{bit}(j,n) = 1$, so auch $\mathrm{bit}(j,m) = 1$, und umgekehrt, falls $\mathrm{bit}(j,m) = 1$, so auch $\mathrm{bit}(j,n) = 1$. Die letzten beiden Zeilen drücken folgendes aus: Falls $i \neq 0$, so ist $\mathrm{bit}(0,m) = 1$ und für jedes $j < i$ mit $\mathrm{bit}(j,m) = 1$ ist $\big(j{+}1 = i$ oder $\mathrm{bit}(j{+}1,m) = 1\big)$.

Insgesamt drückt die Formel succ_h also aus, dass $m+1=n$ ist.

$$
\begin{aligned}
\mathit{succ}_h(x,y) \;:=\quad & \exists y' \Big(\\
& E(y,y') \;\wedge\; \forall y'' \Big(\big(E(y,y'') \wedge \neg y''{=}y'\big) \rightarrow \mathit{less}_{h-1}(y',y'') \Big) \;\wedge \\
& \qquad\qquad \forall x' \Big(E(x,x') \rightarrow \neg \mathit{eq}_{h-1}(x',y') \Big) \;\wedge \\
& \forall y'' \Big(\big(E(y,y'') \wedge \mathit{less}_{h-1}(y',y'')\big) \rightarrow \exists x'' \, \big(E(x,x'') \wedge \mathit{eq}_{h-1}(x'',y'')\big) \Big) \;\wedge \\
& \forall x'' \Big(\big(E(x,x'') \wedge \mathit{less}_{h-1}(y',x'')\big) \rightarrow \exists y'' \, \big(E(y,y'') \wedge \mathit{eq}_{h-1}(y'',x'')\big) \Big) \;\wedge \\
& \neg \mathit{min}(y') \rightarrow \Big(\exists x' \big(E(x,x') \wedge \mathit{min}(x')\big) \wedge \\
& \qquad \forall x' \Big(\big(E(x,x') \wedge \mathit{less}_{h-1}(x',y')\big) \rightarrow \big(\exists z (\mathit{succ}_{h-1}(x',z) \wedge (z{=}y' \vee E(x,z)))\big) \Big) \Big) .
\end{aligned}
$$

Wegen $||\mathit{less}_{h-1}|| = \mathcal{O}((h{-}1)^2)$ und $||\mathit{eq}_{h-1}|| = \mathcal{O}(h{-}1)$, folgt (auf ähnliche Weise wie im Beweis von Lemma 4.23), dass $||\mathit{succ}_h|| = \mathcal{O}(\sum_{h' \leqslant h} h'^2) = \mathcal{O}(h^3)$.

(e): Wegen $\mathit{Turm}(0) = 1$, muss die Formel max_0 ausdrücken, dass $\mathfrak{A}_a$ isomorph zu $\mathcal{B}(0)$ ist, d.h. aus genau einem Knoten besteht. Wir können daher $\mathit{max}_0(x) := \neg\exists y \, E(x,y)$ wählen. Für $h \geqslant 1$ wählen wir

$$\mathit{max}_h(x) \;:=\; \mathit{encoding}_h(x) \;\wedge\; \mathit{max}'_h(x) \,,$$

wobei die Formel $\mathit{max}'_h(x)$ folgendermaßen per Induktion nach h definiert ist: Für $h = 0$ setze $\mathit{max}'_0(x) := \mathit{max}_0(x) := \neg\exists y \, E(x,y)$. Für $h \geqslant 1$ beachte, dass $\mathit{Turm}(h){-}1 = 2^{\mathit{Turm}(h-1)}{-}1$. Daher besteht der Baum $\mathcal{B}' := \mathcal{B}(\mathit{Turm}(h){-}1)$ aus einem Wurzelknoten, an den für jedes i mit $0 \leqslant i < \mathit{Turm}(h{-}1)$ ein Kind t_i angehängt ist, so dass $\mathcal{B}'_{t_i}$ isomorph zu $\mathcal{B}(i)$ ist. Wir wählen

$$
\begin{aligned}
\mathit{max}'_h(x) \;:=\; & \exists y \, \big(E(x,y) \wedge \mathit{min}(y)\big) \;\wedge \\
& \qquad \forall y \, \Big(E(x,y) \;\rightarrow \big(\, \mathit{max}'_{h-1}(y) \;\vee\; \exists z \, \big(E(x,z) \wedge \mathit{succ}_{h-1}(y,z)\big) \, \big) \Big) .
\end{aligned}
$$

Man sieht leicht, dass die Formel $\mathit{max}_h(x)$ das Gewünschte ausdrückt. Außerdem erhalten wir (auf ähnliche Art wie im Beweis von Lemma 4.23) wegen $||\mathit{succ}_{h-1}|| = \mathcal{O}((h{-}1)^3)$ und $||\mathit{eq}_{h-1}|| = \mathcal{O}(h{-}1)$, dass $||\mathit{max}'_h|| = \mathcal{O}(\sum_{h' \leqslant h} h'^3) = \mathcal{O}(h^4)$. Somit folgt $||\mathit{max}_h|| = \mathcal{O}(h^4)$. Insgesamt ist der Beweis von Lemma 4.24 damit beendet. □

Wir haben nun alle Werkzeuge bereitgestellt, die zum Beweis von Theorem 4.20 nötig sind.

Beweis von Theorem 4.20:
Wir nutzen die Baum-Kodierung natürlicher Zahlen, die in Definition 4.21 eingeführt wurde. Für $h \geqslant 1$ sei

$$\mathcal{W}_h \quad := \quad \bigsqcup_{0 \leqslant j < \mathit{Turm}(h)} \mathcal{B}(j)$$

der Wald, der aus der disjunkten Vereinigung der Bäume $\mathcal{B}(j)$ für alle $j \in \{0,..,\mathit{Turm}(h)-1\}$ besteht. Außerdem sei für jedes $i \in \{0,..,\mathit{Turm}(h)-1\}$

$$\mathcal{W}_h^{-i} \quad := \quad \bigsqcup_{\substack{0 \leqslant j < \mathit{Turm}(h) \\ \text{so dass } j \neq i}} \mathcal{B}(j)$$

der Wald, der aus der disjunkten Vereinigung der Bäume $\mathcal{B}(j)$ für alle j mit $j \neq i$ und $j \in \{0,..,\mathit{Turm}(h)-1\}$ besteht.

Nun sei $\mathit{root}(x)$ eine FO$[E]$-Formel, die ausdrückt, dass der Knoten x den Ein-Grad 0 hat, d.h. $\mathit{root}(x) := \neg\exists y\, E(y,x)$.
Unter Verwendung der Formeln aus Lemma 4.24 wählen wir den FO$[E]$-Satz φ_h wie folgt:

$$\begin{aligned} \varphi_h \;:=\; & \exists x \,\big(\mathit{root}(x) \wedge \mathit{min}(x)\big) \;\wedge \\ & \forall y \,\Big(\big(\mathit{root}(y) \wedge \neg \mathit{max}_h(y)\big) \rightarrow \exists z\big(\mathit{root}(z) \wedge \mathit{succ}_h(y,z)\big)\Big)\,. \end{aligned}$$

Gemäß Lemma 4.24 wissen wir, dass $||\varphi_h|| = \mathcal{O}(h^4)$, und dass

$$\mathcal{W}_h \models \varphi_h \qquad \text{und,} \quad \text{für alle } i < \mathit{Turm}(h), \quad \mathcal{W}_h^{-i} \not\models \varphi_h\,. \tag{4.5}$$

Sei nun φ' ein zu φ_h äquivalenter FO$[E]$-Satz in *Gaifman-Normalform*. Aus (4.5) folgt, dass

$$\mathcal{W}_h \models \varphi' \qquad \text{und,} \quad \text{für alle } i < \mathit{Turm}(h), \quad \mathcal{W}_h^{-i} \not\models \varphi'\,. \tag{4.6}$$

Unser Ziel ist, zu zeigen, dass $H := ||\varphi'|| \geqslant \mathit{Turm}(h)$. Wir führen einen Beweis durch Widerspruch und nehmen im Folgenden an, dass $H < \mathit{Turm}(h)$.

Da φ' in Gaifman-Normalform ist, ist φ' eine Boolesche Kombination basis-lokaler Sätze $\chi_1,..,\chi_K$, wobei jedes χ_k (für $k \in \{1,..,K\}$) von der Form

$$\chi_k \;:=\; \exists x_1 \cdots \exists x_{\ell_k} \Big(\Big(\bigwedge_{1 \leqslant i < j \leqslant \ell_k} \mathit{dist}(x_i,x_j) > 2r_k \Big) \;\wedge\; \Big(\bigwedge_{i=1}^{\ell_k} \psi_k(x_i) \Big) \Big)$$

ist mit $\ell_k, r_k \in \mathbb{N}_{\geqslant 1}$ und $\psi_k(x)$ r_k-lokal um x. Insbesondere gilt

$$\ell_1 + \cdots + \ell_K \;\leqslant\; ||\varphi'|| \;=:\; H\,. \tag{4.7}$$

Wir können o.B.d.A. annehmen, dass es eine Zahl $\tilde{K}$ mit $0 \leqslant \tilde{K} \leqslant K$ gibt, so dass

$$\text{für jedes } k \leqslant \tilde{K}, \quad \mathcal{W}_h \models \chi_k\,, \qquad \text{und} \qquad \text{für jedes } k > \tilde{K}, \quad \mathcal{W}_h \not\models \chi_k\,. \tag{4.8}$$

Für jedes $k \leqslant \tilde{K}$ wissen wir, dass $\mathcal{W}_h \models \chi_k$, d.h. es gibt Knoten $t_1^{(k)}, \ldots, t_{\ell_k}^{(k)}$ in $\mathcal{W}_h$, so dass die Formel

$$\Big(\bigwedge_{1 \leqslant i < j \leqslant \ell_k} \mathit{dist}(x_i,x_j) > 2r_k \Big) \;\wedge\; \Big(\bigwedge_{i=1}^{\ell_k} \psi_k(x_i) \Big) \tag{4.9}$$

in $\mathcal{W}_h$ erfüllt ist, wenn jede Variable x_i mit dem Knoten $t_i^{(k)}$ interpretiert wird. Die Menge $\{\, t_i^{(k)} \,:\, k \leqslant \tilde{K} \text{ und } i \leqslant \ell_k \,\}$ besteht aus höchstens $\ell_1 + \cdots + \ell_{\tilde{K}} \leqslant H$ verschiedenen Knoten (siehe (4.7)). Da wir annehmen, dass $H < \mathit{Turm}(h)$, und da $\mathcal{W}_h$ aus $\mathit{Turm}(h)$ disjunkten Bäumen besteht, muss es mindestens eine disjunkte Komponente $\mathcal{B}$ von $\mathcal{W}_h$ geben, in der keiner der Knoten aus $\{\, t_i^{(k)} \,:\, k \leqslant \tilde{K} \text{ und } i \leqslant \ell_k \,\}$ liegt. Sei $j \in \{0, .., \mathit{Turm}(h)-1\}$ so dass $\mathcal{B} = \mathcal{B}(j)$.

Natürlich enthält nun der Wald $\mathcal{W}_h^{-j}$, der aus $\mathcal{W}_h$ durch Löschen der disjunkten Komponente $\mathcal{B}(j)$ entsteht, immer noch sämtliche Knoten aus $\{\, t_i^{(k)} \,:\, k \leqslant \tilde{K} \text{ und } i \leqslant \ell_k \,\}$.

Hinsichtlich (4.9) beachte, dass jede Formel $\psi_k(x_i)$ r_k-lokal um x_i ist. Wenn die Variablen x_i mit den Knoten $t_i^{(k)}$ interpretiert werden, kann die Formel daher nur über die r_k-Nachbarschaft von $t_i^{(k)}$ "sprechen"; und diese r_k-Nachbarschaft ist in $\mathcal{W}_h^{-j}$ identisch zur r_k-Nachbarschaft in $\mathcal{W}_h$. Aus (4.9) erhalten wir daher, dass $\mathcal{W}_h^{-j} \models \chi_k$, für jedes $k \leqslant \tilde{K}$.

Betrachten wir nun die Formeln χ_k mit $k > \tilde{K}$. Aus (4.8) wissen wir, dass $\mathcal{W}_h \not\models \chi_k$, d.h., $\mathcal{W}_h \models \neg\chi_k$, wobei die Formel $\neg\chi_k$ von der folgenden Form ist:

$$\neg\exists x_1 \cdots \exists x_{\ell_k} \Big(\Big(\bigwedge_{1 \leqslant i < j \leqslant \ell_k} \mathit{dist}(x_i, x_j) > 2r_k \Big) \;\wedge\; \Big(\bigwedge_{i=1}^{\ell_k} \psi_k(x_i) \Big) \Big) .$$

Da $\psi_k(x_i)$ r_k-lokal um x_i ist, und da $\mathcal{W}_h^{-j}$ aus $\mathcal{W}_h$ durch Löschen einer disjunkten Komponente entsteht, gilt auch $\mathcal{W}_h^{-j} \models \neg\chi_k$. Insgesamt haben wir nun folgendes gezeigt:

$$\text{für jedes } k \leqslant \tilde{K}, \quad \mathcal{W}_h^{-j} \models \chi_k \,, \qquad \text{und} \qquad \text{für jedes } k > \tilde{K}, \quad \mathcal{W}_h^{-j} \not\models \chi_k \,. \tag{4.10}$$

(4.10) und (4.8) besagen, dass $\mathcal{W}_h^{-j}$ genau dieselben basis-lokalen Sätze aus $\{\chi_1, \ldots, \chi_K\}$ erfüllt wie $\mathcal{W}_h$. Da φ' eine Boolesche Kombination der Sätze $\chi_1, .., \chi_K$ ist, erhalten wir daher, dass $\mathcal{W}_h^{-j} \models \varphi' \iff \mathcal{W}_h \models \varphi'$. Dies widerspricht jedoch (4.6). Insgesamt ist damit der Beweis von Theorem 4.20 beendet. □

5 Fixpunktlogiken

In Kapitel 2.2 haben wir verschiedene Erweiterungen der Logik erster Stufe um Fixpunkt-Konstrukte kennengelernt, insbesondere die Logiken LFP, IFP und PFP. Standen dort vor allem die Anwendung dieser Logiken zur Beschreibung von Komplexitätsklassen im Vordergrund, so werden wir in diesem Kapitel genauer auf die Eigenschaften von Fixpunktlogiken an sich eingehen.

Anschließend an Abschnitt 3.3 werden wir zunächst zeigen, dass sich alle bisher behandelten Fixpunktlogiken in die Logik $L^{\omega}_{\infty\omega}$ einbetten lassen. Anschließend werden wir auf eine syntaktische Variante solcher Logiken eingehen, die das Aufschreiben von Formeln stark vereinfacht und modularisiert.

5.1 Fixpunktlogiken und $L^{\omega}_{\infty\omega}$

Zur Erinnerung: In Definition 1.11 hatten wir bereits festgelegt, dass

- Fin die Klasse aller endlichen Strukturen bezeichnet (über beliebigen Signaturen),
- FinOrd die Klasse aller endlichen linear geordneten Strukturen bezeichnet (über beliebigen Signaturen, die ein 2-stelliges Relationssymbol $<$ enthalten, das stets durch eine lineare Ordnung auf dem Universum der jeweiligen Struktur interpretiert wird).

5.1 Definition. (a) Für Logiken $\mathscr{L}$ und $\mathscr{L}'$ und eine Klasse C von Strukturen schreiben wir *"$\mathscr{L} \leqslant \mathscr{L}'$ auf C"*, falls es für jede Formel $\varphi \in \mathscr{L}$ eine Formel $\varphi' \in \mathscr{L}'$ gibt, die zu φ auf C äquivalent ist, d.h. $\mathit{frei}(\varphi') = \mathit{frei}(\varphi)$ und für alle $\mathfrak{A} \in \mathsf{C}$ und alle Interpretationen $\vec{a} \in A$ der freien Variablen von φ gilt: $\mathfrak{A} \models \varphi[\vec{a}] \iff \mathfrak{A} \models \varphi'[\vec{a}]$.

(b) Entsprechend schreiben wir *"$\mathscr{L} = \mathscr{L}'$ auf C"*, falls $\mathscr{L} \leqslant \mathscr{L}'$ und $\mathscr{L}' \leqslant \mathscr{L}$ auf C.
Analog schreiben wir *"$\mathscr{L} < \mathscr{L}'$ auf C"* , falls $\mathscr{L} \leqslant \mathscr{L}'$ auf C und $\mathscr{L}' \not\leqslant \mathscr{L}$ auf C.

(c) Ist $\mathscr{K}$ eine Komplexitätsklasse, so schreiben wir *"$\mathscr{L} = \mathscr{K}$ auf C"*, falls $\mathscr{L}$ die Klasse $\mathscr{K}$ auf C *beschreibt* (im Sinne von Definition 2.1).

(d) Entsprechend schreiben wir *"$\mathscr{L} \leqslant \mathscr{K}$ auf C"*, falls die Datenkomplexität des Auswertungsproblems für $\mathscr{L}$ auf C in $\mathscr{K}$ liegt (im Sinne von Definition 2.1 (a)).
Analog schreiben wir *"$\mathscr{K} \leqslant \mathscr{L}$ auf C"*, falls jedes Problem in $\mathscr{K}$ durch einen Satz in $\mathscr{L}$ beschrieben werden kann (im Sinne von Definition 2.1 (b)).

5.2 Satz. *Es gilt:* $\mathrm{LFP} \leqslant \mathrm{IFP} \leqslant \mathrm{PFP} < L^{\omega}_{\infty\omega}$ *auf* Fin.

Beweis: LFP $\leqslant$ IFP $\leqslant$ PFP auf Fin wurde schon in Kapitel 2.2 (Proposition 2.37 und 2.45) gezeigt.

Wir zeigen als nächstes, dass $L^{\omega}_{\infty\omega} \not\leqslant$ PFP. Sei dazu $J \subseteq \mathbb{N}$ eine unentscheidbare Menge, z.B. die Menge der Gödel-Nummern aller Turing-Maschinen, die bei leerer Eingabe halten. Sei φ_J der in Beispiel 3.49 konstruierte $L^3_{\infty\omega}$-Satz, der genau die Strukturen $\mathfrak{A} := (A, <^{\mathfrak{A}})$ als Modelle hat, für die $|A| \in J$ und $<^{\mathfrak{A}}$ eine lineare Ordnung auf A ist. Da J unentscheidbar ist, ist auch $\mathrm{Mod}_{\mathsf{Fin}}(\varphi)$ unentscheidbar. Andererseits ist jede in PFP definierbare Klasse von Strukturen entscheidbar (und kann laut Lemma 2.44 sogar in PSPACE entschieden werden). Folglich kann es keinen zum $L^3_{\infty\omega}$-Satz φ_J äquivalenten PFP-Satz geben.

Es bleibt zu zeigen, dass PFP $\leqslant L^{\omega}_{\infty\omega}$ auf Fin, d.h. dass jede PFP-Formel φ auf Fin äquivalent ist zu einer $L^{\omega}_{\infty\omega}$-Formel $\hat{\varphi}$. Der Beweis folgt per Induktion über den Formelaufbau. Der einzige interessante Fall ist, wenn φ die Form

$$[\mathbf{pfp}_{R,\vec{x}}\, \psi](\vec{t})$$

hat. Dabei sei $r := \mathit{ar}(R)$, $\vec{x} = x_1, \ldots, x_r$ und $\vec{t} = t_1, \ldots, t_r$.

Gemäß Induktionsannahme ist ψ äquivalent zu einer $L^{\omega}_{\infty\omega}$-Formel $\hat{\psi}$. Wegen $L^{\omega}_{\infty\omega} = \bigcup_{k\in\mathbb{N}} L^k_{\infty\omega}$ gibt es ein $k \in \mathbb{N}$ so dass $\hat{\psi} \in L^k_{\infty\omega}$. Insbesondere kommen in $\hat{\psi}$ nur k verschiedene Variablen erster Stufe vor, und $\mathit{frei}(\hat{\psi}) = \mathit{frei}(\psi) \supseteq \{x_1, \ldots, x_r\}$.

Seien $\vec{y} := y_1, \ldots, y_r$ neue Variablen erster Stufe, die *nicht* in $\hat{\psi}$ vorkommen. Wir definieren induktiv für jedes $\alpha \in \mathbb{N}$ eine $L^{k+r}_{\infty\omega}$-Formel $\hat{\psi}^{\alpha}(\vec{x})$, die die α-te Stufe der partiellen Fixpunktinduktion über $\hat{\psi}$ definiert:

- $\hat{\psi}^0(\vec{x}) := \neg x_1{=}x_1$ (d.h. $\hat{\psi}^0$ ist eine Formel, die *nie* erfüllt ist)
- $\hat{\psi}^{\alpha+1}(\vec{x})$ entsteht aus $\hat{\psi}(\vec{x})$, indem jedes Vorkommen eines Atoms der Form $R(\vec{u})$, für ein Tupel $\vec{u} = u_1, \ldots, u_r$ von Variablen erster Stufe und/oder Konstantensymbolen aus σ durch die (zur Formel $\hat{\psi}^{\alpha}(\vec{u})$ äquivalenten) Formel

$$\exists y_1 \cdots \exists y_r \Big(\bigwedge_{i=1}^{r} y_i{=}u_i \;\wedge\; \exists x_1 \cdots \exists x_r \big(\big(\bigwedge_{i=1}^{r} x_i{=}y_i \big) \wedge \hat{\psi}^{\alpha}(\vec{x}) \big) \Big)$$

  ersetzt wird.[1]

  Hierbei ist die Verwendung der neuen Variablen $y_1, \ldots, y_r$ nötig, da die Variablen $x_1, \ldots, x_r$ als Terme in $\vec{u}$ vorkommen könnten.

Per Induktion nach α zeigt man leicht, dass für jede Struktur $\mathfrak{A}$ und alle $\vec{a} \in A^r$ gilt:

$$\vec{a} \in R^{\alpha} \quad \Longleftrightarrow \quad \mathfrak{A} \models \hat{\psi}^{\alpha}[\vec{a}]\,,$$

[1] Zur Erinnerung: In Kapitel 3.3 hatten wir vereinbart, *Substitutionen* in Bezug auf k-Variablen Logiken zu vermeiden und lieber explizite Variablenumbenennungen zu verwenden.

wobei R^α die α-te Stufe der partiellen Fixpunktinduktion über ψ in $\mathfrak{A}$ bezeichnet. Daher ist die Formel $\varphi := [\mathbf{pfp}_{R,\vec{x}}\,\psi](\vec{t})$ äquivalent zur $L^{k+r}_{\infty\omega}$-Formel

$$\hat{\varphi} \quad := \quad \bigvee_{\alpha\in\mathbb{N}} \Big(\hat{\psi}^\alpha(\vec{t}) \;\wedge\; \forall\vec{x}\,\big(\hat{\psi}^\alpha(\vec{x}) \leftrightarrow \hat{\psi}^{\alpha+1}(\vec{x})\big) \Big)\,.$$

Hierbei steht $\hat{\psi}^\alpha(\vec{t})$ wiederum für $\exists\vec{y}\,\Big(\bigwedge_{i=1}^{r} y_i{=}t_i \;\wedge\; \exists\vec{x}\,\big(\; \bigwedge_{i=1}^{r} x_i{=}y_i \;\wedge\; \hat{\psi}^\alpha(\vec{x})\;\big)\Big)$. □

Aus Satz 5.2, Beispiel 3.61 und Theorem 3.62 folgt sofort:

5.3 Korollar. *(a) Die Klasse* Even *ist nicht* PFP*-definierbar in* Fin.

(b) Die Klasse Hamilton *ist nicht* PFP*-definierbar in* Fin.

Da man selbstverständlich die Klasse Even in Polynomialzeit entscheiden kann, folgt daraus wiederum:

5.4 Korollar. *Es gilt*

(a) PFP $<$ PSPACE *auf* Fin *(zum Vergleich beachte:* PFP $=$ PSPACE *auf* FinOrd*)*

(b) LFP $\leqslant$ IFP $<$ PTIME *auf* Fin *(zum Vergleich beachte:* LFP $=$ IFP $=$ PTIME *auf* FinOrd*)*

(c) PTIME $\not\leqslant$ PFP *auf* Fin

(d) Falls PTIME $\neq$ PSPACE*, so gilt* PFP $\not\leqslant$ PTIME *auf* Fin.
PTIME *und* PFP *liegen vermutlich also schief zueinander.*

Beweis: *Übung.* □

In den folgenden Abschnitten wird die Ausdrucksstärke der verschiedenen Fixpunktlogiken genauer untersucht.

5.2 Simultane Fixpunkte

5.5 Definition. Sei $m \in \mathbb{N}_{\geqslant 1}$ und seien $A_1, \ldots, A_m$ Mengen. Ferner sei für jedes $i \in \{1, \ldots, m\}$ eine Abbildung

$$F_i \;:\; \mathrm{Pot}(A_1) \times \cdots \times \mathrm{Pot}(A_m) \;\to\; \mathrm{Pot}(A_i)$$

gegeben.

(a) Die *simultane Abbildung über* $F_1, \ldots, F_m$, kurz: $\vec{F} := (F_1, \ldots, F_m)$, ist definiert als

$$\begin{array}{rcl} \vec{F}: \;\; \mathrm{Pot}(A_1) \times \cdots \times \mathrm{Pot}(A_m) & \to & \mathrm{Pot}(A_1) \times \cdots \times \mathrm{Pot}(A_m) \\ \vec{R} \;:=\; (R_1, \ldots, R_m) & \mapsto & \big(F_1(\vec{R}), \ldots, F_m(\vec{R})\big)\,. \end{array}$$

(b) $\vec{F}$ heißt *monoton*, wenn für alle $\vec{R} = (R_1, .., R_m)$ und $\vec{Q} = (Q_1, .., Q_m)$ im Definitionsbereich von $\vec{F}$ gilt: Ist $\vec{R} \subseteq \vec{Q}$, so auch $\vec{F}(\vec{R}) \subseteq \vec{F}(\vec{Q})$.
Hierbei bedeutet "$\vec{R} \subseteq \vec{Q}$", dass $R_i \subseteq Q_i$ für alle $i \in \{1, .., m\}$.

(c) $\vec{F}$ heißt *inflationär*, wenn für alle $\vec{R} = (R_1, .., R_m)$ im Definitionsbereich von $\vec{F}$ gilt: $\vec{R} \subseteq \vec{F}(\vec{R})$.

5.6 Lemma. *Sei $m \in \mathbb{N}_{\geqslant 1}$ und seien $A_1, .., A_m$ Mengen. Ferner sei für jedes $i \in \{1, .., m\}$ eine Abbildung*

$$F_i \ : \ \mathrm{Pot}(A_1) \times \cdots \times \mathrm{Pot}(A_m) \ \to \ \mathrm{Pot}(A_i)$$

gegeben, die komponentenweise monoton ist, d.h. es gilt für alle $R_1 \subseteq A_1, \ldots, R_m \subseteq A_m$, für alle $j \in \{1, .., m\}$ und alle $R'_j \subseteq A_j$ mit $R_j \subseteq R'_j$, dass

$$F_i(R_1, .., R_{j-1}, R_j, R_{j+1}, .., R_m) \quad \subseteq \quad F_i(R_1, .., R_{j-1}, R'_j, R_{j+1}, .., R_m) \, .$$

Dann ist die simultane Abbildung $\vec{F} := (F_1, .., F_m)$ monoton.

Beweis: *Übung.* □

Analog zu dem Vorgehen in Kapitel 2.2 können wir nun *Induktionsstufen* von simultanen Abbildungen bilden: Sind wie zuvor Abbildungen $F_1, \ldots, F_m$ mit

$$F_i \ : \ \mathrm{Pot}(A_1) \times \cdots \times \mathrm{Pot}(A_m) \ \ \to \ \ \mathrm{Pot}(A_i)$$

gegeben, so definieren wir induktiv für jedes $\alpha \in \mathbb{N}$ ein Tupel $\vec{R}^\alpha = (R_1^\alpha, .., R_m^\alpha)$ von Mengen durch

- $\vec{R}^0 := (\varnothing, \ldots, \varnothing)$
- $\vec{R}^{\alpha+1} := \vec{F}(\vec{R}^\alpha)$.

Das Tupel $\vec{R}^\alpha$ heißt *α-te Induktionsstufe* (oder auch: α-te Stufe) von $\vec{F} = (F_1, .., F_m)$.

Ist $\vec{F}$ *monoton* und sind die Mengen $A_1, .., A_m$ *endlich*, so existiert ein $\alpha \in \mathbb{N}$ mit

$$\vec{R}^\alpha \ = \ \vec{R}^{\alpha+1} \ =: \ \vec{R}^\infty .$$

Analog zum Satz von Knaster und Tarski (Satz 2.15) zeigt man leicht, dass für *montone* Abbildungen $\vec{F}$ der

$$\textit{kleinste simultane Fixpunkt} \ \ \mathbf{s\text{-}lfp}(\vec{F})$$

immer existiert, und dass gilt: $\mathbf{s\text{-}lfp}(\vec{F}) = \vec{R}^\infty$.

Die i-te Komponente des simultanen kleinsten Fixpunkts von $\vec{F}$ bezeichnen wir im Folgenden mit $\mathbf{s\text{-}lfp}(\vec{F})_i$ bzw. mit R_i^∞.

5.7 Definition (simultane Abbildungen, die durch Logik-Formeln definiert werden)**.** Sei σ eine Signatur, sei $m \in \mathbb{N}_{\geqslant 1}$, und seien $R_1, .., R_m$ Relationsvariablen der Stelligkeiten $r_1, .., r_m$. Für jedes $i \in \{1, .., m\}$ sei eine Formel $\varphi_i(\vec{x}_i, R_1, .., R_m)$ über der Signatur σ gegeben, wobei $\vec{x}_i$ ein Tupel aus r_i verschiedenen Variablen erster Stufe ist.
Auf einer endlichen σ-Struktur $\mathfrak{A}$ definiert φ_i eine Abbildung

$$\begin{aligned} F_{\varphi_i,\mathfrak{A}} \ : \ \ \mathrm{Pot}(A^{r_1}) \times \cdots \times \mathrm{Pot}(A^{r_m}) \ \ &\to \ \ \mathrm{Pot}(A^{r_i}) \\ (R_1, .., R_m) \ \ &\mapsto \ \ \big\{\, \vec{a} \in A^{r_i} \ : \ (\mathfrak{A}, R_1, .., R_m) \models \varphi_i[\vec{a}] \,\big\}. \end{aligned}$$

Die von $\vec{\varphi} := (\varphi_1, \ldots, \varphi_m)$ in $\mathfrak{A}$ definierte simultane Abbildung ist

$$\vec{F}_{\vec{\varphi},\mathfrak{A}} \ := \ \big(\, F_{\varphi_1,\mathfrak{A}}, \ \ldots, \ F_{\varphi_m,\mathfrak{A}} \,\big)\,.$$

Sind alle φ_i *positiv* in den Variablen $R_1, .., R_m$, so folgt mit Lemma 5.6 leicht, dass auf jeder endlichen Struktur $\mathfrak{A}$ die simultane Abbildung $\vec{F}_{\vec{\varphi},\mathfrak{A}}$ *monoton* ist, und dass somit der simultane Fixpunkt $\mathbf{s\text{-}lfp}(\vec{F}_{\vec{\varphi},\mathfrak{A}})$ existiert, den wir oft auch mit $\mathbf{s\text{-}lfp}(\vec{\varphi})^{\mathfrak{A}}$ bezeichnen.

5.8 Beispiel. In diesem Beispiel soll eine simultane Fixpunktinduktion verwendet werden, um in gerichteten Graphen Pfade gerader bzw. ungerader Länge zu definieren. Die Idee ist, eine simultane Induktion über zweistellige Variablen R_1 und R_2 zu führen, so dass im simultanen kleinsten Fixpunkt gilt: in R_1 kommen alle Paare (a, b) vor, zwischen denen ein Pfad ungerader Länge existiert, und entsprechend kommen in R_2 alle Paare (a, b) vor, zwischen denen ein Pfad gerader Länge existiert. Sei dazu

$$\begin{aligned} \varphi_1(x, y, R_1, R_2) \ &:= \ \ E(x, y) \ \vee \ \exists z\big(\, E(x, z) \wedge R_2(z, y) \,\big) \\ \varphi_2(x, y, R_1, R_2) \ &:= \ \ \ x = y \ \vee \ \exists z\big(\, E(x, z) \wedge R_1(z, y) \,\big). \end{aligned}$$

In einem Graphen $G := (V, E)$ definiert $\vec{\varphi} := (\varphi_1, \varphi_2)$ die Abbildung $\vec{F}_{\vec{\varphi},G} = (F_{\varphi_1,G}, F_{\varphi_2,G})$, wobei für jedes $i \in \{1, 2\}$ und alle $R_1, R_2 \subseteq V^2$ gilt:

$$F_{\varphi_i,G}(R_1, R_2) \quad := \quad \big\{\, (u, v) \in V^2 \ : \ (G, R_1, R_2) \models \varphi_i[u, v] \,\big\}\,.$$

Die ersten Induktionsstufen der dadurch ausgelösten Fixpunktinduktion lauten

$$\begin{aligned} R_1^0 &:= \varnothing & R_2^0 &:= \varnothing \\ R_1^1 &:= \left\{ (u, v) : \begin{array}{l} \text{es gibt Pfad der Länge 1} \\ \text{von } u \text{ nach } v \end{array} \right\} & R_2^1 &:= \{\, (u, u) : u \in V \,\} \\ R_1^2 &:= R_1^1 & R_2^2 &:= \left\{ (u, v) : \begin{array}{l} \text{es gibt Pfad der Länge} \\ \text{0 oder 2 von } u \text{ nach } v \end{array} \right\} \\ R_1^3 &:= \left\{ (u, v) : \begin{array}{l} \text{es gibt Pfad der Länge} \\ \text{1 oder 3 von } u \text{ nach } v \end{array} \right\} & R_2^3 &:= R_2^2 \\ &\ \ \vdots & &\ \ \vdots \end{aligned}$$

Insgesamt gilt also

$$\mathbf{s\text{-}lfp}(\vec{F}_{\vec{\varphi},G})_1 \;=\; \big\{\,(u,v)\;:\;\text{es gibt einen Pfad ungerader Länge von } u \text{ nach } v\,\big\}$$

$$\mathbf{s\text{-}lfp}(\vec{F}_{\vec{\varphi},G})_2 \;=\; \big\{\,(u,v)\;:\;\text{es gibt einen Pfad gerader Länge von } u \text{ nach } v\,\big\}\,.$$

5.9 Notation. Für Formeln $\varphi_1(\vec{x}_1, R_1, .., R_m),\ \ldots,\ \varphi_m(\vec{x}_m, R_1, .., R_m)$ schreiben wir

$$S \;:=\; \begin{cases} R_1(\vec{x}_1) & \leftarrow & \varphi_1(\vec{x}_1, R_1, \ldots, R_m) \\ & \vdots & \\ R_m(\vec{x}_m) & \leftarrow & \varphi_m(\vec{x}_m, R_1, \ldots, R_m) \end{cases}$$

und nennen S ein "System von Formeln".

Ist $\mathfrak{A}$ eine endliche Struktur, so schreiben wir oft $\mathbf{s\text{-}lfp}(S)^{\mathfrak{A}}$ um $\mathbf{s\text{-}lfp}(\varphi_1, .., \varphi_m)^{\mathfrak{A}}$ (d.h. $\mathbf{s\text{-}lfp}(\vec{F}_{\vec{\varphi},\mathfrak{A}})$) zu bezeichnen. Für $i \in \{1, .., m\}$ bezeichnet $\mathbf{s\text{-}lfp}(S)^{\mathfrak{A}}_i$ die i-te Komponente von $\mathbf{s\text{-}lfp}(S)^{\mathfrak{A}}$.

5.10 Definition (Simultane kleinste Fixpunktlogik S-LFP)**.** Sei σ eine Signatur.
Die Formelmenge S-LFP$[\sigma]$ ist induktiv durch die Regeln (A1), (A2), (A3), (BC) und (Q1) der Logik erster Stufe sowie die folgende Regel (S-LFP) definiert:

(S-LFP): Ist $m \in \mathbb{N}_{\geqslant 1}$, sind $R_1, .., R_m$ Relationsvariablen der Stelligkeiten $r_1, .., r_m \in \mathbb{N}_{\geqslant 1}$, sind $\varphi_1, .., \varphi_m$ S-LFP$[\sigma]$-Formeln, von denen jede positiv in den Relationsvariablen $R_1, .., R_m$ ist, und ist für alle $i \in \{1, .., m\}$ $\vec{x}_i$ ein Tupel, das aus r_i verschiedenen Variablen erster Stufe besteht, so ist für jedes $i \in \{1, .., m\}$

$$[\mathbf{lfp}\ R_i : S](\vec{t})$$

eine S-LFP$[\sigma]$-Formel, wobei $\vec{t}$ ein r_i-Tupel aus Variablen erster Stufe und/oder Konstantensymbolen aus σ ist und

$$S \;:=\; \begin{cases} R_1(\vec{x}_1) & \leftarrow & \varphi_1(\vec{x}_1, R_1, \ldots, R_m) \\ & \vdots & \\ R_m(\vec{x}_m) & \leftarrow & \varphi_m(\vec{x}_m, R_1, \ldots, R_m)\,. \end{cases}$$

Die Semantik der Logik S-LFP ist analog zur Semantik der Logik erster Stufe definiert, wobei für eine Formel $\psi := [\mathbf{lfp}\ R_i : S](\vec{t})$ und eine endliche $(\sigma \dot{\cup} \mathit{frei}(\psi))$-Struktur $\mathfrak{A}$ gilt:

$$\mathfrak{A} \;\models\; [\mathbf{lfp}\ R_i : S](\vec{t}) \qquad :\Longleftrightarrow \qquad \vec{t}^{\,\mathfrak{A}} \;\in\; \mathbf{s\text{-}lfp}(S)^{\mathfrak{A}}_i\,.$$

5.11 Beispiel. Seien φ_1, φ_2 die Formeln aus Beispiel 5.8 und sei

$$S \;:=\; \begin{cases} R_1(x,y) & \leftarrow & \varphi_1(x, y, R_1, R_2) \\ R_2(x,y) & \leftarrow & \varphi_2(x, y, R_1, R_2)\,. \end{cases}$$

Dann gilt für jeden Graphen $G := (V, E)$ und alle Knoten $u, v \in V$:

$$G \models \big([\mathbf{lfp}\ R_1 : S](x,y)\big)\,[u,v] \quad\Longleftrightarrow\quad \begin{array}{l}\text{in } G \text{ gibt es einen Pfad} \\ \textit{ungerader} \text{ Länge von } u \text{ nach } v.\end{array}$$

Das Verwenden simultaner Fixpunkte erlaubt es oft, Formeln sehr viel übersichtlicher und modularer zu schreiben, als es die herkömmliche kleinste Fixpunktlogik erlaubt. Es stellt sich hierbei natürlich die Frage, ob *simultane* Fixpunkte die Ausdrucksstärke der resultierenden Logik erhöhen. Wie wir als nächstes zeigen werden, ist dies nicht der Fall. Dazu zeigen wir zunächst ein allgemeines Lemma, das in der Literatur bisweilen auch als das *Bekić*-Prinzip bezeichnet wird.

5.12 Lemma (Auflösen simultaner kleinster Fixpunkte)**.** *Seien A_1, A_2 endliche Mengen und $F_1 : \mathrm{Pot}(A_1) \times \mathrm{Pot}(A_2) \to \mathrm{Pot}(A_1)$ sowie $F_2 : \mathrm{Pot}(A_1) \times \mathrm{Pot}(A_2) \to \mathrm{Pot}(A_2)$ monotone Abbildungen.*

Für jede Menge $R_1 \subseteq A_1$ definieren wir eine Abbildung $F_2^{R_1}$ durch

$$\begin{array}{rcccl} F_2^{R_1} & : & \mathrm{Pot}(A_2) & \to & \mathrm{Pot}(A_2) \\ & & R_2 & \mapsto & F_2(R_1, R_2)\,. \end{array}$$

$F_2^{R_1}$ entsteht also aus F_2, indem die erste Komponente auf R_1 fixiert wird. Sei ferner G_1 die Abbildung

$$\begin{array}{rcccl} G_1 & : & \mathrm{Pot}(A_1) & \to & \mathrm{Pot}(A_1) \\ & & R_1 & \mapsto & F_1\big(R_1, \mathbf{lfp}(F_2^{R_1})\big)\,. \end{array}$$

Dann gilt: $\mathbf{s\text{-}lfp}(F_1, F_2)_1 = \mathbf{lfp}(G_1)$.

Beweis: Sei $S_1^\infty := \mathbf{lfp}(G_1)$ und $(R_1^\infty, R_2^\infty) := \mathbf{s\text{-}lfp}(F_1, F_2)$. Wir müssen zeigen, dass $R_1^\infty = S_1^\infty$.

Wir zeigen zunächst, dass $R_1^\infty \subseteq S_1^\infty$:
Sei dazu für jedes $\alpha \in \mathbb{N}$ (R_1^α, R_2^α) die α-te Stufe der simultanen Fixpunktinduktion für $\vec{F} = (F_1, F_2)$. Per Induktion nach α zeigen wir, dass für jedes $\alpha \in \mathbb{N}$ gilt: $R_1^\alpha \subseteq S_1^\infty$ und $R_2^\alpha \subseteq \mathbf{lfp}(F_2^{S_1^\infty})$. Daraus folgt dann insbesondere, dass $R_1^\infty \subseteq S_1^\infty$.

Der Induktionsanfang für $\alpha = 0$ gilt trivialerweise, da $R_1^0 = R_2^0 = \varnothing$. Für den Induktionsschritt von α nach $\alpha{+}1$ gilt:

$$\begin{array}{rcll} R_1^{\alpha+1} & \stackrel{\text{def}}{=} & F_1(R_1^\alpha, R_2^\alpha) & \\ & \subseteq & F_1\big(S_1^\infty, \mathbf{lfp}(F_2^{S_1^\infty})\big) & \text{nach Ind.ann. und wg. Monotonie von } F_1 \\ & \stackrel{\text{def}}{=} & G_1(S_1^\infty) & \\ & = & S_1^\infty & \text{da } S_1^\infty \text{ ein Fixpunkt von } G_1 \text{ ist} \end{array}$$

und

$$\begin{array}{rcll} R_2^{\alpha+1} & \stackrel{\text{def}}{=} & F_2(R_1^\alpha, R_2^\alpha) & \\ & \subseteq & F_2\big(S_1^\infty, \mathbf{lfp}(F_2^{S_1^\infty})\big) & \text{nach Ind.ann. und wg. Monotonie von } F_2 \\ & = & F_2^{S_1^\infty}(\mathbf{lfp}(F_2^{S_1^\infty})) & \text{nach Definition von } F_2^{S_1^\infty} \\ & = & \mathbf{lfp}(F_2^{S_1^\infty}) & \text{da } \mathbf{lfp}(F_2^{S_1^\infty}) \text{ ein Fixpunkt von } F_2^{S_1^\infty} \text{ ist.} \end{array}$$

Somit sind wir fertig mit dem Induktionsschritt.

Wir zeigen nun, dass $R_1^\infty \supseteq S_1^\infty$*:*
Aus der Definition von $F_2^{R_1^\infty}$ folgt, dass R_2^∞ ein Fixpunkt von $F_2^{R_1^\infty}$ ist. Somit ist

$$\mathbf{lfp}(F_2^{R_1^\infty}) \quad \subseteq \quad R_2^\infty \,.$$

Außerdem ist

$$G_1(R_1^\infty) \;\stackrel{\text{def}}{=}\; F_1(R_1^\infty, \mathbf{lfp}(F_2^{R_1^\infty})) \;\subseteq\; F_1(R_1^\infty, R_2^\infty) \;=\; R_1^\infty \,.$$

Also gilt $G_1(R_1^\infty) \subseteq R_1^\infty$. Aus dem Satz von Knaster und Tarski (Satz 2.15) folgt daher, dass $\mathbf{lfp}(G_1) \subseteq R_1^\infty$. Wegen $S_1^\infty \stackrel{\text{def}}{=} \mathbf{lfp}(G_1)$ folgt also: $R_1^\infty \supseteq S_1^\infty$. □

Mit Hilfe von Lemma 5.12 kann nun leicht folgender Satz bewiesen werden.

5.13 Satz. S-LFP $=$ LFP *auf* Fin.

Beweis: Offensichtlich ist LFP $\leqslant$ S-LFP. Der Beweis von S-LFP $\leqslant$ LFP wird per Induktion über den Formelaufbau geführt. Der einzige interessante Fall sind dabei Formeln $\psi := [\mathbf{lfp}\ R_1 : S](\vec{x})$, wobei nach Induktionsannahme vorausgesetzt werden kann, dass

$$S \;:=\; \left\{ \begin{array}{rcl} R_1(\vec{x}_1) & \leftarrow & \varphi_1(\vec{x}_1, R_1, .., R_m) \\ & \vdots & \\ R_m(\vec{x}_m) & \leftarrow & \varphi_m(\vec{x}_m, R_1, .., R_m) \end{array} \right.$$

ein System von LFP-Formeln ist.

Der Einfachheit halber betrachten wir hier nur den Fall, dass $m = 2$ ist (der allgemeine Fall lässt sich analog lösen). Wir zeigen, dass ψ äquivalent (auf Fin) zu einer LFP-Formel $\Phi(\vec{x}_1)$ ist. Setze dazu

$$\Phi(\vec{x}_1) \quad := \quad \Big[\, \mathbf{lfp}_{R_1,\vec{x}_1}\ \varphi_1\Big(\vec{x}_1,\ R_1,\ R_2(\vec{u})\ /\ [\mathbf{lfp}_{R_2,\vec{x}_2}\, \varphi_2](\vec{u})\Big) \,\Big](\vec{x}_1),$$

wobei $\varphi_1\Big(\vec{x}_1,\ R_1,\ R_2(\vec{u})\ /\ [\mathbf{lfp}_{R_2,\vec{x}_2}\, \varphi_2](\vec{u})\Big)$ diejenige Formel ist, die aus φ_1 entsteht, indem jedes Atom der Form $R_2(\vec{u})$ ersetzt wird durch die Formel $[\mathbf{lfp}_{R_2,\vec{x}_2}\, \varphi_2](\vec{u})$.
Im Folgenden schreiben wir kurz φ^*, um die Formel $\varphi_1\Big(\vec{x}_1,\ R_1,\ R_2(\vec{u})\ /\ [\mathbf{lfp}_{R_2,\vec{x}_2}\, \varphi_2](\vec{u})\Big)$ zu bezeichnen.

Sei $r_1 := ar(R_1)$, $r_2 := ar(R_2)$ und sei $\mathfrak{A}$ eine σ-Struktur. Für jedes $i \in \{1, 2\}$ sei

$$F_i \;:\; \mathrm{Pot}(A^{r_1}) \times \mathrm{Pot}(A^{r_2}) \;\to\; \mathrm{Pot}(A^{r_i})$$

die durch φ_i in $\mathfrak{A}$ gegebene Abbildung. Ferner sei

$$G_1 \;:\; \mathrm{Pot}(A^{r_1}) \;\to\; \mathrm{Pot}(A^{r_1})$$

die durch φ^* in $\mathfrak{A}$ gegebene Abbildung. Offenbar gilt für alle $R \subseteq A^{r_1}$, dass

$$G_1(R) \;=\; F_1(R, \mathbf{lfp}(F_2^R))\,,$$

wobei F_2^R wie in Lemma 5.12 definiert ist. Lemma 5.12 liefert, dass $\mathbf{s\text{-}lfp}(F_1, F_2)_1 = \mathbf{lfp}(G_1)$. Somit ist $\Phi(\vec{x}_1)$ äquivalent (auf Fin) zu $[\mathbf{lfp}\ R_1 : S](\vec{x}_1)$. □

5.14 Bemerkung. Durch die in Satz 5.13 skizzierte Übersetzung von S-LFP-Formeln wird die Stelligkeit der Relationsvariablen nicht verändert. Also sind auch *monadisches* S-LFP (in dem nur Relationsvariablen der Stelligkeit 1 zugelassen sind) und *monadisches* LFP äquivalent auf Fin.

Analog zu S-LFP kann man auch *simultanes* IFP *(*S-IFP*)* und *simultanes* PFP *(*S-PFP*)* definieren. Auch hier können wir zeigen, dass S-IFP = IFP und S-PFP = PFP auf Fin — allerdings muss ein anderer Beweis geführt werden, der die Stelligkeit der Relationsvariablen, über denen ein Fixpunkt gebildet wird, erhöht.

5.15 Satz. S-IFP = IFP *auf* Fin.

Beweis: Der Beweis erfolgt wiederum über den Formelaufbau. Der einzige interessante Fall sind Formeln $\psi(\vec{x}) := [\mathbf{ifp}\ R_1 : S](\vec{x})$, wobei

$$S \;:=\; \begin{cases} R_1(\vec{x}_1) & \leftarrow & \varphi_1(\vec{x}_1, R_1, .., R_m) \\ & \vdots & \\ R_m(\vec{x}_m) & \leftarrow & \varphi_m(\vec{x}_m, R_1, .., R_m) \end{cases}$$

ein System von IFP-Formeln ist.

Sei $\mathfrak{A}$ eine endliche Struktur. Der Einfachheit halber nehmen wir an, dass die Relationsvariablen $R_1, .., R_m$ alle dieselbe Stelligkeit r besitzen und dass $|A| \geqslant m$ ist (der allgemeine Fall lässt sich analog lösen). Die Idee ist nun, an Stelle der m Relationen $R_1, .., R_m$ eine einzige Relation R der Stelligkeit $r{+}1$ induktiv zu erzeugen, so dass die letzte Komponente von R angibt zu welcher der Relationen $R_1, .., R_m$ das jeweilige Tupel gehört.

Zur Konstruktion einer IFP-Formel, die äquivalent zu $\psi(\vec{x}) := [\mathbf{ifp}\ R_1 : S](\vec{x})$ ist, seien $c_1, .., c_m$ paarweise verschiedene Variablen erster Stufe, die in keiner der Formeln $\varphi_1, .., \varphi_m$ vorkommen. Für jedes $j \in \{1, .., m\}$ sei $\varphi_j^*(\vec{x}, R)$ diejenige Formel, die aus $\varphi_j(\vec{x}, R_1, .., R_m)$ entsteht, indem für jedes $i \in \{1, .., m\}$ jedes Vorkommen eines Atoms der Form $R_i(\vec{u})$ durch das Atom $R(\vec{u}, c_i)$ ersetzt wird. Die letzte Komponente in R spielt also die Rolle eines Zählers oder Markers, der angibt, zu welcher der Relationen $R_1, .., R_m$ das Tupel $\vec{u}$ gehören soll.

Man sieht nun leicht, dass die S-IFP-Formel $\psi(\vec{x}) := [\mathbf{ifp}\ R_1 : S](\vec{x})$ äquivalent (auf der Klasse aller Strukturen $\mathfrak{A}$ mit $|A| \geqslant m$) ist zur IFP-Formel $\Psi(\vec{x}) :=$

$$\exists c_1 \cdots \exists c_m \Big(\Big(\bigwedge_{1 \leqslant i < j \leqslant m} \neg c_i{=}c_j \Big) \;\wedge\; \big[\, \mathbf{ifp}_{R,\vec{x},c} \bigvee_{j=1}^{m} \big(c{=}c_j \;\wedge\; \varphi_j^*(\vec{x}) \big) \big] (\vec{x}, c_1) \Big)\,.$$

□

Mit dem gleichen Beweis zeigt man auch S-PFP $=$ PFP. Selbstverständlich funktioniert dieser Beweis auch für die Logik LFP. Zusammen mit Lemma 5.12 folgt daraus:

5.16 Korollar. *Jede Formel $\varphi \in$ LFP, in der alle* $\mathbf{lfp}$*-Operatoren nur positiv vorkommen, ist äquivalent zu einer Formel $\psi \in$ LFP mit nur einem* $\mathbf{lfp}$*-Operator.*

Beweisidee: Verschachtelte Fixpunkte werden mit Hilfe von Lemma 5.12 zu Systemen simultaner Fixpunkte aufgelöst. Diese Systeme simultaner Fixpunkte werden dann mit Hilfe der im Beweis von Satz 5.15 beschriebenen Methode zu einem einzigen Fixpunkt aufgelöst. Etwas Sorgfalt ist hier bei Booleschen Kombinationen geboten, wenn deren Teilformeln jeweils selbst wieder Fixpunktoperatoren enthalten. Details: *Übung*. □

5.3 Die Stage-Comparison Methode

In diesem Kapitel wird die so genannte *Stage-Comparison Methode* vorgestellt, die sich oft als sehr nützlich im Zusammenhang mit kleinsten und inflationären Fixpunktlogiken herausstellt. In gewisser Hinsicht erlaubt sie, Eigenschaften von Formeln, die man per Induktion über die Induktionsstufen beweisen kann, in den Logiken selbst zu definieren. Die Methode hat verschiedene wichtige Anwendungen, insbesondere werden wir sie verwenden, um die Äquivalenz von IFP und LFP auf Fin nachzuweisen.

Zunächst jedoch einige Vorbemerkungen. Sei $\varphi(\vec{x}, R)$ eine LFP-Formel oder eine IFP-Formel. Gemäß der Definition inflationärer Fixpunkte ist offensichtlich der über die Formel φ gebildete inflationäre Fixpunkt identisch mit dem über der Formel $\big(R(\vec{x}) \vee \varphi\big)$ gebildeten inflationären Fixpunkt. Falls φ monoton in R ist, so sind natürlich auch die über φ bzw. über $\big(R(\vec{x}) \vee \varphi\big)$ gebildeten kleinsten Fixpunkte identisch. Wir können daher immer o.B.d.A. annehmen, dass alle Formeln φ, die in Fixpunktoperatoren $[\mathbf{ifp}_{R,\vec{x}}\,\varphi](\vec{t})$ oder $[\mathbf{lfp}_{R,\vec{x}}\,\varphi](\vec{t})$ vorkommen, die Form $\varphi = \big(R(\vec{x}) \vee \varphi'\big)$ haben. Dies wird später noch wichtig werden.

5.17 Notation. Sei σ eine Signatur und sei R eine Relationsvariable. Sei außerdem $\varphi(\vec{x}, R)$ eine Formel.

(a) Ist $\psi(\vec{x})$ eine Formel, so schreiben wir $\varphi\big(\vec{x},\ R(\vec{u})/\psi(\vec{u})\big)$ für die Formel, die man aus φ erhält, wenn jedes Vorkommen eines Atoms der Form $R(\vec{u})$ (für beliebige Tupel $\vec{u}$ aus Variablen erster Stufe und/oder Konstantensymbolen aus σ) durch die Formel $\psi(\vec{u})$ ersetzt wird.

(b) Sind $\psi_p(\vec{x})$ und $\psi_n(\vec{x})$ Formeln, so schreiben wir

$$\varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u})\,/\,\psi_p(\vec{u}),\ \textit{neg}\ R(\vec{u})\,/\,\psi_n(\vec{u})\big)$$

für die Formel, die man aus φ erhält, wenn man

- jedes *positive* Vorkommen eines Atoms der Form $R(\vec{u})$ durch die Formel $\psi_p(\vec{u})$ ersetzt, und

- jedes *negative* Vorkommen eines Atoms der Form $R(\vec{u})$ durch $\neg\psi_n(\vec{u})$ ersetzt.

Man beachte hier, dass auch bei negativen Vorkommen von $R(\vec{u})$, z.B. als $\neg R(\vec{u})$, nur das Atom $R(\vec{u})$ ersetzt wird, nicht aber das gesamte $\neg R(\vec{u})$. Aus $\neg R(\vec{u})$ würde also $\neg\neg\psi_n(\vec{u})$.

Im weiteren Verlauf des Kapitels werden wir für $\psi_p(\vec{x})$ und $\psi_n(\vec{x})$ Formeln konstruieren, die zu $R(\vec{x})$ bzw. $\neg R(\vec{x})$ äquivalent sind. Dann ist natürlich φ äquivalent zu $\varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u})\,/\,\psi_p(\vec{u}),\ \textit{neg}\ R(\vec{u})\,/\,\psi_n(\vec{u})\big)$.

Wir führen nun den zentralen Begriff der Stage-Comparison-Methode ein: die so genannten *Stage-Comparison-Relationen*, die von Yannis Moschovakis in den 1970er Jahren eingeführt wurden, aber auch schon früher in anderer Form im Rahmen der Rekursionstheorie verwendet wurden.

5.18 Definition (Stage-Comparison-Relationen)**.** Sei R ein k-stelliges Relationssymbol, sei $\varphi(R,\vec{x})$ eine Formel (z.B. aus FO, LFP oder IFP), und sei $\mathfrak{A}$ eine endliche Struktur.

(a) Der *Rang* $|\vec{a}|_\varphi$ *eines Tupels* $\vec{a}\in A^k$ *bzgl.* φ ist definiert als

$$|\vec{a}|_\varphi := \begin{cases} \min\{\alpha\in\mathbb{N} : \vec{a}\in R^\alpha\} & \text{falls } \vec{a}\in \mathbf{ifp}(F_{\varphi,\mathfrak{A}}) \\ \infty & \text{sonst.}\end{cases}$$

R^α bezeichnet hier die α-te Stufe des durch φ in $\mathfrak{A}$ definierten inflationären Fixpunkts.

Der Rang $|\vec{a}|_\varphi$ gibt also an, in der wievielten Induktionsstufe das Tupel $\vec{a}$ "zum Fixpunkt hinzukommt".

Ist φ positiv in R, so kann an Stelle des inflationären auch der kleinste Fixpunkt verwendet werden — dies ergibt natürlich den gleichen Rang.

(b) Die *Stage-Comparison-Relationen* $\leqslant_\varphi$ und $\prec_\varphi$ sind wie folgt definiert:
Für alle $\vec{a}\in A^k$ und alle $\vec{b}\in A^k$ gilt

$$\vec{a}\leqslant_\varphi \vec{b} \qquad \text{falls} \quad |\vec{a}|_\varphi \leqslant |\vec{b}|_\varphi < \infty \qquad \text{insbes: } \vec{a},\vec{b}\in R^\infty$$

und

$$\vec{a}\prec_\varphi \vec{b} \qquad \text{falls} \quad |\vec{a}|_\varphi < |\vec{b}|_\varphi \text{ und } |\vec{a}|_\varphi\neq\infty \qquad \text{insbes: } \vec{a}\in R^\infty,\ \text{aber eventuell } \vec{b}\notin R^\infty.$$

5.19 Satz. *Sei $\varphi(R,\vec{x})$ eine* IFP*-Formel. Dann sind die Stage-Comparison-Relationen $\leqslant_\varphi$ und $\prec_\varphi$ in* IFP *definierbar.*

Beweis: Sei $\varphi(R,\vec{x})$ eine IFP-Formel. O.B.d.A. nehmen wir an, dass φ die Form $\big(R(\vec{x})\vee\varphi'\big)$ hat. Wir zeigen, dass das Paar $(\leqslant_\varphi,\prec_\varphi)$ durch den simultanen Fixpunkt des Systems

$$S := \begin{cases} \vec{x}\leqslant\vec{y} & \leftarrow \quad \varphi\big(\vec{x},\ R(\vec{u})\,/\,\vec{u}\prec\vec{y}\big)\ \wedge\ \varphi\big(\vec{y},\ R(\vec{u})\,/\,\vec{u}\prec\vec{y}\big) \\ \vec{x}\prec\vec{y} & \leftarrow \quad \varphi\big(\vec{x},\ R(\vec{u})\,/\,\vec{u}\prec\vec{x}\big)\ \wedge\ \neg\varphi\big(\vec{y},\ R(\vec{u})\,/\,\vec{u}\prec\vec{x}\big)\end{cases}$$

definiert wird. Hierbei sind $\leqslant$ und $\prec$ zwei $2k$-stellige Relationssymbole, die wir der Übersichtlichkeit halber in Infix-Notation schreiben.

Behauptung: *Sei $\mathfrak{A}$ eine endliche Struktur. Für jedes $\alpha \in \mathbb{N}$ sei $(\leqslant^\alpha, \prec^\alpha)$ die α-te Stufe der simultanen Induktion über S in $\mathfrak{A}$. Dann gilt für alle $\alpha \in \mathbb{N}$ und alle $\vec{a}, \vec{b} \in A^k$:*

(i) $(\vec{a}, \vec{b}) \in \leqslant^\alpha \iff |\vec{b}|_\varphi \leqslant \alpha$ *und* $|\vec{a}|_\varphi \leqslant |\vec{b}|_\varphi$

(ii) $(\vec{a}, \vec{b}) \in \prec^\alpha \iff |\vec{a}|_\varphi \leqslant \alpha$ *und* $|\vec{a}|_\varphi < |\vec{b}|_\varphi$.

Aus dieser Behauptung folgt natürlich sofort die Aussage von Satz 5.19, denn laut der Behauptung gilt

$$\vec{a} \leqslant_\varphi \vec{b} \iff \mathfrak{A} \models \big([\mathbf{ifp}\ \leqslant : S](\vec{x}, \vec{y})\big)\,[\vec{a}, \vec{b}]$$

und

$$\vec{a} \prec_\varphi \vec{b} \iff \mathfrak{A} \models \big([\mathbf{ifp}\ \prec : S](\vec{x}, \vec{y})\big)\,[\vec{a}, \vec{b}]\,.$$

Wir führen den Beweis der Behauptung per Induktion nach α. Für $\alpha = 0$ ist die Behauptung klar, da es keine Elemente vom Rang 0 gibt, und da die 0-te Stufe der Induktion die leere Menge ist.

Sei also $\alpha \geqslant 1$, so dass die Behauptung schon für $\alpha{-}1$ bewiesen ist. Wir zeigen zunächst, dass *(i)* für α gilt.

Dazu betrachte zunächst Tupel $\vec{b}$ vom Rang $\xi := |\vec{b}|_\varphi \leqslant \alpha$. Insbesondere ist $\xi \geqslant 1$ (da es keine Elemente vom Rang 0 gibt), und laut Induktionsannahme gilt

$$\big\{\, \vec{u} \in A^k \,:\, \vec{u} \prec^{\alpha-1} \vec{b} \,\big\} \quad = \quad \big\{\, \vec{u} \in A^k \,:\, |\vec{u}|_\varphi < \xi \,\big\}\,.$$

Also wird die Formel $\varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big)$ in $\mathfrak{A}$ erfüllt, wenn $\prec$ mit $\prec^{\alpha-1}$ und $\vec{y}$ mit $\vec{b}$ belegt wird. D.h. es gilt:

$$\big(\mathfrak{A}, \prec^{\alpha-1}, \vec{b}\big) \models \varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big)\,.$$

Wird nun $\prec$ mit $\prec^{\alpha-1}$, $\vec{y}$ mit $\vec{b}$ und $\vec{x}$ durch ein *beliebiges* Tupel $\vec{a} \in A^k$ belegt, so gilt

$$\big(\mathfrak{A}, \prec^{\alpha-1}, \vec{a}, \vec{b}\big) \models \varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big) \quad \iff \quad |\vec{a}|_\varphi \leqslant \xi \stackrel{\text{def}}{=} |\vec{b}|_\varphi\,.$$

Insgesamt gilt daher für alle $\vec{b}$ mit $|\vec{b}|_\varphi \leqslant \alpha$ und für alle $\vec{a} \in A^k$, dass $(\vec{a}, \vec{b}) \in \leqslant^\alpha \iff |\vec{a}|_\varphi \leqslant |\vec{b}|_\varphi$.

Ist hingegen $\vec{b}$ ein Tupel mit $\xi := |\vec{b}|_\varphi > \alpha$, so ist laut Induktionsannahme

$$\big\{\, \vec{u} \in A^k \,:\, \vec{u} \prec^{\alpha-1} \vec{b} \,\big\} \quad = \quad \big\{\, \vec{u} \in A^k \,:\, |\vec{u}|_\varphi \leqslant \alpha{-}1 \,\big\}\,.$$

Wegen $|\vec{b}|_\varphi \geqslant \alpha{+}1$ kann kann daher die Formel $\varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big)$ nicht in $\mathfrak{A}$ erfüllt sein, wenn $\vec{y}$ mit $\vec{b}$ und $\prec$ mit $\prec^{\alpha-1}$ belegt ist. Deshalb gilt für alle $\vec{b} \in A^k$ mit $|\vec{b}|_\varphi > \alpha$

und für alle $\vec{a} \in A^k$, dass $(\vec{a}, \vec{b}) \notin \leqslant^\alpha$.
Insgesamt liefert dies Teil *(i)* der Behauptung.

Zum Beweis von Teil *(ii)* betrachte zunächst Tupel $\vec{a} \in A^k$ vom Rang $\xi := |\vec{a}|_\varphi \leqslant \alpha$. Laut Induktionsannahme gilt

$$\left\{ \vec{u} \in A^k \,:\, \vec{u} \prec^{\alpha-1} \vec{a} \right\} \quad = \quad \left\{ \vec{u} \in A^k \,:\, |\vec{u}|_\varphi < \xi \right\},$$

und die Formel $\varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big)$ wird in $\mathfrak{A}$ erfüllt, wenn $\vec{x}$ mit $\vec{a}$ und $\prec$ mit $\prec^{\alpha-1}$ belegt wird. D.h. es gilt:

$$\big(\mathfrak{A}, \prec^{\alpha-1}, \vec{a}\big) \;\models\; \varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big)\,.$$

Wird nun $\prec$ mit $\prec^{\alpha-1}$, $\vec{x}$ mit $\vec{a}$ und $\vec{y}$ durch ein *beliebiges* $\vec{b} \in A^k$ belegt, so gilt

$$\big(\mathfrak{A}, \prec^{\alpha-1}, \vec{a}, \vec{b}\big) \;\models\; \neg\varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big) \quad \iff \quad |\vec{b}|_\varphi \not\leqslant \xi \stackrel{\text{def}}{=} |\vec{a}|_\varphi\,.$$

Insgesamt gilt daher für alle $\vec{a}$ mit $|\vec{a}|_\varphi \leqslant \alpha$ und für alle $\vec{b} \in A^k$, dass $(\vec{a}, \vec{b}) \in \prec^\alpha \iff |\vec{a}|_\varphi < |\vec{b}|_\varphi$.

Ist hingegen $\vec{a}$ ein Tupel mit $\xi := |\vec{a}|_\varphi > \alpha$, so ist laut Induktionsannahme

$$\left\{ \vec{u} \in A^k \,:\, \vec{u} \prec^{\alpha-1} \vec{a} \right\} \quad = \quad \left\{ \vec{u} \in A^k \,:\, |\vec{u}|_\varphi \leqslant \alpha{-}1 \right\}.$$

Wegen $|\vec{a}|_\varphi \geqslant \alpha{+}1$ kann kann daher die Formel $\varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big)$ nicht in $\mathfrak{A}$ erfüllt sein, wenn $\vec{x}$ mit $\vec{a}$ und $\prec$ mit $\prec^{\alpha-1}$ belegt wird. Deshalb gilt für alle $\vec{a}$ mit $|\vec{a}|_\varphi > \alpha$ und für alle $\vec{b} \in A^k$, dass $(\vec{a}, \vec{b}) \notin \prec^\alpha$.
Insgesamt liefert dies Teil *(ii)* der Behauptung. □

Aus den Stage-Comparison-Relationen kann nun sofort der inflationäre Fixpunkt einer Formel definiert werden:

5.20 Lemma.
Für jede IFP*-Formel* $\varphi(R, \vec{x})$ *ist* $[\mathbf{ifp}_{R,\vec{x}}\,\varphi](\vec{x})$ *äquivalent zu* $\vec{x} \leqslant_\varphi \vec{x}$ *auf* Fin.

Beweis: klar. □

Als nächstes werden wir zeigen, dass die inflationären Stage-Comparison-Relationen einer beliebigen LFP-Formel $\varphi(R, \vec{x})$, die nicht positiv in R zu sein braucht, bereits in LFP definiert werden können. Mit Hilfe des vorhergehenden Lemmas folgt dann leicht, dass LFP und IFP dieselbe Ausdrucksstärke auf Fin besitzen.

5.21 Satz. *Sei* $\varphi(R, \vec{x})$ *eine* LFP*-Formel, nicht unbedingt positiv in* R*. Dann sind die Stage-Comparison-Relationen* $\leqslant_\varphi$ *und* $\prec_\varphi$ *in* LFP *definierbar.*

Beweis: O.B.d.A. nehmen wir an, dass φ die Form $\big(R(\vec{x}) \vee \varphi'\big)$ hat. Zunächst betrachten wir noch einmal das System

$$S \;:=\; \begin{cases} \vec{x} \leqslant \vec{y} \;\leftarrow\; \varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big) \;\wedge\; \varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big) \\ \vec{x} \prec \vec{y} \;\leftarrow\; \varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big) \;\wedge\; \neg\varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big) \end{cases}$$

aus dem Beweis von Satz 5.19. Ziel ist es, dieses System in ein äquivalentes System von Formeln umzuschreiben, die *positiv* in $\prec$ und $\leqslant$ sind. Das Problem dabei ist, dass überall, wo in φ die Variable R negativ vorkommt, in $\varphi\big(\vec{x},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{y}\big)$ auch die Variable $\prec$ negativ vorkommt. Entsprechend kommt $\prec$ überall dort in $\neg\,\varphi\big(\vec{y},\ R(\vec{u}) \,/\, \vec{u} \prec \vec{x}\big)$ negativ vor, wo R positiv steht. Wir brauchen also eine Definition des Komplements R^c von R durch Formeln, die positiv in $\prec$ und $\leqslant$ sind. Dazu nutzen wir aus, dass für jede Induktionsstufe R^α mit $\alpha \geqslant 1$ gilt:

$$(R^\alpha)^c \quad = \quad \{\vec{a} \in A^k \ :\ \vec{c} \prec \vec{a}\}\,,$$

wobei $\vec{c}$ ein beliebiges Tupel vom Rang α ist.

Wir nehmen zunächst an, dass wir die Relationen $\leqslant$ und $\prec$ schon bis zu einer Stufe $\beta > 0$ definiert hätten, d.h. es gilt

$$\begin{aligned} & \vec{a} \leqslant^\beta \vec{b} \quad \iff \quad |\vec{a}|_\varphi \leqslant |\vec{b}|_\varphi \leqslant \beta \\ \text{und} \qquad & \vec{a} \prec^\beta \vec{b} \quad \iff \quad |\vec{a}|_\varphi \leqslant \beta \ \text{ und } \ |\vec{a}|_\varphi < |\vec{b}|_\varphi\,. \end{aligned}$$

Wir konstruieren nun eine Formel $\psi_\leqslant$, die positiv in $\leqslant$ und in $\prec$ ist, so dass für alle $\vec{a}, \vec{b} \in A^k$ gilt:

$$(*) \qquad (\mathfrak{A}, \leqslant^\beta, <^\beta) \models \psi_\leqslant[\vec{a}, \vec{b}] \quad \iff \quad 1 < |\vec{b}|_\varphi \leqslant \beta + 1 \ \text{ und } \ |\vec{a}|_\varphi \leqslant |\vec{b}|_\varphi\,.$$

Dazu setzen wir

$$\begin{aligned} \psi_\leqslant(\vec{x}, \vec{y}) \ := \quad \exists \vec{z} \Big(\ & \vec{z} \prec \vec{y} \ \wedge\ \varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big) \ \wedge \\ & \varphi\big(\vec{y},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big) \ \Big). \end{aligned}$$

Man sieht leicht, dass die Formel $\psi_\leqslant$ positiv in $\leqslant$ und in $\prec$ ist. Wir prüfen nun nach, dass $(*)$ erfüllt ist: Ist $\vec{b}$ ein Tupel, so dass $1 < |\vec{b}|_\varphi \stackrel{\text{def}}{=} \xi \leqslant \beta + 1$, so können wir für $\vec{z}$ ein Tupel $\vec{c}$ vom Rang $\xi - 1$ wählen und wissen, dass dann $\vec{c} \prec^\beta \vec{b}$ erfüllt ist. Außerdem erfüllt $\vec{b}$ die Formel $\varphi\big(\vec{y},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big)$; und $\vec{a} \in A^k$ erfüllt $\varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big)$ genau dann, wenn $|\vec{a}|_\varphi \leqslant \xi = |\vec{b}|_\varphi$.
Ist andererseits $\vec{b}$ ein Tupel mit $|\vec{b}|_\varphi \stackrel{\text{def}}{=} \xi > \beta + 1$, so gibt es kein Tupel $\vec{c}$ für $\vec{z}$ mit $\vec{c} \prec^\beta \vec{b}$, so dass $\vec{b}$ die Formel $\varphi\big(\vec{y},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big)$ erfüllt (denn dann wäre $\vec{b}$ vom Rang $\leqslant \beta + 1$).

Auf ähnliche Art konstruieren wir nun eine Formel $\psi_\prec$, die positiv in $\leqslant$ und in $\prec$ ist, so dass für alle $\vec{a}, \vec{b} \in A^k$ gilt:

$$(**) \qquad (\mathfrak{A}, \leqslant^\beta, <^\beta) \models \psi_\prec[\vec{a}, \vec{b}] \quad \iff \quad |\vec{a}|_\varphi \leqslant \beta + 1 \ \text{ und } \ |\vec{a}|_\varphi < |\vec{b}|_\varphi\,.$$

Dazu setzen wir

$$\begin{aligned} \psi_\prec(\vec{x}, \vec{y}) \ := \quad \exists \vec{z} \Big(\ & \vec{z} \leqslant \vec{z} \ \wedge\ \varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big) \ \wedge \\ & \neg\ \varphi\big(\vec{y},\ \textit{pos}\ R(\vec{u}) \,/\, \neg\vec{z} \prec \vec{u},\ \textit{neg}\ R(\vec{u}) \,/\, \neg\vec{u} \leqslant \vec{z}\big) \Big). \end{aligned}$$

Man sieht leicht, dass die Formel $\psi_\prec$ positiv in $\leqslant$ und in $\prec$ ist. Wir prüfen nun nach, dass $(**)$ erfüllt ist: Ist $\vec{a}$ ein Tupel, so dass $1 < |\vec{a}|_\varphi \stackrel{\text{def}}{=} \xi \leqslant \beta + 1$, so können wir für $\vec{z}$ ein Tupel $\vec{c}$ vom Rang $\xi{-}1$ wählen und wissen, dass dann $\vec{c} \leqslant^\beta \vec{c}$ erfüllt ist. Außerdem erfüllt $\vec{a}$ die Formel $\varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big)$; und $\vec{b} \in A^k$ erfüllt $\neg\,\varphi\big(\vec{y},\ \textit{pos}\ R(\vec{u}) \,/\, \neg\vec{z} \prec \vec{u},\ \textit{neg}\ R(\vec{u}) \,/\, \neg\vec{u} \leqslant \vec{z}\big)$ genau dann, wenn $|\vec{b}|_\varphi > \xi{+}1 \stackrel{\text{def}}{=} |\vec{a}|_\varphi$. Ist andererseits $\vec{a}$ ein Tupel mit $|\vec{a}|_\varphi \stackrel{\text{def}}{=} \xi > \beta + 1$, so gibt es kein Tupel $\vec{c}$ für $\vec{z}$ mit $\vec{c} \leqslant^\beta \vec{c}$, so dass $\vec{a}$ die Formel $\varphi\big(\vec{x},\ \textit{pos}\ R(\vec{u}) \,/\, \vec{u} \leqslant \vec{z},\ \textit{neg}\ R(\vec{u}) \,/\, \vec{z} \prec \vec{u}\big)$ erfüllt (denn dann wäre $\vec{a}$ vom Rang $\leqslant \beta{+}1$).

Wir haben jetzt also Formeln $\psi_\leqslant$ und $\psi_\prec$, die für eine gegebene Stufe $\beta > 1$ der Induktion über $\leqslant$ und $\prec$ die nächste Stufe $\beta + 1$ definieren. Wir brauchen nun also nur noch Formeln, die $\leqslant^1, \prec^1$ definieren. Dazu beachte, dass für alle $\vec{a}, \vec{b} \in A^k$ gilt:

- $(\vec{a}, \vec{b}) \in \leqslant^1 \iff \mathfrak{A} \models \big(\varphi(\vec{x}, \varnothing) \wedge \varphi(\vec{y}, \varnothing)\big)[\vec{a}, \vec{b}]$
- $(\vec{a}, \vec{b}) \in \prec^1 \iff \mathfrak{A} \models \big(\varphi(\vec{x}, \varnothing) \wedge \neg\varphi(\vec{y}, \varnothing)\big)[\vec{a}, \vec{b}]$.

Hierbei bedeutet $\varphi(\vec{x}, \varnothing)$, dass in φ alle Vorkommen eines Atoms $R(\vec{u})$ durch eine Formel ersetzt werden, die niemals erfüllt ist (z.B. $\exists x \neg x{=}x$) und analog für die anderen Formeln.

Fügen wir jetzt beide Teile zusammen so erhalten wir folgendes System

$$T := \begin{cases} \vec{x} \leqslant \vec{y} & \leftarrow \quad \big(\varphi(\vec{x}, \varnothing) \wedge \ \varphi(\vec{y}, \varnothing)\big) \ \vee\ \psi_\leqslant(\vec{x}, \vec{y}) \\ \vec{x} \prec \vec{y} & \leftarrow \quad \big(\varphi(\vec{x}, \varnothing) \wedge \neg\varphi(\vec{y}, \varnothing)\big) \ \vee\ \psi_\prec(\vec{x}, \vec{y})\,. \end{cases}$$

Nach Konstruktion sind alle Formeln in T positiv in $\prec$ und in $\leqslant$, und es gilt für alle endlichen Strukturen $\mathfrak{A}$ und alle Tupel $\vec{a}, \vec{b} \in A^k$:

$$\begin{aligned} \mathfrak{A} \ &\models\ \big(\,[\mathbf{lfp}\ \leqslant\!\!: T](\vec{x}, \vec{y})\,\big)[\vec{a}, \vec{b}] \qquad \iff \qquad |\vec{a}|_\varphi \leqslant |\vec{b}|_\varphi \neq \infty \\ \mathfrak{A} \ &\models\ \big(\,[\mathbf{lfp}\ \prec\!\!: T](\vec{x}, \vec{y})\,\big)[\vec{a}, \vec{b}] \qquad \iff \qquad \vec{a} \in R^\infty \text{ und } |\vec{a}|_\varphi < |\vec{b}|_\varphi\,. \end{aligned}$$

□

Zusammen mit Lemma 5.20 folgt das nächste Theorem mittels einer leichten Induktion über den Formelaufbau, bei der **ifp**-Operatoren von innen nach außen in entsprechende LFP-Formeln umgewandelt werden.

5.22 Theorem (Gurevich, Shelah, 1986)**.** IFP $=$ LFP *auf* Fin,
das heißt, jede Formel $\varphi \in$ IFP ist auf der Klasse aller endlichen Strukturen äquivalent zu einer Formel $\varphi^* \in$ LFP.

Beweis: "$\geqslant$" wurde bereits in Proposition 2.37 gezeigt.
"$\leqslant$" folgt per Induktion nach dem Aufbau von IFP-Formeln. Einziger interessanter Fall: Gegeben sei eine IFP-Formel ψ der Form $[\mathbf{ifp}_{R,\vec{x}}\,\varphi(R, \vec{x})](\vec{t})$. Gemäß Induktionsannahme können wir o.B.d.A. annehmen, dass $\varphi(R, \vec{x})$ eine LFP-Formel ist (die nicht unbedingt positiv in R sein muss). Aus Lemma 5.20 wissen wir, dass $[\mathbf{ifp}_{R,\vec{x}}\,\varphi(R, \vec{x})](\vec{t})$ äquivalent ist zu

$\vec{t} \leqslant_\varphi \vec{t}$. Satz 5.21 besagt, dass die Stage-Comparison-Relation $\leqslant_\varphi$ in LFP definiert werden kann. Somit ist die Formel $[\mathbf{ifp}_{R,\vec{x}}\, \varphi(R,\vec{x})](\vec{t})$ äquivalent (auf Fin) zu einer LFP-Formel. □

5.23 Bemerkung. Theorem 5.22 gilt auch im Unendlichen (d.h. für die Klasse *aller* Strukturen), ist aber aufwändiger zu beweisen — dies wurde in Stephan Kreutzers Dissertation (2002) gezeigt.