Vorlesung Logik und Komplexität

Sommersemester 2005

Stephan Kreutzer und Nicole Schweikardt
Institut für Informatik
Humboldt-Universität zu Berlin

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 $\mathsf{Artikel}$ mit den Attributen Titel, Autor, $\mathsf{Konferenz}$, Jahr.

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

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

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

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

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

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

Als nächstes wollen wir folgende Anfrage stellen:

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

Formal lässt sich diese Anfrage folgendermaßen ausdrücken:

Relation Artikel:

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

Tabelle 0.1: Beispiel-Relation in einer Bibliotheks-Datenbank

In der Datenbankanfragesprache SQL:

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

In der Logik erster Stufe (FO):

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

Unsere dritte Beispiel-Anfrage lautet:

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

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

In SQL:

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

In FO:

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

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

Als letztes Beispiel betrachten wir folgende Anfrage:

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

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

In SQL:

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

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

Wichtige Fragestellungen bzgl. Logik und Datenbanken:

Gegeben eine Datenbank-Anfragesprache bzw. Logik,

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

0.2 Logiken zur Beschreibung von Berechnungsproblemen

0.2 Beispiel (3-Färbbarkeit von Graphen).

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

Somit gilt:

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

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

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

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

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

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

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

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

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

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

Wichtige Fragestellungen bzgl. Logik und Komplexitätstheorie:

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

0.3 Logiken zur Hardware- und Prozess-Spezifikation und Verifikation

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

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

Wichtige Fragestellungen bzgl. Spezifikation und Verifikation:

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

0.4 Aufbau der Vorlesung

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

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

0.5 Literatur

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

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

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

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

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

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

1 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.

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

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

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

1.1 Logik erster Stufe (FO)

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

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

verwiesen.

1.1 Definition (Signatur, Struktur).

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

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

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

(b) Eine σ-*Struktur*

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

besteht aus

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

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

1.2 Beispiele.

- Ein Graph $G = (V, E^G)$ ist eine $\{E\}$-Sturktur, wobei E ein 2-stelliges Relationssymbol ist.
- $\mathfrak{N} := (\mathbb{N}, <^{\mathfrak{N}}, +^{\mathfrak{N}}, \times^{\mathfrak{N}}, 0^{\mathfrak{N}}, 1^{\mathfrak{N}})$ ist eine $\{<, +, \times, 0, 1\}$-Struktur, wobei $<$ ein 2-stelliges Relationssymbol, $+$ und $\times$ 3-stellige Relationssymbole und 0 und 1 Konstantensymbole sind. Hier sollen $0^{\mathfrak{N}}$ und $1^{\mathfrak{N}}$ die natürlichen Zahlen 0 und 1 bezeichnen, $<^{\mathfrak{N}}$ soll die natürliche lineare Ordnung auf $\mathbb{N}$ bezeichnen, und $+^{\mathfrak{N}}$ bzw. $\times^{\mathfrak{N}}$ die Graphen der Addition bzw. Multiplikation, d.h.
$$+^{\mathfrak{N}} \; := \; \{(a,b,c) \in \mathbb{N}^3 \mid a+b=c\} \qquad \text{und} \qquad \times^{\mathfrak{N}} \; := \; \{(a,b,c) \in \mathbb{N}^3 \mid a \cdot b = c\}.$$
- Unsere Bibliotheksdatenbank aus Beispiel 0.1 ist eine $\{\mathsf{Artikel}\}$-Struktur, wobei $\mathsf{Artikel}$ ein 4-stelliges Relationssymbol ist. Das Universum dieser Struktur ist eine Teilmenge der Menge aller Worte über den Zeichen des ASCII-Alphabets.

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

(a) Ein *Homomorphismus* $h : \mathfrak{A} \to \mathfrak{B}$ ist eine Abbildung $h : A \to B$, für die gilt:

- $h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$, für alle Konstantensymbole $c \in \sigma$,
- $h(R^{\mathfrak{A}}) \subseteq R^{\mathfrak{B}}$, für alle Relationssymbole $R \in \sigma$.[1]

(b) Ein Homomorphismus $h : \mathfrak{A} \to \mathfrak{B}$ heißt *Isomorphismus*, falls

- h eine *Bijektion* von A nach B ist und
- $\vec{a} \in R^{\mathfrak{A}} \iff h(\vec{a}) \in R^{\mathfrak{B}}$,
für alle Relationssymbole $R \in \sigma$ und alle $\vec{a} \in A^{ar(R)}$.

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

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

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

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

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

(b) Die Formelmenge FO[σ] ist induktiv wie folgt definiert:

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

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

(BC) Sind $\psi, \varphi_1, \varphi_2$ Formeln in FO[σ], so gehören auch die folgenden Formeln zu FO[σ]:
- $\neg\psi$ (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 FO[σ] und $x \in \text{Var}_1$, so gehören auch folgende Formeln zu FO[σ]:
- $\exists x\, \psi$ (Existenzquantor)
- $\forall x\, \psi$ (Allquantor).

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

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

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

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

1.7 Definition (Semantik der Logik erster Stufe). Sei σ eine Signatur, $\mathfrak{I} = (\mathfrak{A}, \beta)$ eine σ-Interpretation und φ eine FO[σ]-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 $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\; \big(\beta(v_1), \ldots, \beta(v_k)\big) \in R^{\mathfrak{A}}$.
- 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 $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\; \big(\, \mathfrak{I} \models \varphi_1 \text{ oder } \mathfrak{I} \models \varphi_2 \,\big)$.
  - $(\varphi_1 \wedge \varphi_2)$, so $\mathfrak{I} \models \varphi \;:\Longleftrightarrow\; \big(\, \mathfrak{I} \models \varphi_1 \text{ und } \mathfrak{I} \models \varphi_2 \,\big)$.

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

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

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

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

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

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

- 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*$(\varphi) = \emptyset$.

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

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

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

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

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

(e) Für eine Signatur $\sigma = \{R_1, .\,., R_k, c_1, .\,., c_\ell\}$ schreiben wir oft FO$[R_1, .\,., R_k, c_1, .\,., c_\ell]$ an Stelle von FO$[\{R_1, .\,., R_k, c_1, .\,., c_\ell\}]$, 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) $\mathrm{Th}_{\mathscr{L}}(\mathfrak{A}) := \{\varphi \in \mathscr{L} \mid \mathfrak{A} \models \varphi\}$ ist die *$\mathscr{L}$-Theorie* der Struktur $\mathfrak{A}$.

(b) $\mathrm{Mod}_{\mathsf{K}}(\psi) := \{\mathfrak{B} \in \mathsf{K} \mid \mathfrak{B} \models \psi\}$ 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 *nicht-deterministische Turing-Maschine* (kurz: NTM) ist ein 6-Tupel

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

bestehend aus

- einer endlichen Menge Q von Zuständen,
- einem Arbeitsalphabet Γ mit ausgezeichnetem *Blank-Symbol* $\Box$,
- einem Eingabealphabet $\Sigma \subseteq \Gamma \setminus \{\Box\}$,
- 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.

<u>Beachte:</u> u ist ein Wort *endlicher* Länge. Die unendlich vielen $\Box$-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 $C_0(w) := (q_0, 0, w\Box)$.

(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

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

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

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

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

Bemerkung: Ist M deterministisch und C keine Endkonfiguration, so enthält $\text{Next}_M(C)$ genau 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 $L(M) := \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$ 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 nicht-deterministische Turing-Maschinen genau dieselben Sprachen entscheiden können:

1.17 Satz.
Jede durch eine nicht-deterministische 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

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

Das *Halteproblem auf leerem Eingabewort* ist die Sprache

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

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

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:*

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

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

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

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.

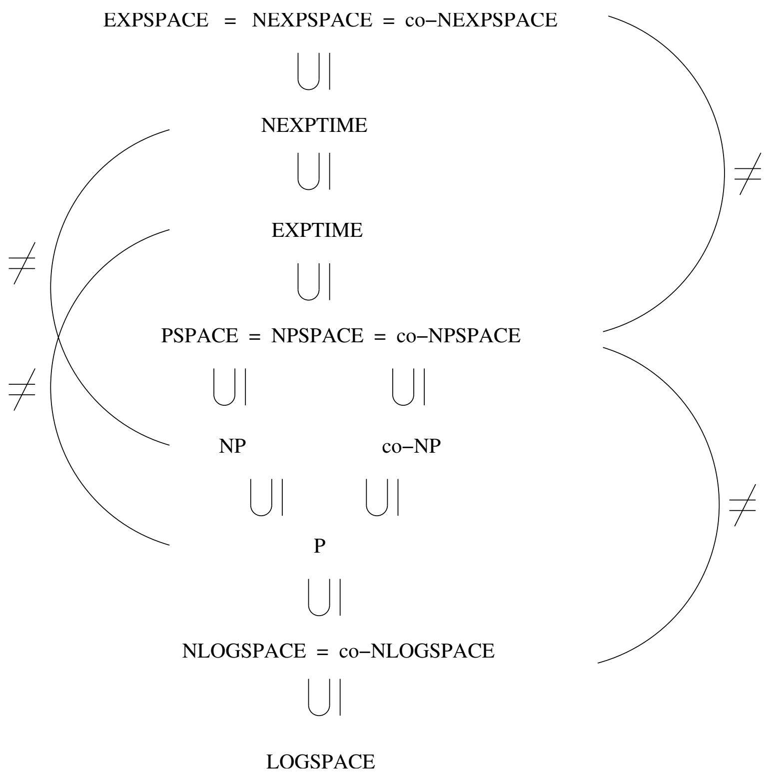


Abbildung 1.1: Inklusionsstruktur der Komplexitätsklassen.

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*

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

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 ist wegen ihres engen Zusammenhangs zur Logik zweiter Stufe für diese Vorlesung von Interesse; siehe Kapitel 2 (Deskriptive Komplexität). Um die Komplexitätsklassen der Polynomialzeit-Hierarchie einführen zu können, benötigen wir den Begriff der *Orakel-Turing-Maschinen*.

1.40 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 $B \in \mathrm{NP}$.

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:

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

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

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

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

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

1.42 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.
- $\mathrm{NP} \subseteq \mathrm{P}^{\mathrm{NP}}$ 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 $\mathrm{P}^{\mathrm{NP}} = \mathrm{P}^{\textsc{Sat}} = \mathrm{P}^{\textsc{Tsp}} = \mathrm{P}^{\textsc{Clique}}$.
- $\mathrm{P}^{\mathrm{NP}} \subseteq \textsc{Pspace}$.

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 zum Begriff der *Polynomialzeit-Hierarchie* (auch: *Polynomielle Hierarchie* bzw. *Polynomiale Hierarchie*):

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

- $\Delta_0^p \; := \; \Sigma_0^p \; := \; \Pi_0^p \; := \; \mathrm{P}$.
- $\Delta_{k+1}^p \; := \; \mathrm{P}^{\Sigma_k^p}, \qquad \Sigma_{k+1}^p \; := \; \mathrm{NP}^{\Sigma_k^p}, \qquad \Pi_{k+1}^p \; := \; \text{co-}\Sigma_{k+1}^p$.

Des Weiteren setzen wir $\mathrm{PH} \; := \; \bigcup_{k \in \mathbb{N}} \Sigma_k^p$.

1.44 Bemerkung. Man sieht leicht, dass

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

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

und

$$\mathrm{PH} \; \subseteq \; \mathrm{PSPACE}.$$

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

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

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

1.45 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 $\varphi(\mathfrak{A}) := \{(a_1, .\,., a_r) \in A^r \mid \mathfrak{A} \models \varphi[a_1, .\,., a_r]\}$.

$$
\begin{tikzcd}[column sep=small]
 & \mathrm{PSPACE} & \\
 & \mathrm{PH} \arrow[u, phantom, "\subseteq" sloped] & \\
 & \vdots \arrow[u, phantom, "\subseteq" sloped] & \\
 & \Delta^p_{k+1} \arrow[u, phantom, "\subseteq" sloped] & \\
\Sigma^p_k \arrow[ur, phantom, "\subseteq" sloped] & & \Pi^p_k \arrow[ul, phantom, "\subseteq" sloped] \\
 & \Delta^p_3 \arrow[ul, phantom, "\subseteq" sloped] \arrow[ur, phantom, "\subseteq" sloped] & \\
\Sigma^p_2 \arrow[ur, phantom, "\subseteq" sloped] & & \Pi^p_2 \arrow[ul, phantom, "\subseteq" sloped] \\
 & \Delta^p_2 \arrow[ul, phantom, "\subseteq" sloped] \arrow[ur, phantom, "\subseteq" sloped] & \\
\mathrm{NP} = \Sigma^p_1 \arrow[ur, phantom, "\subseteq" sloped] & & \Pi^p_1 = \text{co-NP} \arrow[ul, phantom, "\subseteq" sloped] \\
 & \mathrm{P} = \Sigma^p_0 = \Delta^p_0 = \Pi^p_0 = \Delta^p_1 \arrow[ul, phantom, "\subseteq" sloped] \arrow[ur, phantom, "\subseteq" sloped] &
\end{tikzcd}
$$

Abbildung 1.2: Inklusionsstruktur der Komplexitätsklassen der Polynomialzeit-Hierarchie.

(d) SUBSUMPTIONSPROBLEM FÜR $\mathscr{L}$ IN K (auch bekannt als *Query Containment Problem*):
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

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

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:

1.46 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

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

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

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

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

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

1.48 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.

1.49 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

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

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 $j \in \{0, .., n^r-1\}$ gilt:

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

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

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

1.51 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:

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

1.52 Bemerkung. Die Standardkodierung $\mathit{enc}(\mathfrak{A})$ 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 $\mathit{rg}_{<_{lex}}(\vec{a})$-te Speicherzelle zu lesen. Insgesamt heißt das:

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

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

1.3.2 Datenkomplexität, Programmkomplexität, kombinierte Komplexität

1.53 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:

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

(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:

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

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

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

1.54 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.

1.55 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

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

$\mathscr{K}$-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

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

$\mathscr{K}$-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

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

$\mathscr{K}$-vollständig ist.

1.3.3 Die Komplexität des Auswertungsproblems für FO

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

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

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

$$\mathrm{EVAL}(\mathfrak{A}, \varphi(x_1, \ldots, x_s), a_1, \ldots, a_s)$$

angeben, der bei Eingabe einer endlichen Struktur $\mathfrak{A}$, einer FO-Formel φ und Belegungen $a_1, \ldots, a_s \in A$ für die freien Variablen von φ entscheidet, ob $\mathfrak{A} \models \varphi[a_1, \ldots, a_s]$; 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 $enc(\varphi)$ 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 1.56 ergibt sich direkt:

1.57 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 1.56, die das Problem

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

löst und bei Eingaben $enc(\mathfrak{A})\#enc(\varphi)$ mit $n := |enc(\mathfrak{A})|$ und $m := |enc(\varphi)|$ 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

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

Man kann leicht eine DTM M_φ bauen, die bei Eingabe *enc*$(\mathfrak{A})$ genau das tut, was M bei Eingabe *enc*$(\mathfrak{A})\#$*enc*(φ) tun würde. M_φ löst dann das Problem L_φ, und da die Formel φ fest ist, ist $m = |enc(\varphi)|$ 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 $\mathrm{AC}^0 \subsetneqq$ LOGSPACE. (Wir werden das in dieser Vorlesung allerdings nicht beweisen.)
Hinsichtlich der *kombinierten Komplexität* ist ein PSPACE-Algorithmus aber das bestmögliche:

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

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

1.3.4 Der Satz von Trachtenbrot

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:

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

1.59 Theorem (Satz von Trachtenbrot, 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 $L_{\text{ERF-FO}[\sigma]\text{-}\mathsf{Fin}}$ entscheidet, d.h., die bei Eingabe der Kodierung eines FO$[\sigma]$-Satzes entscheidet, ob dieser ein endliches Modell hat.

<u>Idee:</u> 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

- $Q = \{0, 1, .., s_Q\}$ für ein $s_Q \in \mathbb{N}$, Anfangszustand $q_0 = 0$, $F \subseteq Q$,
- $\Gamma = \{0, 1, .., s_\Gamma\}$ für ein $s_\Gamma \in \mathbb{N}$, Blank-Symbol $\Box = 0$,
- $s := \max\{s_Q, s_\Gamma\}$,
- 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.

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.

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

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

(3.) $(t, p) \in K^{\mathfrak{A}} \Longleftrightarrow$ 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

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

Die Relationen $<^{\mathfrak{A}_M}$ und $succ^{\mathfrak{A}_M}$ 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:

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

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

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

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:

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

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

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

wobei die Formeln der letzten Zeile wie folgt gewählt sind:

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

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

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

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

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

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

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 $\Box = 0$:

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

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.)

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

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 $a_0, a_1, a_2, ..$, die den natürlichen Zahlen $0, 1, 2, \ldots$ entsprechen, und schränkt man $\mathfrak{A}$ ein auf das Teiluniversum
$$\{a_i \mid i \in A_M\},$$
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. $\square$

1.60 Bemerkung. Man kann sogar zeigen, dass der Satz von Trachtenbrot für die Signatur $\{E\}$ gilt, die aus nur einem 2-stelligen Relationssymbol besteht. (Daraus folgt dann natürlich auch, dass der Satz von Trachtenbrot 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 1.59 benutzten Signatur σ auf geeignete Weise durch Strukturen über der Signatur $\{E\}$ zu kodieren. Näheres dazu am Ende von Kapitel 2 (Deskriptive Komplexität; Korollar **??**).

Anwendungen des Satzes von Trachtenbrot:

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

Ordnungsinvarianz:

1.61 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:
$$(\mathfrak{A}, <_1^{\mathfrak{A}}) \models \varphi \iff (\mathfrak{A}, <_2^{\mathfrak{A}}) \models \varphi.$$

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

1.62 Satz. *Für jede Signatur σ, die mindestens ein 2-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: Angenommen, das Problem der Ordnungsinvarianz ist doch entscheidbar, d.h. es gibt eine TM M, die bei Eingabe eines 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 FO$[\sigma]$-Satzes ψ entscheidet, ob ψ ein endliches Modell hat, d.h., die das Erfüllbarkeitsproblem für FO$[\sigma]$ auf Fin löst. Dies steht aber im Widerspruch zum Satz von Trachtenbrot (Theorem 1.59 und Bemerkung 1.60).
Sei Q ein 1-stelliges Relationssymbol, das nicht zu σ gehört. Bei Eingabe eines FO$[\sigma]$-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:

$$\varphi \text{ ist ordnungsinvariant auf } \mathsf{Fin} \quad\Longleftrightarrow\quad \psi \text{ hat kein endliches Modell,}$$

denn: Für alle endlichen $\sigma\dot\cup\{Q\}$-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$[\sigma]$ entscheidet. Dies ist aber ein Widerspruch zum Satz von Trachtenbrot. $\square$

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:

1.63 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 $\varphi(x_1,\ldots,x_s)\in$ 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 $\varphi(\mathfrak{A},R_i^{\mathfrak{A}}) := \{\vec a\in A^s \mid (\mathfrak{A},R_i^{\mathfrak{A}})\models\varphi[\vec a]\}$.

1.64 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 Problem*

> 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 *?*

ist unentscheidbar.

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 FO$[\sigma]$-Satz φ.
> *Frage:* Ist φ *allgemeinguültig*, 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 Trachtenbrot erhält man, dass die Einschränkung des Allgemeingültigkeitsproblems auf die Klasse Fin aller *endlichen* Strukturen weder entscheidbar noch semi-entscheidbar ist:

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

> ALLGEMEINGÜLTIGKEIT AUF Fin:
> *Eingabe:* Ein 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

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

(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

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

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: $\mathrm{Mod}_{\mathsf{S}}(\varphi) = \{\mathfrak{A} \in \mathsf{S} \mid \mathfrak{A} \models \varphi\}$.

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) $\text{Var}_2 := \{\text{Var}_i^k \mid i, k \in \mathbb{N}_{\geqslant 1}\}$ 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 $v_1, .., v_k \in \text{Var}_1 \cup \{c \in \sigma \mid c \text{ ist ein Konstantensymbol}\}$, so gehört $X(v_1, .., v_k)$ 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 $\varphi(x_1, .., x_s, X_1, .., X_t)$ um anzudeuten, dass

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

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: $\mathrm{Mod}_{\mathsf{Fin}}(\Phi_{\text{3-col}}) = \{G \mid G \text{ ist ein 3-färbbarer Graph}\}$.

(b) Die Formel

$$\begin{aligned}\Phi_{\text{conn}} \;:=\; \forall X \Big(\; & \big(\exists x\, X(x) \,\wedge\, \exists y\, \neg X(y)\big) \;\to \\ & \exists u\, \exists v\, \big((E(u,v) \vee E(v,u)) \wedge X(u) \wedge \neg X(v)\big)\Big)\end{aligned}$$

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:

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

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

(c) Ein *Hamiltonkreis* in einem Graphen $G = (V,E)$ ist eine Folge $v_0,v_1,..,v_{n-1}$ von Knoten, so dass $V = \{v_0,..,v_{n-1}\}$ und es eine Kante von v_n nach v_1 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

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

Die Formel Φ_{Ham} ist von der Form

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

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 1.59 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 $v_0, v_1, .., v_{n-1}$, 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 ESO$[\sigma]$ aller *existentiellen* SO$[\sigma]$*-Formeln* besteht aus allen SO$[\sigma]$-Formeln Φ, die von der Gestalt

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

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

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

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

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 $\varphi \in$ FO$[\sigma \dot\cup \{X_1, \ldots, X_d\}]$.
Aus Lemma 1.56 folgt, dass es eine deterministische Turing-Maschine M' gibt, die das Problem

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

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 $enc(\mathfrak{A}')$, 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: $L_{\mathsf{Fin}}(\Phi) \in$ NP.

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

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

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
- $n^k \geqslant |enc(\mathfrak{A})|$.

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

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

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

$$rg_{<_{lex}}(\vec{t}) \ = \ t.$$

Analog wird jede Bandposition $p \in \{0,1,..,n^k{-}1\}$ durch ein k-Tupel $\vec{p} := (p_1,..,p_k) \in \{0,..,n{-}1\}^k$ kodiert. Ein Lauf von M bei Eingabe $enc(\mathfrak{A})$ 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 $\{0,..,n{-}1\}^k$, und somit mit Zahlen aus $\{0,1,..,n^k{-}1\}$ identifiziert werden können, benötigen wir außerdem eine lineare Ordnung $<^{\mathfrak{A}}$ auf dem Universum A von $\mathfrak{A}$, deren Nachfolger-Relation $succ^{\mathfrak{A}}$, sowie Elemente $z_0^{\mathfrak{A}}$ und $z_{max}^{\mathfrak{A}}$ für das kleinste und das größte Element in A bzgl. der Ordnung $<^{\mathfrak{A}}$.

Der ESO[σ]-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 $(z_{max}, \ldots, z_{max})$ repräsentiert wird, in dem *akzeptierenden* Zustand q_{akz} ist:

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

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

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

sowie eine FO-Formel

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

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

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

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

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 $\vec{z}_0 := (z_0, \ldots, z_0)$ 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 $\vec{z}_0 := (z_0, \ldots, z_0)$ kodiert wird), und für jedes $\vec{p} \in \{0, \ldots, n{-}1\}^k$ gilt: Auf Bandposition $i := \mathit{rg}_{<_{lex}}(\vec{p}) \in \{0, 1, \ldots, n^k{-}1\}$ 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

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

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:

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

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

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 $0, 1, .., n^k-1$), dass für jede endliche σ-Struktur $\mathfrak{A}$ gilt:

$\mathfrak{A} \models \Phi \iff$ es gibt einen Lauf von M bei Eingabe *enc*($\mathfrak{A}$), der nach n^k-1 Schritten in dem akzeptierenden Zustand q_{akz} 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[σ]-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 $d \geqslant 0$, $X_1, .., X_d$ Relationsvariablen und φ ein FO[$\sigma \dot\cup \{X_1, .., X_d\}$]-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

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

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 zur 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:

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

(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.

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

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:

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

(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]

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

[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

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

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

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

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

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

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 $j \in \{0, 1, .., |A|\}$ gelten, dass $R^j \subsetneqq R^{j+1}$ (denn sonst würde in jeder der Stufen $1, 2, .., |A|, |A|+1$ mindestens ein neues Element hinzukommen, in A gibt es aber nur $|A|$ viele Elemente).
Somit muss es also ein $i \in \{0, 1, .., |A|\}$ 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

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

$\square$

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. $\square$

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 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:

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

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 ***$F_{\varphi,\mathfrak{A}}$ 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

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

(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 der Übung 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

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

die *positiv* in $R_1, .., R_k$ und *negativ* in $S_1, .., S_\ell$ 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*. $\square$

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

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

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

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:

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

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

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

(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:

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

(c) *Eine definierbare Wortsprache:*

Sei $\sigma = \{<, P_a, P_b\}$ die Signatur, um Wörter $w \in \{a,b\}^*$ als σ-Strukturen $\mathfrak{A}_w$ zu kodieren (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:

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

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 $R^0, R^1, R^2, ..$ 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 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:

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

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

und ist $\vec{t} = t_1, \ldots, t_k$ 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, um Wörter $w \in \{a,b\}^*$ als σ-Strukturen $\mathfrak{A}_w$ zu kodieren (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:

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

Wir wählen $\varphi(R,x) :=$

$$\begin{aligned} &\Big(\text{“}P_a(0)\text{”} \ \wedge\ \text{“}P_b(\mathit{max})\text{”} \ \wedge\ \big(\text{“}x = 0\text{”} \vee \text{“}x = \mathit{max}\text{”}\big)\Big) \quad \vee \\ \exists y\, \exists z\, \Big(&\ y < z \ \wedge\ R(y) \ \wedge\ R(z) \ \wedge\ \forall v\, \big(\text{“}y < v < z\text{”} \to \neg R(v)\big) \wedge \\ &\ \text{“}P_a(y{+}1)\text{”} \ \wedge\ \text{“}P_b(z{-}1)\text{”} \ \wedge\ \big(\text{“}x = y{+}1\text{”} \ \vee\ \text{“}x = z{-}1\text{”}\big) \Big). \end{aligned}$$

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

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

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 $[\mathbf{ifp}_{R,\vec{x}}\,\varphi](\vec{t})$ 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 $R^0, R^1, R^2, ..$ 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 $i, j \in \{0, .., 2^n\}$, 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 $R^0, R^1, R^2, \dots$ der Iterationsstufen von der Form

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

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

2.40 Bemerkung. Für *inflationäre* 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 $R^0, R^1, R^2, ..$ 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 $z_X := \sum_{i<n} w_i \cdot 2^i \in \{0, .., 2^n-1\}$ — via

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

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

$$\big(\forall z\, R(z)\big) \ \lor\ \exists y \Big(\neg R(y) \ \land\ \forall z \big(y > z \to R(z)\big) \ \land \big(x = y \ \lor\ (x > y \land R(x))\big)\Big).$$

Durch Betrachten der Binärdarstellungen sieht man leicht, dass

- $R^0 = \emptyset$,
- 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 $k \in \mathbb{N}_{\geqslant 1}$,
- $\vec{x} = x_1, .., x_k$ 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[σ]-Formel.

2.43 Definition (Semantik von PFP[σ]-Formeln). Die Semantik von PFP[σ]-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:

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

□

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

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

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 und
- $n^k \geqslant |enc(\mathfrak{A})|$.

- $Q = \{0, 1, .., s_Q\}$ 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 $\Box = 2$.
- $s := \max\{s_Q, s_\Gamma\}$.

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

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

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 *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.
- $(max, \vec{t}, \vec{max}, q) \in R^{\mathfrak{A}} \iff M$ 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 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.:

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

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

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

Die Formel φ ist von der Form

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

wobei $\varphi_{Zahlen}(z_0,..,z_s)$ eine 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 $enc(\mathfrak{A})$ 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 $enc(\mathfrak{A})$ an Position $rg^{\mathfrak{A}}_{<_{lex}}(\vec{p})$ das Symbol 0 bzw. das Symbol 1 steht. Wir wählen

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

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

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

Wir wählen

$$
\begin{aligned}
&\varphi_{\mathit{Schritt}}(R, u, \vec{t}, \vec{p}, z) \ := \\
&\exists \vec{t}\,' \, \exists \vec{p}\,' \Big(\mathit{succ}_{<_{\mathit{lex}}}(\vec{t}\,', \vec{t}) \ \wedge \ R(0, \vec{t}\,', \vec{p}\,', 0) \ \wedge && \text{(Kopfpos. } \vec{p}\,' \text{ zum Zeitpkt. } \vec{t}\,' := \vec{t}-1) \\
&\quad \bigvee_{\substack{q' \in Q \setminus F, \gamma' \in \Gamma, \\ (q,\gamma,m) := \delta(q',\gamma')}} \Big(R(\mathit{max}, \vec{t}\,', m\vec{a}x, z_{q'}) \ \wedge \ R(z_1, \vec{t}\,', \vec{p}\,', z_{\gamma'}) \ \wedge && \text{(zum Zeitpkt. } \vec{t}\,'\text{: in Zustand } q'\text{, liest Symbol } \gamma') \\
&\qquad \Big(\ \big(u = \mathit{max} \ \wedge \ \vec{p} = m\vec{a}x \ \wedge \ z = z_q\big) && \text{(zum Zeitpkt. } \vec{t} \text{ in Zustand } q) \\
&\qquad \ \vee \ \big(u = 0 \ \wedge \ \chi_m(\vec{p}\,', \vec{p}) \ \wedge \ z = 0\big) && \text{(zum Zeitpkt. } \vec{t} \text{ Kopfpos. } \vec{p}\,' + m) \\
&\qquad \ \vee \ \big(u = z_1 \ \wedge \ \text{“}\vec{p}\,' = \vec{p}\text{”} \ \wedge \ z = z_\gamma\big) && \text{(zum Zeitpkt. } \vec{t} \text{ Symbol } \gamma \text{ auf Pos. } \vec{p}\,') \\
&\qquad \ \vee \ \big(u = z_1 \ \wedge \ \text{“}\vec{p}\,' \neq \vec{p}\text{”} \ \wedge \ R(z_1, \vec{t}\,', \vec{p}, z)\big) \Big) \Big) \Big) && \text{(auf Pos. } \vec{p} \neq \vec{p}\,' \text{ zum Zeitpkt. } \vec{t} \text{ gleiches Symbol wie zum Zeitpkt. } \vec{t}\,')
\end{aligned}
$$

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

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

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:

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

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 $\vec{t} = (t_1, \ldots, t_k) \in A^k$ 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:

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

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

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

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:

$$\text{LFP bzw. IFP beschreibt } \textit{nicht } \textsc{Ptime} \text{ auf } \mathsf{Fin},$$

$$\text{PFP beschreibt } \textit{nicht } \textsc{Pspace} \text{ auf } \mathsf{Fin}.$$

Aus den Theoremen 2.46 und 2.47 ergibt sich direkt:

2.49 Korollar. $\textsc{Pspace} = \textsc{Ptime} \iff \text{PFP} = \text{LFP}$ *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 4 (Fixpunktlogiken) auch beweisen werden:

$$\textsc{Pspace} = \textsc{Ptime} \iff \text{PFP} = \text{LFP auf } \mathsf{FinOrd} \iff \text{PFP} = \text{LFP auf } \mathsf{Fin}.$$

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 $\textsc{Pspace}$ nutzen, um die $\textsc{Pspace}$-Vollständigkeit des Auswertungsproblems für FO zu beweisen (dies wurde in Kapitel 1, Satz 1.58 schon erwähnt, aber noch nicht bewiesen):

2.51 Satz. *Die kombinierte Komplexität des Auswertungsproblems für* FO *auf* FinOrd *ist* $\textsc{Pspace}$*-vollständig.*

Beweis: Mit Satz 1.57 wurde bereits gezeigt, dass die kombinierte Komplexität des Auswertungsproblems für FO in $\textsc{Pspace}$ liegt. Im Folgenden zeigen wir, dass das Problem auch $\textsc{Pspace}$-hart ist, d.h. dass sich jedes Problem $L \in \textsc{Pspace}$ auf das Auswertungsproblem für FO reduzieren lässt. Jedes Problem $L \in \textsc{Pspace}$ kann man, für eine geeignete Signatur σ, mit einer Klasse $\mathsf{C} \subseteq \mathsf{FinOrd}$ (bzw. der zugehörigen Menge L_{C}) identifizieren. Da nach Voraussetzung $L_{\mathsf{C}} \in \textsc{Pspace}$ ist, liefert Theorem 2.47, dass es einen 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 $\vec{x} = x_1, .., x_r$.

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 $v_1, .., v_N$ Variablen erster Stufe und seien $\vec{x}, \vec{y}_1, \ldots, \vec{y}_N$ jeweils r-Tupel von Variablen erster Stufe. Für jedes $i \in \{0, .., N\}$ 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 $\vec{b}_1, .., \vec{b}_N \in A^r$ und alle $d_1, .., d_N \in \{0^{\mathfrak{A}}, max^{\mathfrak{A}}\}$ gilt:

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

Bevor wir die genaue Konstruktion der Formeln φ_i (für $i \in \{0, .., N\}$) 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

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

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

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

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

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

Dabei besagt

- Zeile 2, dass $R^{2^N} \stackrel{\text{def}}{=} F_{\varphi,\mathfrak{A}}{}^{2^N}(\varnothing) \ = \ \{\vec{y}_j \mid v_j \neq 0\}$,
- Zeile 3, dass $(m\vec{a}x, 0) \in R^{2^N}$,
- 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:

$$\mathfrak{A} \in \mathsf{C} \iff \mathfrak{A} \models [\mathbf{pfp}_{R,\vec{x}}\, \varphi](m\vec{a}x, 0) \iff \mathfrak{A} \models \psi_{\Phi,\mathfrak{A}}.$$

Somit ist die Abbildung, die jeder σ-Struktur $\mathfrak{A} \in \mathsf{FinOrd}$ 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

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

<u>*Induktionsanfang $i = 0$:*</u> $2^i = 2^0 = 1$. 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 $|\varphi_0| = \mathcal{O}(|\varphi| \cdot r \cdot N) = \text{poly}(n)$.

<u>*Induktionsschritt $i \to i{+}1$:*</u> 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:

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

setzen

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

und nutzen die Formel φ_i, um die Mengen $\{\vec{y}\,'_j \mid v'_j \neq 0\}$ und $\{\vec{y}\,''_j \mid v''_j \neq 0\}$ 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.

$$\begin{array}{l}
\varphi_{i+1}(\vec{x}, \vec{y}_1, v_1, .., \vec{y}_N, v_N) \;:= \\[2ex]
\exists \vec{y}\,'_1\, \exists v'_1 \;\cdots\; \exists \vec{y}\,'_N\, \exists v'_N \;\; \exists \vec{y}\,''_1\, \exists v''_1 \;\cdots\; \exists \vec{y}\,''_N\, \exists v''_N \;\Big(\\
\quad \bigwedge_{j=1}^{N} \big(v''_j \neq 0 \,\wedge\, \vec{x} = \vec{y}\,''_j\big) \;\wedge \\
\quad \bigwedge_{j=1}^{N} \big(\, v'_j = 0 \,\vee\, v'_j = \mathit{max}\big) \;\wedge\; \bigwedge_{j=1}^{N} \big(\, v''_j = 0 \,\vee\, v''_j = \mathit{max}\big) \;\wedge \\
\quad \forall \vec{w}_1 \forall u_1 \cdots \forall \vec{w}_N \forall u_N \;\Big(\big(\, \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}, v)} \;\vee\; \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}\,', v')}\,\big) \;\to \\
\qquad\qquad \big(\, \forall \vec{x} \;\;(\;\; \varphi_i(\vec{x}, \vec{w}_1, u_1, .., \vec{w}_N, u_N) \;\leftrightarrow \\
\qquad\qquad\qquad \Big(\big(\, \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}, v)} \;\wedge\; \bigvee_{j=1}^{N} (\, v'_j \neq 0 \,\wedge\, \vec{x} = \vec{y}\,'_j\,)\,\big) \vee \\
\qquad\qquad\qquad \big(\, \overrightarrow{(\vec{w}, u)} = \overrightarrow{(\vec{y}\,', v')} \;\wedge\; \bigvee_{j=1}^{N} (\, v''_j \neq 0 \,\wedge\, \vec{x} = \vec{y}\,''_j\,)\,\big)\Big)\,)\,\big)\Big)\Big),
\end{array}$$

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

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

2.53 Definition (Transitive Hüllen-Logik TC). Sei σ eine Signatur.
Die Formelmenge $\text{TC}[\sigma]$ 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 $\text{TC}[\sigma]$-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 $\text{TC}[\sigma]$-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

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

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

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

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 $\mathfrak{A}_n := (\{0,..,n\},\mathit{succ}^n)$ mit $\mathit{succ}^n := \{(i,i{+}1) \mid 0 \leqslant i < n\}$ die natürliche lineare Ordnung auf $\{0,..,n\}$, d.h. es gilt für alle $n \in \mathbb{N}$ und alle $i,j \in \{0,..,n\}$, dass

$$\mathfrak{A}_n \models \psi[i,j] \quad\Longleftrightarrow\quad i < j.$$

(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 $\mathfrak{B}_n := (\{0,..,n\},\mathit{succ}^n,0)$ die Addition auf $\{0,..,n\}$. D.h. für alle $n \in \mathbb{N}$ und alle $a,b,c \in \{0,..,n\}$ 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:

$$\mathfrak{A} \;\models\; \exists z_0\,\exists z_{\mathit{max}}\,\Big(\,\forall x\,\big(\,\neg\,x < z_0 \;\;\wedge\;\; \neg\,z_{\mathit{max}} < x\,\big)\;\wedge\\ \big[\,\mathbf{tc}_{x,y}\;\exists z\big(x < z < y\;\wedge\forall z'(x < z' < y \to z' = z)\big)\big](z_0,z_{\mathit{max}})\,\Big)$$

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:

$$\mathbf{dtc}(R) \;:=\; \left\{ (\vec{x},\vec{y}) \in A^{2k} \;\middle|\; \begin{array}{l} \text{es gibt im Graphen } G_R := (V_R, E_R) \text{ mit } V_R := A^k \\ \text{und } E_R := \{(\vec{u},\vec{v}) \in A^k \times A^k \mid (\vec{u},\vec{v}) \in R\} \text{ einen} \\ \textit{deterministischen} \text{ Pfad von } \vec{x} \text{ zu } \vec{y}\text{, d.h. einen Pfad} \\ \text{der Länge} \geqslant 1\text{, so dass es für jeden Knoten } \vec{u} \text{ auf} \\ \text{diesem Pfad (außer evtl. } \vec{y}\text{) } \textit{genau einen} \text{ Knoten } \vec{v} \\ \text{mit } (\vec{u},\vec{v}) \in E_R \text{ gibt} \end{array} \right\}.$$

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

$$\big[\,\mathbf{tc}_{\vec{x},\vec{y}}\;\Big(\varphi(\vec{x},\vec{y}) \;\wedge\; \forall\vec{z}\big(\varphi(\vec{x},\vec{z}) \rightarrow \vec{z}=\vec{y}\big)\Big)\;\big]\;(\vec{s},\vec{t}).$$

$\square$

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 "→" noch Biimplikationspfeile "↔" 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

$$\begin{aligned} \psi' \;\; := \;\; \exists \vec{z} \Big(\; & \big(\, \vec{s} = \vec{z} \,\vee\, \big[\, \mathbf{dtc}_{\vec{x},\vec{y}} \big(\varphi(\vec{x},\vec{y}) \,\wedge\, \neg\, \vec{y} = \vec{t}\big) \,\big]\, (\vec{s},\vec{z}) \,\big) \\ & \wedge\, \Big(\big(\forall \vec{u}\, \neg\varphi(\vec{z},\vec{u}) \;\;\vee\;\; \exists \vec{u}\, \exists \vec{v}\, (\varphi(\vec{z},\vec{u}) \wedge \varphi(\vec{z},\vec{v}) \wedge \neg\, \vec{u} = \vec{v})\big) & (1) \\ & \qquad \vee \big[\, \mathbf{dtc}_{\vec{x},\vec{y}} \, \big(\varphi(\vec{x},\vec{y}) \,\wedge\, \neg\, \vec{y} = \vec{t}\big) \,\big]\, (\vec{z},\vec{z}) \,\Big) \Big). & (2) \end{aligned}$$

□

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 (Satz **??**).

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 1.56 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 nicht-deterministische 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 $|enc(\mathfrak{A})| \leqslant n^k$.

<u>*Idee:*</u> Jede Konfiguration von M bei Eingabe *enc*$(\mathfrak{A})$ ist durch ein Tupel $(q, p_{\text{Eingabe}}, p, w)$ eindeutig bestimmt, wobei $q \in Q$ den aktuellen Zustand bezeichnet, $p_{\text{Eingabe}} \in \{0,..,n^k\}$ die Kopfposition auf dem Eingabeband, $p \in \{0,..,k\cdot \lfloor \log n \rfloor\}$ 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 *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.

<u>*Genauer:*</u> 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 $\vec{x}_\gamma = (x_{\gamma,0},..,x_{\gamma,k-1})$, für $\gamma \in \Gamma$, beschreiben die Beschriftung des Arbeitsbandes folgendermaßen. Für alle $i,j \in \{0,..,n-1\}$ gilt:

an Position $j \cdot \log n + i$ steht das Symbol $\gamma \iff$ das i-te Bit der Zahl $x_{\gamma,j}$ ist 1, d.h. $\big\lfloor \frac{x_{\gamma,j}}{2^i} \big\rfloor$ ist ungerade.

Man kann zeigen (*Übung*), dass es posDTC[<]-Formeln $\varphi_{Bit=1}(x,y)$ und $\varphi_{Bit=0}(x,y)$ 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}, $\varphi_{Akzeptiere}$ und $\varphi_{Schritt}$ 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

$$\psi \;:=\; \neg\left[\,\mathbf{tc}_{\vec{x},\vec{y}}\;\varphi(\vec{x},\vec{y})\,\right](\vec{s},\vec{t}\,)$$

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:

$$\begin{array}{ll} & \mathfrak{A} \models \neg[\mathbf{tc}_{\vec{x},\vec{y}}\,\varphi](\vec{s},\vec{t}\,) \\ \iff & \left|\{\vec{b} \in A^k \;:\; \mathrm{Dist}^{\mathfrak{A}}_{\varphi}(\vec{s},\vec{b}) < \infty\}\right| \;=\; \left|\{\vec{b} \in A^k \;:\; \mathrm{Dist}^{\mathfrak{A}}_{\varphi(\vec{x},\vec{y})\,\wedge\,\neg\vec{y}=\vec{t}}(\vec{s},\vec{b}) < \infty\}\right| \qquad (*)\end{array}$$

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 $\{0,..,n{-}1\}$ identifizieren. Unter Verwendung der lexikographischen Ordnung $<^{\mathfrak{A}}_{lex}$ identifizieren wir $k{+}1$-Tupel $\vec{c} \in A^{k+1}$ mit Zahlen in $\{0,..,n^{k+1}{-}1\}$, indem wir dem Tupel $\vec{c}$ die Zahl

$$\left[\vec{c}\,\right]^{\mathfrak{A}} \;:=\; rg_{<^{\mathfrak{A}}_{lex}}(\vec{c}) \;\in\; \{0,..,n^{k+1}{-}1\}$$

zuordnen. Insbesondere gilt

$$|A|^k \;=\; n^k \;=\; \left[\,(1,\underbrace{0,0,\cdots,0}_{k})\,\right]^{\mathfrak{A}},$$

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 $dist_{\chi}(\vec{x},\vec{y},\vec{u})$ (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:

$$\mathfrak{A} \models dist_{\chi}[\vec{a},\vec{b},\vec{d}\,] \iff \mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{a},\vec{b}) \;\leqslant\; \left[\vec{d}\,\right]^{\mathfrak{A}}.$$

Dazu wählen wir

$$dist_{\chi}(\vec{x},\vec{y},\vec{u}) \;:=\; \left[\;\mathbf{tc}_{\vec{x},\vec{u},\vec{x}',\vec{u}'}\;\chi(\vec{x},\vec{x}\,') \wedge succ_{<_{lex}}(\vec{u},\vec{u}\,')\,\right](\vec{x},\vec{0},\vec{y},\vec{u}).$$

(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 $anzahl_{\chi}(\vec{x},\vec{z})$, 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:

$$\mathfrak{A} \models anzahl_{\chi}\,[\vec{s},\vec{c}\,] \iff \left[\vec{c}\,\right]^{\mathfrak{A}} = \left|\,\{\vec{b} : \mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{b}) < \infty\,\}\,\right|.$$

Dazu betrachten wir zunächst die Funktion

$$\mathrm{Anz}^{\mathfrak{A}}_{\chi} \;:\; A^k \times \{0,..,n^k\} \;\to\; \{0,..,n^k\}$$

mit
$$\mathrm{Anz}^{\mathfrak{A}}_{\chi}(\vec{s},d) \;:=\; \big|\{\vec{b}\in A^k \,:\, \mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{b})\leqslant d\}\big|\,.$$
Für jedes feste $\vec{s}\in A^k$ können wir $\mathrm{Anz}^{\mathfrak{A}}_{\chi}(\vec{s},d)$ induktiv für $d=0,1,2,\ldots$ folgendermaßen bestimmen:

<u>Induktionsanfang $d=0$:</u>

$\mathrm{Anz}^{\mathfrak{A}}_{\chi}(\vec{s},0)=0$, denn für alle $\vec{b}\in A^k$ gilt gemäß Definition, dass $\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{b})\geqslant 1$.

<u>Induktionssschritt $d\to d{+}1$:</u>

Sei $c := \mathrm{Anz}^{\mathfrak{A}}_{\chi}(\vec{s},d)$. Gesucht ist die Zahl c', so dass $\mathrm{Anz}^{\mathfrak{A}}_{\chi}(\vec{s},d{+}1) = c'$. 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: $\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{y})\leqslant d{+}1$.

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:
$$\text{Falls}\quad \big[\vec{z}\big]^{\mathfrak{A}} = \mathrm{Anz}^{\mathfrak{A}}_{\chi}\big(\vec{s},\big[\vec{u}\big]^{\mathfrak{A}}\big),\quad \text{so}\quad \big[\vec{z}\,'\big]^{\mathfrak{A}} = \mathrm{Anz}^{\mathfrak{A}}_{\chi}\big(\vec{s},\big[\vec{u}\,'\big]^{\mathfrak{A}}\big).$$
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
$$\mathrm{Dist}^{\mathfrak{A}}_{\chi}\big(\vec{s},\vec{y}\,'\big) \;>\; \big[\vec{u}\,'\big]^{\mathfrak{A}}.$$
Falls $\big[\vec{u}\,'\big]^{\mathfrak{A}} = \big[\vec{u}\big]^{\mathfrak{A}}{+}1$ und $\big[\vec{z}\big]^{\mathfrak{A}} = \mathrm{Anz}^{\mathfrak{A}}_{\chi}\big(\vec{s},\big[\vec{u}\big]^{\mathfrak{A}}\big)$, so gilt:

- Ist $\big[\vec{u}\big]^{\mathfrak{A}}=0$, so $\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{y}\,') > \big[\vec{u}\,'\big]^{\mathfrak{A}} \iff \mathfrak{A}\not\models\chi(\vec{s},\vec{y}\,')$.
- Ist $\big[\vec{u}\big]^{\mathfrak{A}}>0$, so $\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{y}\,') > \big[\vec{u}\,'\big]^{\mathfrak{A}} \iff$ es gibt $\vec{z}$ Elemente $\vec{w}\,'\neq\vec{y}\,'$, so dass $\mathrm{Dist}^{\mathfrak{A}}_{\chi}(\vec{s},\vec{w}\,')\leqslant\big[\vec{u}\big]^{\mathfrak{A}}$ und $\mathfrak{A}\not\models\chi(\vec{w}\,',\vec{y}\,')$.

Genau dies wird durch die folgende posTC-Formel bewerkstelligt, die in der Formel δ_χ an die Stelle von "$\neg dist_\chi(\vec{s},\vec{y}\,',\vec{u}\,')$" gesetzt werden muss:

$$\Big(\vec{u}=\vec{0} \;\wedge\; \neg\,\chi(\vec{s},\vec{y}\,')\Big) \vee$$
$$\Big(\vec{u}\neq\vec{0} \;\wedge\; \left[\mathbf{tc}_{\vec{w},\vec{q},\vec{w}',\vec{q}'} \begin{pmatrix} \big(\, succ_{<_{lex}}(\vec{w},\vec{w}\,') \;\wedge\; \vec{q}=\vec{q}\,'\big) \vee \\ \big(\, succ_{<_{lex}}(\vec{w},\vec{w}\,') \;\wedge\; succ_{<_{lex}}(\vec{q},\vec{q}\,') \;\wedge \\ \quad \vec{w}\,'\neq\vec{y}\,' \;\wedge\; dist_\chi(\vec{s},\vec{w}\,',\vec{u}) \;\wedge\; \neg\,\chi(\vec{w}\,',\vec{y}\,')\big) \end{pmatrix}\right] (\vec{0},\vec{0},m\vec{a}x,\vec{z}\,)\Big).$$

(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 $\mathrm{Dist}_\chi^{\mathfrak{A}}(\vec{s},\vec{w}\,')\leqslant[\vec{u}]^{\mathfrak{A}}$ und $\mathfrak{A}\not\models\chi(\vec{w},\vec{y}\,')$.)

Schließlich wählen wir

$$anzahl_\chi(\vec{s},\vec{z}) \;:=\; \big[\,\mathbf{tc}_{\vec{u},\vec{z},\vec{u}',\vec{z}'}\; succ_{<_{lex}}(\vec{u},\vec{u}\,') \wedge \zeta_\chi(\vec{u},\vec{z},\vec{u}\,',\vec{z}\,',\vec{s}\,)\big]\,(\underbrace{0\vec{0}}_{\triangleq 0},\vec{0},\underbrace{1\vec{0}}_{\triangleq n^k},\vec{z}\,)\,.$$

Dies ist eine posTC-Formel, die besagt, dass $[\vec{z}]^{\mathfrak{A}}$ 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 $anzahl_\chi(\vec{s},\vec{z})$, dass $[\vec{z}]^{\mathfrak{A}}$ 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

$$\exists\vec{z}\,\Big(anzahl_\varphi(\vec{s},\vec{z}) \;\wedge\; anzahl_{\varphi(\vec{x},\vec{y})\wedge\neg\vec{y}=\vec{t}}(\vec{s},\vec{z})\Big)$$

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 abgschlossen 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 $L\in$ NLOGSPACE. "$\subseteq$:" analog. □

Zusammenfassung der logischen Charakterisierungen:

In Kapitel 2 wurden die in Abbildung 2.1 dargestellten logischen Charakterisierungen von Komplexitätsklassen behandelt.

$$
\begin{array}{cccc}
\mathrm{PFP} & \longrightarrow & \text{beschreibt auf } \mathsf{FinOrd} \longrightarrow & \textsc{Pspace} \\
 & & & \cup\!\mid \\
\mathrm{SO} & \longrightarrow & \text{beschreibt auf } \mathsf{Fin} \longrightarrow & \mathrm{PH} \\
 & & & \cup\!\mid \\
\mathrm{ESO} & \longrightarrow & \text{beschreibt auf } \mathsf{Fin} \longrightarrow & \mathrm{NP} \\
 & & & \cup\!\mid \\
\mathrm{LFP},\ \mathrm{IFP} & \longrightarrow & \text{beschreibt auf } \mathsf{FinOrd} \longrightarrow & \textsc{Ptime} \\
 & & & \cup\!\mid \\
\mathrm{TC},\ \mathrm{posTC} & \longrightarrow & \text{beschreibt auf } \mathsf{FinOrd} \longrightarrow & \textsc{NLogspace} \\
 & & & \cup\!\mid \\
\mathrm{DTC},\ \mathrm{posDTC} & \longrightarrow & \text{beschreibt auf } \mathsf{FinOrd} \longrightarrow & \textsc{Logspace}
\end{array}
$$

Abbildung 2.1: Logische Beschreibungen von Komplexitätsklassen.

2.4 Interpretationen und Logische Reduktionen

2.69 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: $\mathfrak{A} \in \mathsf{C} \iff \mathfrak{A} \models \varphi$.)

2.70 Definition. Sei σ eine Signatur. Mit σ-$\mathsf{STRUKTUREN}$ bezeichnen wir die Klasse aller σ-Strukturen.

Ziel in diesem Abschnitt:

Begriff für "logische Reduktionen", analog 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 $w \in A \iff f(w) \in B$ jetzt:

Eine Klasse C von τ-Strukturen, eine Klasse D von σ-Strukturen und eine “logisch definierbare” Reduktion $I : \tau\text{-STRUKTUREN} \to \sigma\text{-STRUKTUREN}$, so dass für alle τ-Strukturen $\mathfrak{A}$ gilt: $\mathfrak{A} \in \mathsf{C} \iff \mathcal{I}(\mathfrak{A}) \in \mathsf{D}$.
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:

2.71 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 $1 \leqslant i \leqslant m$).[6] 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 $\vec{x}_j = x_{j,1}, .., x_{j,k}$).

(b) Eine Interpretation I von σ in τ definiert eine Abbildung

$$I \ : \ \tau\text{-STRUKTUREN} \ \to \ \sigma\text{-STRUKTUREN},$$

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

$$U \ := \ \varphi_{\text{Univ}}(\mathfrak{A}) \ = \ \{\vec{a} \in A^k \mid \mathfrak{A} \models \varphi_{\text{Univ}}[\vec{a}]\}.$$

- Für jedes $R_i \in \sigma$ ist $R_i^{I(\mathfrak{A})}$ die r_i-stellige Relation

$$R_i^{I(\mathfrak{A})} \ := \ \varphi_{R_i}(\mathfrak{A}) \cap U^{r_i} \ = \ \{(\vec{a}_1, .., \vec{a}_{r_i}) \in U^{r_i} \mid \mathfrak{A} \models \varphi_{R_i}[\vec{a}_1, .., \vec{a}_{r_i}]\}.$$

(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 $I(\mathfrak{A})$).

2.72 Beispiele.
Sei $\tau := \{<\}$ die Signatur für lineare Ordnungen und S die Klasse aller endlichen linearen Ordnungen $\mathfrak{A} = (A, <^{\mathfrak{A}})$.

[6] 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.

(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[7] Graphen $I_2(\mathfrak{A}) = G = (V, E^G)$ zuordnet, so dass gilt

$$|A| \text{ ist gerade} \iff G \text{ ist zusammenhängend}.$$

Idee: Ist $A = \{0, .., n\}$, 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 $C_1 = \{0, 2, 4, .., n\}$ und $C_2 = \{1, 3, 5, .., n{-}1\}$. Ist n ungerade (also $|A|$ gerade), so besteht G aus einem Kreis $C = \{0, 2, 4, .., n{-}1, 1, 3, 5, .., n\}$.

[7] 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 $(w, v) \in E$.

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

$$\begin{aligned}
\psi_{\text{Univ}}(x) \;&:=\; x = x\\
\psi_E(x,y) \;&:=\; \quad \varphi_E(x,y) \,\vee\, \varphi_E(y,x) && (1)\\
&\qquad \vee \big((\text{“}x{=}min\text{”} \wedge \text{“}y{=}max\text{”}) \,\vee\, (\text{“}y{=}min\text{”} \wedge \text{“}x{=}max\text{”})\big) && (2)\\
&\qquad \vee \big((\text{“}x{=}min{+}1\text{”} \wedge \text{“}y{=}max{-}1\text{”}) \,\vee\, (\text{“}y{=}min{+}1\text{”} \wedge \text{“}x{=}max{-}1\text{”})\big) && (3)
\end{aligned}$$

Dabei ist $\varphi_E(x,y)$ die Formel aus (a) und

$$\begin{aligned}
\text{“}x{=}min\text{”} \;&:=\; \neg\,\exists z\; z < x\,,\\
\text{“}y{=}max\text{”} \;&:=\; \neg\,\exists z\; y < z\,,\\
\text{“}x{=}min{+}1\text{”} \;&:=\; \exists x'\,\big(\,x' < x \,\wedge\, \forall z\,(z < x \rightarrow z = x')\,\big)\,,\\
\text{“}y{=}max{-}1\text{”} \;&:=\; \exists y'\,\big(\,y < y' \,\wedge\, \forall z\,(y < z \rightarrow z = y')\,\big)\,.
\end{aligned}$$

2.73 Definition ($\mathscr{L}$-Reduktion). Sei $\mathscr{L}$ eine Logik und sei $k \in \mathbb{N}_{\geqslant 1}$. Seien σ und τ Signaturen mit $\sigma = \{R_1, .., R_m\}$. 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:

(1.) $I(\mathfrak{A}) \in \mathsf{S}_2$ und

(2.) $\mathfrak{A} \in \mathsf{C} \iff I(\mathfrak{A}) \in \mathsf{D}$.

2.74 Beispiel. Die Interpretation I_2 aus Beispiel 2.72 (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.

2.75 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 $\hat{X}$,

$$\psi^I \quad := \quad \exists\hat{X}\,\Big(\forall\vec{z}_1\cdots\forall\vec{z}_r\,\big(\hat{X}(\vec{z}_1,..,\vec{z}_r) \to \bigwedge_{j=1}^{k}\varphi_{\text{Univ}}(\vec{z}_j)\big) \ \wedge \ \psi_1^I\Big)$$

bzw.

$$\psi^I \quad := \quad \forall\hat{X}\,\Big(\forall\vec{z}_1\cdots\forall\vec{z}_r\,\big(\hat{X}(\vec{z}_1,..,\vec{z}_r) \to \bigwedge_{j=1}^{k}\varphi_{\text{Univ}}(\vec{z}_j)\big) \ \to \ \psi_1^I\Big).$$

(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:

2.76 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.* □

2.77 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

$$\mathfrak{A} \in \mathsf{C} \iff I(\mathfrak{A}) \in \mathsf{D} \overset{(*)}{\iff} I(\mathfrak{A}) \models \psi \overset{\text{Lemma 2.76}}{\iff} \mathfrak{A} \models \psi^I.$$

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 2.77 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.

2.78 Beispiel. Aus Beispiel 2.74 und Korollar 2.77 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.

2.79 Satz. *Sei $\sigma = \{R_1, .., R_m\}$ 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:*

$$I(\mathfrak{A}) \textit{ ist ein ungerichteter Graph} \quad \textit{und} \quad J\big(I(\mathfrak{A})\big) \cong \mathfrak{A}.$$

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

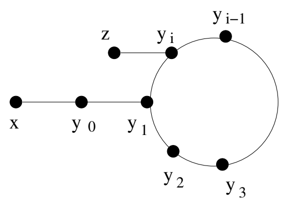


Abbildung 2.2: Das i-Etikett eines Knotens x

gibt (die Skizze bedeutet dabei, dass sämtliche Knoten verschieden sind, zwischen ihnen nur die eingezeichneten Kanten verlaufen und es von den Knoten $y_0, .., y_i, z$ 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 $r_1, .., r_m \in \mathbb{N}_{\geqslant 1}$ die Stelligkeiten der Relationssymbole $R_1, .., R_m$ 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 $j \in \{1, .., m\}$ und jedes r_j-Tupel $\vec{a} = (a_1, .., a_{r_j}) \in R_j^{\mathfrak{A}}$ 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 $i \in \{1, .., r_j\}$) 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

$$v_{j,\vec{a}} — w_{j,\vec{a},i,1} — w_{j,\vec{a},i,2} — \cdots — w_{j,\vec{a},i,i} — v_{a_i} \qquad (*)$$

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

$$\rho_E(x,y) \quad := \quad E(x,y) \ \wedge \ E(y,x) \ \wedge \ \neg x{=}y$$

und

$$\rho_{\neg E}(x,y) \quad := \quad \neg E(x,y) \ \wedge \ \neg E(y,x) \ \wedge \ \neg x{=}y\,.$$

Für jedes $i \geqslant 3$ sei *Etikett*$_i(x, y_0, .\,.\,, y_i, z)$ die folgende FO$[E]$-Formel, die besagt, dass die Knoten $x, y_0, .\,.\,, y_i, z$ ein i-Etikett für Knoten x bilden.

$$\textit{Etikett}_i(x, y_0, .\,.\,, y_i, z) \ :=$$

$$\rho_E(x,y_0) \ \wedge \rho_E(y_0,y_1) \ \wedge \ \bigwedge_{j=1}^{i-1} \rho_E(y_j,y_{j+1}) \ \wedge \ \rho_E(y_i,y_1) \ \wedge \ \rho_E(y_i,z) \ \wedge$$

$$\Big(\rho_{\neg E}(x,z) \ \wedge \ \bigwedge_{j=1}^{i} \rho_{\neg E}(x,y_j)\Big) \ \wedge \ \bigwedge_{j'=0}^{i-1} \Big(\rho_{\neg E}(y_{j'},z) \ \wedge \ \bigwedge_{j=j'+2}^{i} \rho_{\neg E}(y_{j'},y_j)\Big) \ \wedge$$

$$\forall u \forall v \ \Big(\big(\, u = z \vee \bigvee_{j=0}^{i} u = y_j \,\big) \ \wedge \big(\, v \neq x \ \wedge \ v \neq z \ \wedge \bigwedge_{j=0}^{i} v \neq y_j \,\big)\Big) \to \Big(\neg E(u,v) \ \wedge \ \neg E(v,u)\Big)\,.$$

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*$_i(x, y_0, .\,.\,, y_i, z)$ findet man für jedes $i \geqslant 3$ und jedes $u \in \{x, y_0, .\,.\,, y_i, z\}$ 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 $i \in \{5,..,5{+}m\}$ und ein $u \in \{x, y_0, .., y_i, z\}$,
- (j,i,k), für ein $j \in \{1,..,m\}$, $i \in \{1,..,r_j\}$, $k \in \{1,..,i\}$.

Schritt 3: Die (1-dimensionale) Interpretation

$$J \;:=\; \Big(\psi_{\text{Univ}}(v), \big(\psi_{R_j}(v_1,..,v_{r_j})\big)_{j=1,..,m}\Big)$$

von σ in $\{E\}$ ist folgendermaßen definiert:

$$\begin{aligned} \psi_{\text{Univ}}(v) &:= \alpha_{(5,x)}(v) \\ \psi_{R_j}(v_1,..,v_{r_j}) &:= \exists u \Big(\alpha_{(5+j,x)}(u) \wedge \bigwedge_{i=1}^{r_j} \begin{array}{l}\text{“es gibt einen Pfad der}\\ \text{Länge } i{+}1 \text{ von } u \text{ nach } v_i\text{”}\end{array}\Big) \end{aligned}$$

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 $r := \mathit{max}\{r_1,..,r_m\}$ 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:

$$v \text{ ist vom Typ } t_j \quad\Longleftrightarrow\quad c = b_j \;\text{ (und } c \neq b_i \text{ für alle } i \neq j\text{)}$$

(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: $\mathfrak{A} \cong J\big(I(\mathfrak{A})\big)$. $\square$

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.

Der Einfachheit halber betrachten wir in diesem Kapitel oft *relationale* Signaturen, d.h. Signaturen, die keine Konstantensymbole enthalten. (Dies ist keine wesentliche Einschränkung, da man jedes Konstantensymbol c durch ein 1-stelliges Relationssymbol C repräsentieren kann, das durch eine 1-elementige Menge interpretiert wird.)

3.1 Das EF-Spiel für die Logik erster Stufe

3.1.1 Vorbemerkungen

Bevor wir die Spielregeln des EF-Spiels einführen, legen wir zunächst einige nützliche Begriffe fest.

3.1 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 3.1 benötigt man in etwa so viele Quantoren wie es Elemente im Universum von $\mathfrak{A}$ gibt.

3.2 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.

3.3 Beispiele.

- $qr\big(\exists x\, \forall y\, (\, x = y \ \vee\ R(x,y,z)\,)\big) \;=\; 2,$
- $qr\big(\exists x\, \big(T(x) \ \vee\ \forall y\, R(x,y,z)\big)\big) \;=\; 2,$
- $qr\big((\,\exists x\, T(x)\,) \ \vee\ \forall y\, (\, R(y,y,z) \to y = z\,)\big) \;=\; 1.$

3.4 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.

3.5 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:

3.6 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 $\mathfrak{A}_m \equiv_m \mathfrak{B}_m$. □

3.7 Definition (Partieller Isomorphismus). Sei σ eine Signatur und $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen.

(a) Eine Abbildung
$$\pi \ : \ \text{def}(\pi) \ \to \ \text{bild}(\pi)$$
mit $\text{def}(\pi) \subseteq A$ und $\text{bild}(\pi) \subseteq B$ heißt *partieller Isomorphismus* von $\mathfrak{A}$ nach $\mathfrak{B}$, falls gilt:

  (a) π ist *bijektiv*
  (b) für jedes Konstantensymbol $c \in \sigma$ ist $c^{\mathfrak{A}} \in \text{def}(\pi)$ und $\pi(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$,
  (c) für jedes Relationssymbol $R \in \sigma$ der Stelligkeit $k := ar(R)$ und alle k-Tupel $\vec{a} \in \text{def}(\pi)^k$ gilt[1]
$$\vec{a} \ \in \ R^{\mathfrak{A}} \quad \Longleftrightarrow \quad \pi(\vec{a}) \ \in \ R^{\mathfrak{B}}.$$

(b) $\text{Part}(\mathfrak{A}, \mathfrak{B})$ bezeichnet die Menge aller partiellen Isomorphismen von $\mathfrak{A}$ nach $\mathfrak{B}$.

3.8 Bemerkungen.

(a) Ein partieller Isomorphismus π von $\mathfrak{A}$ nach $\mathfrak{B}$ ist ein Isomorphismus von $\mathfrak{A}_{|\text{def}(\pi)}$ auf $\mathfrak{B}_{|\text{bild}(\pi)}$. (Hier bezeichnen wir für jede Struktur $\mathfrak{C}$ und jede Menge $U \subseteq C$ die von U induzierte Substruktur von $\mathfrak{C}$ mit $\mathfrak{C}_{|U}$; vgl. Aufgabe 4 von Übungsblatt 2.)

(b) 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.

[1] Zur Erinnerung: Für $\vec{a} = (a_1, .., a_k)$ ist $\pi(\vec{a}) := \big(\pi(a_1), .., \pi(a_k)\big)$.

(c) Oft identifizieren wir eine Abbildung π mit ihrem Graph

$$\{ \big(a, \pi(a)\big) \mid a \in \mathrm{def}(\pi) \}.$$

Insbesondere bedeutet dann

$$\pi \subseteq \pi',$$

dass π' eine *Erweiterung* von π ist (d.h. $\mathrm{def}(\pi) \subseteq \mathrm{def}(\pi')$ und für alle $a \in \mathrm{def}(\pi)$ ist $\pi'(a) = \pi(a)$).

3.9 Proposition. *Sei σ eine Signatur. $\mathfrak{A}$ und $\mathfrak{B}$ seien σ-Strukturen.*

(a) $\pi := \emptyset$ (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: Klar. □

Man beachte: Die Existenz eines partiellen Isomorphismus $\pi : a_1, .., a_m \mapsto b_1, .., b_m$ bedeutet *nicht*, 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 $\vec{a}' = a'_1, .., a'_k \in A$ und $\vec{b}' = b'_1, .., b'_k \in B$.

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* (auch: Spieler I und Spieler II) 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 $(\mathfrak{A}, \vec{a}') \equiv_m (\mathfrak{B}, \vec{b}')$.

[2] Im Fall $k = 0$ schreiben wir oft $\mathcal{G}_m(\mathfrak{A}, \mathfrak{B})$ an Stelle von $\mathcal{G}_m(\mathfrak{A}, \vec{a}', \mathfrak{B}, \vec{b}')$.

Eine *Partie* des Spiels besteht aus m *Runden*.
In jeder Runde $i \in \{1, .., m\}$ 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

$$\pi \ : \ \left\{ \begin{array}{ccll} a_i & \mapsto & b_i & \text{für alle } i \in \{1, .., m\} \\ c^{\mathfrak{A}} & \mapsto & c^{\mathfrak{B}} & \text{für alle Konstantensymbole } c \in \sigma \\ a'_j & \mapsto & b'_j & \text{für alle } j \in \{1, .., k\} \end{array} \right\}$$

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

$$f_S \ : \ \bigcup_{i=0}^{m-1} \left(A^i \times B^i\right) \ \rightarrow \ A \dot{\cup} B.$$

(Sind $\vec{a} := a_1, .., a_i$ und $\vec{b} := b_1, .., b_i$ 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

$$f_D \ : \ \bigcup_{i=0}^{m-1} \left((A^i \times B^i) \times (A \dot{\cup} B)\right) \ \rightarrow \ A \dot{\cup} B,$$

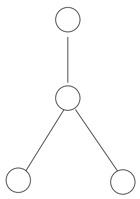
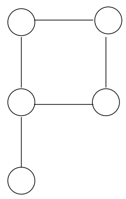
so dass für alle $\vec{a}, \vec{b}, c$ gilt: $f_D(\vec{a}, \vec{b}, c) \in B \iff c \in A$.
(Sind $\vec{a} := a_1, .., a_i$ und $\vec{b} := b_1, .., b_i$ die in den ersten i Runden gewählten Elemente und ist c das von Spoiler in der $i{+}1$-ten Runde gewählte Element, so gibt $f_D(\vec{a}, \vec{b}, c)$ 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.

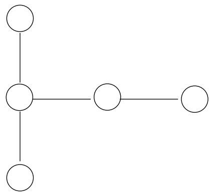
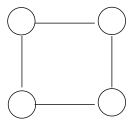
3.10 Beispiele.

(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 $\mathfrak{A} := (\{0,..,8\}, <, 0, 8)$ und $\mathfrak{B} := (\{0,..,7\}, <, 0, 7)$ gilt: Duplicator gewinnt $\mathcal{G}_2(\mathfrak{A}, \mathfrak{B})$ und Spoiler gewinnt $\mathcal{G}_3(\mathfrak{A}, \mathfrak{B})$.

3.11 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:

$$\textit{Duplicator gewinnt das Spiel } \mathcal{G}_m(\mathfrak{A}, \mathfrak{B}) \quad \Longleftrightarrow \quad |A| = |B| \ \textit{ oder } \ |A|, |B| > 2^m.$$

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 ist.
Für $\mathfrak{C} := \mathfrak{A}$ oder $\mathfrak{C} := \mathfrak{B}$ betrachten wir die folgende auf $C \times C$ definierte *Distanzfunktion*

$$\mathrm{Dist}(a, a') \ := \ |\{\, b \in C \ | \ (a <^{\mathfrak{C}} b \leqslant^{\mathfrak{C}} a') \ \text{oder} \ (a' <^{\mathfrak{C}} b \leqslant^{\mathfrak{C}} a)\}\,|\,.$$

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 $i \in \{0,..,m\}$ 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 $\mathfrak{B}$.
Beweis: *Übung*. $\square$

3.1.3 Der Satz von Ehrenfeucht

Ziel dieses Abschnitts ist 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 3.6 kann man also EF-Spiele verwenden, um zu zeigen, dass bestimmte Probleme nicht FO-definierbar sind.

3.12 Definition. Sei σ eine Signatur, $\mathfrak{A}$ eine σ-Struktur, $k \in \mathbb{N}$, $\vec{a} = a_1,..,a_k$ eine Sequenz von Elementen aus $\mathfrak{A}$ und $\vec{x} = x_1,..,x_k$ 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]

$$\varphi^0_{\mathfrak{A},\vec{a}}(\vec{x}) \ := \ \bigwedge \left\{ \psi(\vec{x}) \ \middle| \ \begin{array}{l} \psi \text{ ist atomare oder negierte atomare } \sigma\text{-Formel mit} \\ \mathit{frei}(\psi) \subseteq \{x_1,..,x_k\} \text{ und } \mathfrak{A} \models \psi[\vec{a}] \end{array} \right\}$$

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 $\mathfrak{A}$.

3.13 Bemerkungen.

(a) In der obigen Definition ist $k{=}0$ erlaubt. Der m-Isomorphietyp ist dann ein *Satz* $\varphi^m_{\mathfrak{A}}$.

(b) Ist $\mathfrak{A}$ aus dem Kontext klar, so schreiben wir manchmal $\varphi^m_{\vec{a}}(\vec{x})$ an Stelle von $\varphi^m_{\mathfrak{A},\vec{a}}(\vec{x})$.

(c) Für alle $k, m \in \mathbb{N}$ ist die Menge

$$m\text{-Typen}_k \ := \ \{\, \varphi^m_{\mathfrak{A},\vec{a}}(\vec{x}) \ \mid \ \mathfrak{A} \text{ ist eine } \sigma\text{-Struktur und } \vec{a} \in A^k \,\}$$

endlich.
(Klar für $m{=}0$, da es nur endlich viele verschiedene *atomare* σ-Formeln über den Variablen $x_1,..,x_k$ gibt. Für $m > 0$ folgt die Endlichkeit dann per Induktion.)

[3] Für eine endliche Menge M von Formeln schreiben wir $\bigwedge M$, um die Formel $\bigwedge_{\psi \in M} \psi$ zu bezeichnen.

(d) $\mathfrak{A} \models \varphi^m_{\mathfrak{A},\vec{a}}[\vec{a}]$.

3.14 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 frei$(\psi) \subseteq \{x_1, \ldots, x_k\}$ und qr$(\psi) \leqslant m$ gilt: $\mathfrak{A} \models \psi[\vec{a}] \iff \mathfrak{B} \models \psi[\vec{b}]$.

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.

$m{=}0$:

$$
\begin{aligned}
&\text{Duplicator gewinnt } \mathcal{G}_0(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b}) \\
\overset{\text{Gewinnbed.}}{\Longleftrightarrow}\quad & \pi \;:\; \vec{a}, (c^{\mathfrak{A}})_{c\in\sigma} \;\mapsto\; \vec{b}, (c^{\mathfrak{B}})_{c\in\sigma} \;\in\; \text{Part}(\mathfrak{A}, \mathfrak{B}) \\
\overset{\text{Wahl von } \varphi^0_{\mathfrak{A},\vec{a}}}{\Longleftrightarrow}\quad & \mathfrak{B} \models \varphi^0_{\mathfrak{A},\vec{a}}[\vec{b}]\,.
\end{aligned}
$$

$m \mapsto m{+}1$:

$$
\begin{aligned}
&\text{Duplicator gewinnt } \mathcal{G}_{m+1}(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b}) \\
\Longleftrightarrow\quad & \text{für alle } a \in A \text{ ex. } b \in B, \text{ so dass Duplicator } \mathcal{G}_m(\mathfrak{A}, \vec{a}a, \mathfrak{B}, \vec{b}b) \text{ gewinnt, und} \\
& \text{für alle } b \in B \text{ ex. } a \in A, \text{ so dass Duplicator } \mathcal{G}_m(\mathfrak{A}, \vec{a}a, \mathfrak{B}, \vec{b}b) \text{ gewinnt} \\
\overset{\text{Ind.ann.}}{\Longleftrightarrow}\quad & \text{für alle } a \in A \text{ gibt es ein } b \in B, \text{ so dass } \mathfrak{B} \models \varphi^m_{\mathfrak{A},\vec{a}a}[\vec{b}, b], \text{ und} \\
& \text{für alle } b \in B \text{ gibt es ein } a \in A, \text{ so dass } \mathfrak{B} \models \varphi^m_{\mathfrak{A},\vec{a}a}[\vec{b}, b]. \\
\Longleftrightarrow\quad & \mathfrak{B} \;\models\; \Big(\bigwedge_{a\in A} \exists x_{k+1}\, \varphi^m_{\mathfrak{A},\vec{a}a}(\vec{x}, x_{k+1}) \;\;\wedge\;\; \forall x_{k+1} \bigvee_{a\in A} \varphi^m_{\mathfrak{A},\vec{a}a}(\vec{x}, x_{k+1}) \Big)[\vec{b}] \\
\overset{\text{Def. } \varphi^{m+1}_{\mathfrak{A},\vec{a}}}{\Longleftrightarrow}\quad & \mathfrak{B} \;\models\; \varphi^{m+1}_{\mathfrak{A},\vec{a}}[\vec{b}].
\end{aligned}
$$

Somit sind wir fertig mit dem Beweis von "(b) $\Longleftrightarrow$ (a)".

"(a) $\Longrightarrow$ (c)": Per Induktion nach m.

$m{=}0$: Wie bei "(b) $\Longleftrightarrow$ (a)".

$m \mapsto m{+}1$: Gemäß Voraussetzung gewinnt Duplicator $\mathcal{G}_{m+1}(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})$. Sei $\psi(\vec{x})$ eine FO-Formel mit *frei*$(\psi) \subseteq \{x_1, \ldots, x_k\}$ 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 nur noch den Fall betrachten, dass ψ von der Form

$$\psi \;=\; \exists x_{k+1}\, \chi(\vec{x}, x_{k+1})$$

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. $\mathfrak{B} \models \psi[\vec{b}\,]$. $\square$

3.15 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 3.14 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 3.10 (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.

3.16 Bemerkung. Aus Theorem 3.14 und Bemerkung 3.13 (c) 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. σ-STRUKTUREN ist also eine Vereinigung von *endlich* vielen Äquivalenzklassen von $\equiv_m$. Die Äquivalenzklasse, zu der eine gegebene Struktur $\mathfrak{A}$ gehört, wird dabei durch den Satz $\varphi_{\mathfrak{A}}^m$ definiert.

Die nächste Folgerung aus dem Satz von Ehrenfeucht wird oft benutzt um zu zeigen, dass bestimmte Probleme *nicht* FO-definierbar sind.

3.17 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 3.14.

"(b) $\Longrightarrow$ (a)": Das ist gerade die Aussage von Proposition 3.6.

"(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

$$\psi \ := \ \bigvee \{ \varphi_{\mathfrak{A}}^m \ \mid \ \mathfrak{A} \text{ ist eine } \sigma\text{-Struktur mit } \mathfrak{A} \in \mathsf{C} \ \}$$

definiert. Es gilt also: $\mathsf{C} = \mathrm{Mod}_{\mathsf{S}}(\psi)$. □

Als Anwendung erhalten wir, dass die folgenden Probleme nicht FO-definierbar sind:

3.18 Korollar. *Für die Strukturklassen*

$\mathsf{S}_<$	:	*Klasse aller endlichen linearen Ordnungen*,
$\mathsf{Even}_<$	:	*Klasse aller endlichen linearen Ordnungen gerader Länge*,
UGraphs	:	*Klasse aller endlichen ungerichteten Graphen,*
Conn	:	*Klasse aller endlichen ungerichteten zusammenhängenden Graphen*,
RGraphs	:	*Klasse aller endlichen Graphen $G = (V, E^G, S^G, T^G)$ mit $S^G, T^G \subseteq V$*,
Reach	:	*Klasse aller $G \in$* RGraphs*, in denen es einen Pfad von einem Knoten in S^G zu einem Knoten in T^G gibt*

gilt:

(a) $\mathsf{Even}_<$ *ist nicht* FO*-definierbar in* $\mathsf{S}_<$.

(b) Conn *ist nicht* FO*-definierbar in* $\mathsf{UGraphs}$.

(c) Reach *ist nicht* FO*-definierbar in* $\mathsf{RGraphs}$.

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 3.11 wissen wir, dass Duplicator das m-Runden EF-Spiel auf $\mathfrak{A}_m$ und $\mathfrak{B}_m$ gewinnt. Gemäß Korollar 3.17 ist daher $\mathsf{Even}_<$ nicht FO-definierbar in $\mathsf{S}_<$.
(b): Folgt direkt aus (a) und Beispiel 2.78.
(c): Folgt genauso aus (a) und der Reduktion aus Beispiel 2.72 (a). □

3.1.4 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 sehr nützlich.

3.19 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.

3.20 Definition (m-Isomorphie). Sei σ eine Signatur und $m \in \mathbb{N}$. 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 $\varnothing \neq I_j \subseteq \mathrm{Part}(\mathfrak{A}, \mathfrak{B})$ (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 $a \in \mathrm{def}(q)$ (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 $b \in \mathrm{bild}(q)$ (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 schreiben wir $(I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$ und sagen "$\mathfrak{A}$ und $\mathfrak{B}$ sind m-isomorph vermöge $(I_j)_{j\leqslant m}$".

3.21 Theorem. *Sei σ eine Signatur. $\mathfrak{A}$ und $\mathfrak{B}$ seien σ-Strukturen, $k, m \in \mathbb{N}$ und $\vec{a} = a_1, .., a_k \in A$ und $\vec{b} = b_1, .., b_k \in B$. Dann sind äquivalent:*

(a) Duplicator gewinnt $\mathcal{G}_m(\mathfrak{A}, \vec{a}, \mathfrak{B}, \vec{b})$.

(b) $\left(\vec{a}, (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}, (c^{\mathfrak{B}})_{c\in\sigma}\right) \in W_m(\mathfrak{A}, \mathfrak{B})$ *und* $\big(W_j(\mathfrak{A}, \mathfrak{B})\big)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$.

(c) Es gibt $(I_j)_{j\leqslant m}$, so dass

$$\left(\vec{a}, (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}, (c^{\mathfrak{B}})_{c\in\sigma}\right) \in I_m \quad \text{und} \quad (I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}\,.$$

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 $\left(\vec{a}, (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}, (c^{\mathfrak{B}})_{c\in\sigma}\right)$. 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 $i \in \{0, .., m\}$ gilt:

$(*)_i$: Sind $a'_1, .., a'_i$ bzw. $b'_1, .., b'_i$ die in den Runden $1, .., i$ in A bzw. B gewählten Elemente, so gibt es einen partiellen Isomorphismus $p \in I_{m-i}$, so dass $a_1, .., a_k$, $a'_1, .., a'_i \in \text{def}(p)$ und

$$p(a_j) = b_j \quad \text{für alle } j \in \{1, .., k\} \quad \text{und}$$
$$p(a'_j) = b'_j \quad \text{für alle } j \in \{1, .., i\}.$$

$i = 0$: $(*)_0$ gilt, da $\left(\vec{a}, (c^{\mathfrak{A}})_{c\in\sigma} \mapsto \vec{b}, (c^{\mathfrak{B}})_{c\in\sigma}\right)$.
$i \mapsto i{+}1$: Sei p der partielle Isomorphismus aus I_{m-i}, der laut Induktionsannahme 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 3.14 und Theorem 3.21 ergibt sich:

3.22 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 (c) und (d) ist als *Satz von Fraïssé* (1954) bekannt.

3.23 Beispiel. Für jedes $\ell \in \mathbb{N}_{\geqslant 1}$ sei G_ℓ ein ungerichteter Kreis der Länge $\ell{+}1$, d.h. G_ℓ hat Knotenmenge $\{0,..,\ell\}$ und Kantenmenge

$$E^{G_\ell} := \{(i,i{+}1) \mid i<\ell\} \cup \{(\ell,0)\} \cup \{(i{+}1,i) \mid i<\ell\} \cup \{(0,\ell)\}.$$

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 $\ell{+}1$ und $k{+}1$.
Wir zeigen, dass für alle $m \in \mathbb{N}$ gilt:

$$\text{Sind } \ \ell,k \geqslant 2^m, \ \text{ so gilt } \ G_\ell \cong_m G_{\ell,k}, \ \text{ d.h.}$$

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 $\text{Dist}_G(\cdot,\cdot)$ die Distanzfunktion, also

$$\text{Dist}_G(u,v) := \begin{cases} 0\,, & \text{falls } u=v, \\ \infty\,, & \text{falls } u \neq v \text{ und es in } G \text{ keinen Pfad von } u \text{ nach } v \text{ gibt,} \\ \min\{d \mid \text{es gibt in } G \text{ einen Pfad der Länge } d \text{ von } u \text{ nach } v\}, & \text{sonst.} \end{cases}$$

Für jedes $j \in \{0,..,m\}$ sei I_j die Menge aller partiellen Isomorphismen p von G_ℓ nach $G_{\ell,k}$, für die gilt:

- $|\text{def}(p)| \leqslant m{-}j$, und
- für alle $a,a' \in \text{def}(p)$ gilt:
  $\text{Dist}_{G_\ell}(a,a') = \text{Dist}_{G_{\ell,k}}(p(a),p(a'))$ oder $\text{Dist}_{G_\ell}(a,a'),\ \text{Dist}_{G_{\ell,k}}(p(a),p(a')) \geqslant 2^{j+1}$.

I_m besteht gerade aus der Abbildung "$\varnothing$", deren Definitionsbereich leer ist. Per Induktion kann man (ähnlich wie im Beweis von Satz 3.11) nachweisen, dass I_j für jedes $j \in \{m,m{-}1,..,0\}$ die Hin- und die Her-Eigenschaft hat und dass $I_j \neq \varnothing$.
Insgesamt haben wir damit gezeigt, dass

$$(I_j)_{j \leqslant m} \ : \ G_\ell \cong_m G_{\ell,k}\,.$$

$\square$

3.1.5 Der Satz von Hanf

3.24 Definition (Gaifman-Graph, Distanzfunktion, Nachbarschaft).
Sei σ eine relationale 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

$$E^{\mathcal{G}(\mathfrak{A})} := \left\{ (u,v) \;\middle|\; \begin{array}{l} u \neq v \text{ und es gibt ein } R \in \sigma \text{ und ein Tupel } \vec{a} \in R^{\mathfrak{A}}, \text{ so} \\ \text{dass } u \text{ und } v \text{ als Komponenten in } \vec{a} \text{ vorkommen} \end{array} \right\}.$$

(b) Die *Distanzfunktion* $\mathrm{Dist}_{\mathfrak{A}}(\cdot,\cdot) : A \times A \to \mathbb{N}$ ist definiert durch

$$\mathrm{Dist}_{\mathfrak{A}}(u,v) := \begin{cases} 0\,, & \text{falls } u = v, \\ \infty\,, & \text{falls } u \neq v \text{ und es in } \mathcal{G}(\mathfrak{A}) \text{ keinen Pfad von } u \text{ nach } v \text{ gibt}, \\ \min\{\ell \in \mathbb{N} \mid \text{es gibt in } \mathcal{G}(\mathfrak{A}) \text{ einen Pfad der Länge } \ell \text{ von } u \text{ nach } v\}, & \text{sonst.} \end{cases}$$

(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

$$N_{\mathfrak{A}}(r,a) := \{b \in \mathfrak{A} \mid \mathrm{Dist}_{\mathfrak{A}}(a,b) \leqslant r\}.$$

Für ein $k \in \mathbb{N}$ und k Elemente $\vec{a} = a_1,\ldots,a_k$ aus A ist die r-Umgebung

$$N_{\mathfrak{A}}(r,\vec{a}) := N_{\mathfrak{A}}(r,a_1) \cup \cdots \cup N_{\mathfrak{A}}(r,a_k).$$

Wir schreiben $\tilde{N}_{\mathfrak{A}}(r,\vec{a})$ für die durch die Menge $N_{\mathfrak{A}}(r,\vec{a})$ induzierte Substruktur von $\mathfrak{A}$, d.h.

$$\tilde{N}_{\mathfrak{A}}(r,\vec{a}) := \mathfrak{A}_{|N_{\mathfrak{A}}(r,\vec{a})} = \Big(N_{\mathfrak{A}}(r,\vec{a}),\ \big(R^{\mathfrak{A}} \cap N_{\mathfrak{A}}(r,\vec{a})^{ar(R)}\big)_{R\in\sigma} \Big).$$

3.25 Bemerkung. Für zwei beliebige Elemente $b, b' \in N_{\mathfrak{A}}(r,a)$ ist $\mathrm{Dist}_{\mathfrak{A}}(b,b') \leqslant 2{\cdot}r$.

3.26 Definition (r-Umgebungstyp Typ$(r,a,\mathfrak{A})$).
Sei σ eine relationale Signatur, $\mathfrak{A}$ eine σ-Struktur, $r \in \mathbb{N}$ und $a \in A$. Der r-*Umgebungstyp* Typ$(r,a,\mathfrak{A})$ *von a in* $\mathfrak{A}$ ist der *Isomorphietyp* der $(\sigma\dot\cup\{c\})$-Struktur $\big(\tilde{N}_{\mathfrak{A}}(r,a),a\big)$. D.h. Typ$\big(r,a,\mathfrak{A}\big)$ ist die Klasse aller $(\sigma\dot\cup\{c\})$-Strukturen $\mathfrak{C}$, die isomorph zur Struktur $\big(\tilde{N}_{\mathfrak{A}}(r,a),a\big)$ sind.

Insbesondere gilt für ein Element $b \in A$, dass a und b genau dann denselben r-Umgebungstyp in $\mathfrak{A}$ haben, wenn

$$\Big(\tilde{N}_{\mathfrak{A}}(r,a),a\Big) \cong \Big(\tilde{N}_{\mathfrak{A}}(r,b),b\Big).$$

3.27 Theorem (Satz von Hanf).
Sei σ eine relationale 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 3^m-Umgebung eines Elements in $\mathfrak{A}$ oder $\mathfrak{B}$ weniger als e Elemente hat,

und

(b) für alle 3^m-Umgebungstypen ρ ist

(i) $|\{a \in A : \text{Typ}(3^m, a, \mathfrak{A}) = \rho\}| \; = \; |\{b \in B : \text{Typ}(3^m, b, \mathfrak{B}) = \rho\}|$

oder

(ii) $|\{a \in A : \text{Typ}(3^m, a, \mathfrak{A}) = \rho\}|, \; |\{b \in B : \text{Typ}(3^m, b, \mathfrak{B}) = \rho\}| \; > \; m{\cdot}e,$

dann ist

$$\mathfrak{A} \equiv_m \mathfrak{B},$$

d.h. $\mathfrak{A}$ und $\mathfrak{B}$ lassen sich nicht durch FO$[\sigma]$*-Sätze der Quantorentiefe $\leqslant m$ unterscheiden.*

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.

Vorbemerkungen:

(1) Haben zwei Elemente a und b denselben r-Umgebungstyp, so haben sie auch für jede Zahl $r' < r$ denselben r'-Umgebungstyp.

(2) Da $\mathfrak{A}$ und $\mathfrak{B}$ die Voraussetzungen des Theorems erfüllen, gilt wegen (1) für jede Zahl $r \leqslant 3^m$ und jeden r-Umgebungstyp ρ, dass $\mathfrak{A}$ und $\mathfrak{B}$ die gleiche Anzahl von Elementen vom r-Umgebungstyp ρ haben oder jede der beiden Strukturen mehr als $m{\cdot}e$ Elemente vom r-Umgebungstyp ρ hat.

(3) Für ein Tupel $\vec{a} = a_1, .\,., a_k$ von Elementen aus A schreiben wir $|\vec{a}|$, um die *Länge* des Tupels zu bezeichnen, also $|\vec{a}| = k$.

Um zu zeigen, dass $\mathfrak{A} \equiv_m \mathfrak{B}$, nutzen wir Korollar 3.22 und zeigen, dass es eine Sequenz $(I_j)_{j=0,.\,.,m}$ gibt, so dass

$$(I_j)_{j \leqslant m} \; : \; \mathfrak{A} \cong_m \mathfrak{B}.$$

Für $j \in \{0, .\,., m\}$ wählen wir dazu

$$I_j \; := \; \left\{ \big(\vec{a} \mapsto \vec{b}\big) \in \text{Part}(\mathfrak{A}, \mathfrak{B}) \;\middle|\; \begin{array}{l} \vec{a} = a_1, .\,., a_{|\vec{a}|} \in A, \; \vec{b} = b_1, .\,., b_{|\vec{b}|} \in B, \\ 0 \leqslant |\vec{a}| = |\vec{b}| \leqslant m{-}j, \\ \big(\tilde{N}_{\mathfrak{A}}(3^j, \vec{a}), \vec{a}\big) \cong \big(\tilde{N}_{\mathfrak{B}}(3^j, \vec{b}), \vec{b}\big) \end{array} \right\}.$$

Für $j = m$ beachte man, dass $m{-}j = 0$ ist und daher $|\vec{a}| = |\vec{b}| = 0$. In diesem Fall sind $\vec{a}$ und $\vec{b}$ also "leere" Tupel, $N_{\mathfrak{A}}(3^j, \vec{a}) = \varnothing = N_{\mathfrak{B}}(3^j, \vec{b})$, und I_m besteht aus der Abbildung

"$\varnothing$", deren Definitionsbereich leer ist.
Wir wollen zeigen, dass $(I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$. D.h. wir müssen zeigen, dass $I_j \neq \varnothing$ für alle $j \leqslant m$, und dass das System $(I_j)_{j\leqslant m}$ die Hin- und die Her-Eigenschaft hat.
Da I_m den partiellen Isomorphismus "$\varnothing$" enthält, ist $I_m \neq \varnothing$. Es reicht daher, im Folgenden zu zeigen, dass das System $(I_j)_{j\leqslant m}$ die Hin- und die Her-Eigenschaft hat (denn daraus folgt per Induktion dann auch für alle $j \in \{m, m-1, .., 0\}$, dass $I_j \neq \varnothing$).

Wir beweisen die Hin- und die Her-Eigenschaft induktiv für alle $j < m$.
Sei dazu $m > j \geqslant 0$ und $p := \big(\vec{a} \mapsto \vec{b}\big) \in I_{j+1}$. Gemäß der Definition von I_{j+1} gibt es einen Isomorphismus

$$\pi \;:\; \big(\tilde{N}_{\mathfrak{A}}(3^{j+1}, \vec{a}), \vec{a}\big) \;\cong\; \big(\tilde{N}_{\mathfrak{B}}(3^{j+1}, \vec{b}), \vec{b}\big)\,.$$

<u>*Hin-Eigenschaft:*</u> Sei $a \in A$.
Ziel: Finde eine Erweiterung q von p, so dass $a \in \mathrm{def}(q)$ und $q \in I_j$.

<u>Fall 1:</u> $a \in N_{\mathfrak{A}}(2{\cdot}3^j, \vec{a})$, d.h. $N_{\mathfrak{A}}(3^j, a) \subseteq N_{\mathfrak{A}}(3{\cdot}3^j, \vec{a})$ und $N_{\mathfrak{A}}(3^j, \vec{a}a) \subseteq N_{\mathfrak{A}}(3^{j+1}, \vec{a})$.
Insbesondere liegt a im Definitionsbereich des Isomorphismus π. Wir wählen $b := \pi(a)$.
Die Einschränkung π' von π auf $N_{\mathfrak{A}}(3^j, \vec{a}a)$ ist dann ein Isomorphismus

$$\pi' \;:\; \big(\tilde{N}_{\mathfrak{A}}(3^j, \vec{a}a), \vec{a}a\big) \;\cong\; \big(\tilde{N}_{\mathfrak{B}}(3^j, \vec{b}b), \vec{b}b\big)\,.$$

Daher ist $q := \big(\vec{a}, a \mapsto \vec{b}, b\big) \in I_j$.

<u>Fall 2:</u> $a \notin N_{\mathfrak{A}}(2{\cdot}3^j, \vec{a})$, d.h. $N_{\mathfrak{A}}(3^j, a) \cap N_{\mathfrak{A}}(3^j, \vec{a}) = \varnothing$.
Sei $\rho :=$ Typ$(3^j, a, \mathfrak{A})$ der 3^j-Umbegungstyp von a in $\mathfrak{A}$. Wegen

$$\pi \;:\; \big(\tilde{N}_{\mathfrak{A}}(3{\cdot}3^j, \vec{a}), \vec{a}\big) \;\cong\; \big(\tilde{N}_{\mathfrak{B}}(3{\cdot}3^j, \vec{b}), \vec{b}\big)$$

haben $\tilde{N}_{\mathfrak{A}}(2{\cdot}3^j, \vec{a})$ und $\tilde{N}_{\mathfrak{B}}(2{\cdot}3^j, \vec{b})$ dieselbe Anzahl von Elementen mit 3^j-Umbebungstyp ρ, und zwar höchstens

$$|\vec{a}| \cdot (\text{max. Anzahl von Elementen in einer } 2{\cdot}3^j\text{-Umgebung}) \;\overset{\text{Vor. (a)}}{\leqslant}\; |\vec{a}| \cdot e \;\leqslant\; m{\cdot}e.$$

Außerhalb von $N_{\mathfrak{A}}(2{\cdot}3^j, \vec{a})$ gibt es in A außerdem noch mindestens ein weiteres Element vom 3^j-Umgebungstyp ρ, nämlich das Element a. Wegen Vorbemerkung (2) muss es also auch außerhalb von $N_{\mathfrak{B}}(2{\cdot}3^j, \vec{b})$ in B noch mindestens ein Element b geben, dessen 3^j-Umgebungstyp ρ ist.
Wir wählen die Erweiterung q von $p = \big(\vec{a} \mapsto \vec{b}\big)$ mit

$$q \;:=\; \big(\vec{a}a \mapsto \vec{b}b\big).$$

Um nachzuweisen, dass $q \in I_j$ ist, reicht es zu zeigen, dass es einen Isomorphismus

$$\pi' \;:\; \big(\tilde{N}_{\mathfrak{A}}(3^j, \vec{a}a), \vec{a}a\big) \;\cong\; \big(\tilde{N}_{\mathfrak{B}}(3^j, \vec{b}b), \vec{b}b\big)$$

gibt. Sei $\pi_{\vec{a}}$ die Einschränkung von π auf $N_{\mathfrak{A}}(3^j, \vec{a})$, d.h.

$$\pi_{\vec{a}} \ : \ \big(\tilde{N}_{\mathfrak{A}}(3^j, \vec{a}), \vec{a}\big) \ \cong \ \big(\tilde{N}_{\mathfrak{B}}(3^j, \vec{b}), \vec{b}\big) \, .$$

Wegen $\text{Typ}(3^j, a, \mathfrak{A}) = \rho = \text{Typ}(3^j, b, \mathfrak{B})$ gibt es außerdem einen Isomorphismus

$$\pi_a \ : \ \big(\tilde{N}_{\mathfrak{A}}(3^j, a), a\big) \ \cong \ \big(\tilde{N}_{\mathfrak{B}}(3^j, b), b\big) \, .$$

Wegen

$$\text{def}(\pi_{\vec{a}}) \cap \text{def}(\pi_a) = \varnothing \quad \text{und} \quad \text{bild}(\pi_{\vec{a}}) \cap \text{bild}(\pi_b) = \varnothing$$

ist $\pi' := \pi_{\vec{a}} \,\dot{\cup}\, \pi_a$ der gesuchte Isomorphismus

$$\pi' \ : \ \big(\tilde{N}_{\mathfrak{A}}(3^j, \vec{a}a), \vec{a}a\big) \ \cong \ \big(\tilde{N}_{\mathfrak{B}}(3^j, \vec{b}b), \vec{b}b\big) \, .$$

Insgesamt haben wir gezeigt, dass das System $(I_j)_{j \leqslant m}$ die Hin-Eigenschaft hat.

Her-Eigenschaft: Analog.

Insgesamt erhalten wir, dass $(I_j)_{j \leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$. Mit Korollar 3.22 folgt $\mathfrak{A} \equiv_m \mathfrak{B}$. □

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. Aus dem Satz von Hanf folgt, dass alle FO-Sätze der Quantorentiefe m in dem Sinne "lokal" sind, dass sie nur über Umgebungen vom Radius 3^m "sprechen können". Im Folgenden wird diese "Lokalität" der Logik erster Stufe etwas genauer dargestellt.

3.28 Definition (Hanf-Lokalität). Sei σ eine relationale 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:

$$\big(\tilde{N}_{\mathfrak{A}}(r, a), a\big) \ \cong \ \big(\tilde{N}_{\mathfrak{B}}(r, f(a)), f(a)\big) \, .$$

(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:

$$\text{Falls } \mathfrak{A} \leftrightarrows_r \mathfrak{B}\,, \quad \text{so} \quad \big(\mathfrak{A} \in \mathsf{C} \iff \mathfrak{B} \in \mathsf{C}\big) \, .$$

Die folgende einfache Folgerung aus dem Satz von Hanf besagt, dass jede FO-definierbare Klasse endlicher Strukturen Hanf-lokal ist.

3.29 Theorem (Hanf-Lokalität von FO). *Sei σ eine relationale Signatur und* S *eine Klasse endlicher σ-Strukturen. Dann gilt für jeden* FO$[\sigma]$*-Satz φ: $\mathrm{Mod}_{\mathsf{S}}(\varphi)$ ist Hanf-lokal in* S.

Beweis: Sei φ ein FO$[\sigma]$-Satz und $\mathsf{C} := \mathrm{Mod}_{\mathsf{S}}(\varphi)$. Sei $m := qr(\varphi)$ die Quantorentiefe von φ. Wir wählen $r := 3^m$.
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 $r = 3^m$ gibt es eine Bijektion $f : A \to B$, so dass für alle $a \in A$ gilt:

$$\big(\tilde{N}_{\mathfrak{A}}(3^m, a), a\big) \;\cong\; \big(\tilde{N}_{\mathfrak{B}}(3^m, f(a)), f(a)\big)\,.$$

Da f eine *Bijektion* ist, gilt also für jeden 3^m-Umgebungstyp ρ, dass

$$|\{a \in A : \mathrm{Typ}(3^m, a, \mathfrak{A}) = \rho\}| \quad = \quad |\{b \in B : \mathrm{Typ}(3^m, b, \mathfrak{B}) = \rho\}|\,.$$

Somit ist Bedingung *(b)(i)* des Satzes von Hanf erfüllt. Da $\mathfrak{A}$ und $\mathfrak{B}$ endlich sind, ist auch die Bedingung *(a)* erfüllt. Aus dem Satz von Hanf 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 also insbesondere: $\mathfrak{A} \models \varphi \iff \mathfrak{B} \models \varphi$. ☐

3.30 Bemerkung. Indem man zeigt, dass eine Klasse C *nicht Hanf-lokal* in S ist, kann man (unter Verwendung von Theorem 3.29) 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.

3.31 Beispiel.
Conn ist[4] nicht FO-definierbar in $\mathsf{UGraphs}$.

Beweis: Gemäß Theorem 3.29 reicht es zu zeigen, dass Conn nicht Hanf-Lokal in $\mathsf{UGraphs}$ 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 $\ell{+}1$ Knoten besteht, wobei $\ell > 2r$ ist. D.h. $\mathfrak{B}$ ist gerade der Graph $G_{\ell,\ell}$ aus Beispiel 3.23. Wegen $\ell > 2r$ sieht jede r-Umgebung eines Knotens b von B folgendermaßen aus:

$$\overbrace{\bullet - \bullet - \cdots - \bullet - \bullet}^{r} - \bullet_b - \overbrace{\bullet - \bullet - \cdots - \bullet - \bullet}^{r}$$

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, also die Struktur $G_{2\ell+1}$ aus Beispiel 3.23. Jede r-Umgebung eines Knotens a von A sieht folgendermaßen aus:

$$\overbrace{\bullet - \bullet - \cdots - \bullet - \bullet}^{r} - \bullet_a - \overbrace{\bullet - \bullet - \cdots - \bullet - \bullet}^{r}.$$

[4] Zur Erinnerung: $\mathsf{UGraphs}$ ist die Klasse aller endlichen ungerichteten Graphen, Conn ist die Klasse aller zusammenhängenden endlichen ungerichteten Graphen.

Sie ist also isomorph zu jeder r-Umgebung eines Knotens b von 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. $\square$

3.1.6 Der Satz von Gaifman

In diesem Kapitel werden wir den Satz von Gaifman beweisen, der besagt, daß die Logik erster Stufe nur "lokale" Eigenschaften von Strukturen definieren kann. Um dies zu präzisieren, benötigen wir zunächst noch einige Bemerkungen und Definitionen.

Sei σ eine endliche, relationale Signatur. Wir definieren für alle $r \in \mathbb{N}$ Formeln $\mathit{dist}_{\leqslant r}(x, y)$ induktiv durch $\mathit{dist}_{\leqslant 0}(x, y) := (x = y)$ und

$$\begin{aligned}\mathit{dist}_{\leqslant r+1}(x, y) \quad := \quad & \mathit{dist}_{\leqslant r}(x, y) \ \vee \exists z[\mathit{dist}_{\leqslant r}(x, z) \ \wedge \\ & \bigvee\nolimits_{R\in\sigma} \left(\exists u_1 \ldots, \exists u_{ar(R)} R u_1 \ldots u_{ar(R)} \ \wedge \bigvee\nolimits_{1\leqslant i,j\leqslant n} (u_i = z \wedge u_j = y)\right)]\end{aligned}$$

Für $k \geqslant 1$ und $\overline{x} := x_1 \ldots x_k$ schreiben wir $\mathit{dist}_{\leqslant r}(\overline{x}, y)$ für $\bigvee_{i=1}^{k} \mathit{dist}_{\leqslant r}(x_i, y)$.

3.32 Lemma. *Sei σ eine endliche relationale Signatur, $\mathfrak{A}$ eine σ-Struktur und $a, b \in A$. Dann gilt für alle $r \in \mathbb{N}$ $\mathfrak{A} \models \mathit{dist}_{\leqslant r}[a, b]$ genau dann, wenn $\mathit{Dist}_{\mathfrak{A}}(a, b) \leqslant r$, d.h. die Distanz von a und b im Gaifman-Graph $\mathcal{G}(\mathfrak{A})$ ist höchstens r.*

Beweis: Übung. $\square$

Als nächstes brauchen wir noch eine Variante des *Relativierungslemmas*, die es erlaubt, Formeln auf r-Umgebungen einzuschränken.

3.33 Lemma. *Zu jeder* FO$[\sigma]$*-Formel $\varphi(\overline{x})$ gibt es eine* FO$[\sigma]$*-Formel $\varphi^{N(r,\overline{x})}(\overline{x})$, so daß für alle σ-Strukturen $\mathfrak{A}$ und alle $\overline{a} \in A^k$ gilt:*

$$\mathfrak{A} \models \varphi^{N(r,\overline{x})}[\overline{a}] \qquad \textit{gdw.} \qquad (\tilde{N}(r, \overline{a}), \overline{a}) \models \varphi[\overline{a}].$$

Beweis: Per Induktion über den Formelaufbau übersetzt man die Formel φ in die Formel $\varphi^{N(r,\overline{x})}(\overline{x})$. Die einzig interessanten Fälle sind die Fälle $\exists z\varphi$ und $\forall z\varphi$. Eine Formel $\exists z\psi$ wird übersetzt zu $\exists z\ (\mathit{dist}_{\leqslant r}(\overline{x}, z) \wedge \psi^{N(r,\overline{x})})$ und $\forall z\psi$ wird zu $\forall z\ (\mathit{dist}_{\leqslant r}(\overline{x}, z) \rightarrow \psi^{N(r,\overline{x})})$.

Die Korrektheit dieser Konstruktion nachzuweisen ist zur Übung empfohlen. $\square$

3.34 Definition. Sei σ eine endliche relationale Signatur.

(a) Ein FO$[\sigma]$-Satz χ heißt *basis-lokal* (manchmal auch *einfach-lokal*), falls χ von der Form $\chi := \exists x_1 \ldots \exists x_l \bigwedge_{i<j} \neg \mathit{dist}_{\leqslant 2r}(x_i, x_j) \wedge \bigwedge_{i=1}^{l} \psi^{N(r,x_i)}(x_i)$ ist, wobei $r, l \in \mathbb{N}$ und $\psi(x) \in$ FO$[\sigma]$.

(b) Ein FO$[\sigma]$-Satz φ heißt *lokal*, falls φ eine Boolesche Kombination von basis-lokalen FO$[\sigma]$-Sätzen ist.

Wir werden nun den Satz von Gaifman beweisen, der besagt, daß jeder Satz der Logik erster Stufe äquivalent zu einem lokalen Satz ist. Dieser Satz gilt sowohl für endliche als auch unendliche Strukturen und wird hier auch in voller Allgemeinheit bewiesen.

3.35 Theorem (Satz von Gaifman). *Sei σ eine endliche relationale Signatur. Jeder* $\mathrm{FO}[\sigma]$-*Satz ist äquivalent zu einem lokalen* $\mathrm{FO}[\sigma]$-*Satz.*

Für den folgenden Beweis fixieren wir die Signatur σ und sprechen einfach von Sätzen φ. Diese sind dann immer in $\mathrm{FO}[\sigma]$. Das folgende Lemma liefert den Kern des Beweises des Satzes von Gaifman.

Lemma A. Für jedes $m \in \mathbb{N}$ gibt es ein $M \in \mathbb{N}$, so daß für alle σ-Strukturen $\mathfrak{A}, \mathfrak{B}$ gilt: Falls $\mathfrak{A}$ und $\mathfrak{B}$ dieselben basis-lokalen $\mathrm{FO}[\sigma]$-Sätze vom Quantorenrang höchstens M erfüllen, dann gilt $\mathfrak{A} \equiv_m \mathfrak{B}$.

Wir verschieben den Beweis des Lemmas und überlegen uns, was das Lemma für den Beweis des Satzes bringt. Offensichtlich gibt es für jedes $M \in \mathbb{N}$ bis auf logische Äquivalenz nur endlich viele verschiedene basis-lokale Sätze vom Quantorenrang höchstens M. Sei $\chi_1^M, \dots, \chi_{h(M)}^M$ eine vollständige Liste dieser Sätze.

Lemma B. Sei φ ein Satz, $m := \mathrm{qr}(\varphi)$ und sei M wie in Lemma A. Dann ist φ äquivalent zum lokalen Satz

$$\tilde{\varphi} := \bigvee \left\{ \Big(\bigwedge_{i\in I} \chi_i^M \wedge \bigwedge_{i\notin I} \neg\chi_i^M\Big) : \begin{array}{l} \text{es existiert } I \subseteq \{1,\dots,h(M)\} \\ \text{und eine } \sigma\text{-Struktur } \mathfrak{A}, \text{ so daß } \mathfrak{A} \models \varphi \\ \text{und für alle } 1 \leqslant i \leqslant M\colon \mathfrak{A} \models \chi_i^M \iff i \in I \end{array} \right\}$$

Beweis: Sei $\mathfrak{B}$ eine σ-Struktur. Wir zeigen, daß $\mathfrak{B} \models \varphi$ genau dann, wenn $\mathfrak{B} \models \tilde{\varphi}$. Die Hinrichtung ist dabei klar. Zum Beweis der Rückrichtung gelte also $\mathfrak{B} \models \tilde{\varphi}$. Also existiert eine Indexmenge $I \subseteq \{1,\dots,h(M)\}$ so daß

(1) $\mathfrak{B} \models \bigwedge_{i\in I} \chi_i^M \wedge \bigwedge_{i\notin I} \neg\chi_i^M$, d.h. $\mathfrak{B}$ erfüllt genau die Sätze χ_i^M für die $i \in I$, und

(2) es gibt eine σ-Struktur $\mathfrak{A}$ mit $\mathfrak{A} \models \varphi$ und $\mathfrak{A} \models \chi_i^M$ genau dann, wenn $i \in I$.

Dies bedeutet, daß $\mathfrak{A}$ und $\mathfrak{B}$ dieselben basis-lokalen Sätze vom Quantorenrang höchstens M erfüllen. Nach Lemma A gilt also $\mathfrak{A} \equiv_m \mathfrak{B}$ und somit $\mathfrak{B} \models \varphi$, da $\mathrm{qr}(\varphi) = m$ und $\mathfrak{A} \models \varphi$. □

Aus Lemma B folgt sofort der Satz von Gaifman. Wir müssen nun also nur noch Lemma A beweisen.

Beweis von Lemma A: Die Idee des Beweises ist ähnlich zum Beweis des Satzes von Hanf, allerdings ein wenig schwieriger. Zunächst erinnern wir an die Definition der n-Hintikka-Formel $\varphi^n_{\mathfrak{A},\overline{a}}$. Diese Formel war so definiert, daß für alle $\mathfrak{A}', \overline{a}'$ gilt: $\mathfrak{A}' \models \varphi^n_{\mathfrak{A},\overline{a}}[\overline{a}']$ genau dann, wenn $(\mathfrak{A}', \overline{a}') \equiv_n (\mathfrak{A}, \overline{a})$. Der Quantorenrang von $\varphi^n_{\mathfrak{A},\overline{a}}$ ist gerade n.

Der Einfachhalt halber schreiben wir im Rest des Beweises $\psi_{\overline{a}}^{7^j,n}(\overline{x})$ für $[\varphi_{\tilde{N}(7^j,\overline{a}),\overline{a}}(\overline{x})]^{N(7^j,\overline{x})}$. Die Formel besagt in einer Struktur $\mathfrak{A}'$, daß die 7^j-Umgebungen von $\overline{x}$ in $\mathfrak{A}'$ bzgl. $\equiv_n$ äquivalent zur 7^j-Umgebung von $\overline{a}$ in $\mathfrak{A}$ ist.

Wir definieren induktiv eine Funktion $g : \{0,\ldots,m\} \to \mathbb{N}$. Als Verankerung wählen wir $g(0) := 1$. Für den Induktionsschluß von $g(j)$ auf $g(j+1)$ werden im Verlaufe des Beweises verschiedene Bedingungen angegeben. Die genauen Werte auszurechnen ist jedoch extrem mühsam und soll an dieser Stelle daher auch nicht erfolgen. Als erste Bedingung an g fordern wir die folgende.

Bedingung 0. Für alle $j < m$ gilt $g(j) < g(j+1)$.

Zu jedem $m \in \mathbb{N}$ definieren wir $M := g(m)$. Seien $\mathfrak{A}$ und $\mathfrak{B}$ zwei σ-Strukturen, die dieselben basis-lokalen Sätze vom Quantorenrang höchstens M erfüllen. Wir müssen zeigen, daß $\mathfrak{A}$ und $\mathfrak{B}$ m-äquivalent sind. Gemäß Korollar 3.22 genügt es dazu, $\mathfrak{A} \cong_m \mathfrak{B}$ nachzuweisen. Wir definieren dazu für $j \in \{0,\ldots,m\}$

$$I_j := \{\overline{a} \mapsto \overline{b} \in \mathrm{Part}(\mathfrak{A},\mathfrak{B}) : (\tilde{N}_{\mathfrak{A}}(7^j,\overline{a}),\overline{a}) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j,\overline{b}),\overline{b}) \text{ und } |\overline{a}| = |\overline{b}| \leqslant m - j\}$$

Wie im Beweis des Satzes von Hanf vereinbaren wir für $|\overline{a}| = 0$, daß $(\tilde{N}_{\mathfrak{A}}(7^j,\overline{a}),\overline{a}) = \varnothing$ und $\varnothing \equiv_{g(j)} \varnothing$.

Zu zeigen ist $(I_j)_{j\leqslant m} : \mathfrak{A} \cong_m \mathfrak{B}$. Nach obiger Vereinbarung gilt offensichtlich $\varnothing \mapsto \varnothing \in I_m$ und $I_m \neq \varnothing$. Es bleibt also noch die Hin- und Her-Eigenschaft zu zeigen. Aus Symmetriegründen reicht es, die Hin-Eigenschaft nachzuweisen.

Sei $0 \leqslant j \leqslant m$ und $\overline{a} \mapsto \overline{b} \in I_{j+1}$. Sei weiterhin $a \in A$. Nach Definition von I_{j+1} gilt

$$(\tilde{N}_{\mathfrak{A}}(7^{j+1},\overline{a}),\overline{a}) \equiv_{g(j+1)} (\tilde{N}_{\mathfrak{B}}(7^{j+1},\overline{b}),\overline{b}) \tag{3.1}$$

Wir unterscheiden zwei Fälle, je nachdem wo das neue Element a in Bezug auf die schon gewählten Elemente liegt.

Fall 1. Sei $a \in (N_{\mathfrak{A}}(2\cdot 7^j,\overline{a}))$. Dann gilt

$$(\tilde{N}_{\mathfrak{A}}(7^{j+1},\overline{a}),\overline{a}) \models \exists z \mathit{dist}_{\leqslant 2\cdot 7^j}(\overline{a},z) \wedge \psi_{\overline{a}a}^{7^j,g(j)}(\overline{a},z), \tag{3.2}$$

denn offensichtlich erfüllt a als Wahl für z die Formel. Dies führt zur zweiten Bedingung an die Funktion g.

Bedingung 1. $g(j+1)$ is größer als der Quantorenrang der Formel in (3.2).

Aus (3.1) folgt somit

$$(\tilde{N}_{\mathfrak{B}}(7^{j+1},\overline{b}),\overline{b}) \models \exists z \mathit{dist}_{\leqslant 2\cdot 7^j}(\overline{b},z) \wedge \psi_{\overline{a}a}^{7^j,g(j)}(\overline{b},z),$$

Folglich existiert ein $b \in (N_{\mathfrak{B}}(2\cdot 7^j,\overline{b}))$, so daß $(\tilde{N}_{\mathfrak{B}}(7^{j+1},\overline{b}),\overline{b}) \models \psi_{\overline{a}a}^{7^j,g(j)}(\overline{b},b)$. Somit gilt $(\tilde{N}_{\mathfrak{A}}(7^j,\overline{a}),\overline{a}) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j,\overline{b}),\overline{b})$ und nach Definition von I_j auch $\overline{a}a \mapsto \overline{b}b \in I_j$.

Fall 2. Sei nun $a \notin (N_{\mathfrak{A}}(2 \cdot 7^j, \overline{a}))$, d.h. $(N_{\mathfrak{A}}(7^j, \overline{a})) \cap (N_{\mathfrak{B}}(7^j, \overline{b})) = \varnothing$. Für $1 \leqslant s \in \mathbb{N}$ sei

$$\delta_s^j(x_1, \ldots, x_s) := \bigwedge_{1 \leqslant l < k \leqslant s} \neg \mathit{dist}_{\leqslant 4 \cdot 7^j}(x_l, x_k) \wedge \bigwedge_{l=1}^{s} \psi_a^{7^j, g(j)}(x_l).$$

Die Formel besagt, daß es s verschiedene Elemente $x_1, \ldots, x_s$ gibt, deren Abstand paarweise jeweils größer als $4 \cdot 7^j$ ist, deren 7^j-Umgebungen jedoch alle $g(j)$-äquivalent zur 7^j-Umgebung um a sind.

Sei nun $e_{\mathfrak{A}}$ so gewählt, daß

$$(\tilde{N}_{\mathfrak{A}}(7^{j+1}, \overline{a}), \overline{a}) \models \exists x_1 \ldots \exists x_{e_{\mathfrak{A}}} \delta_{e_{\mathfrak{A}}}^j(x_1, \ldots, x_{e_{\mathfrak{A}}}) \wedge \bigwedge_{l=1}^{e_{\mathfrak{A}}} \mathit{dist}_{\leqslant 2 \cdot 7^j}(\overline{a}, x_l) \tag{3.3}$$

jedoch

$$(\tilde{N}_{\mathfrak{A}}(7^{j+1}, \overline{a}), \overline{a}) \not\models \exists x_1 \ldots \exists x_{e_{\mathfrak{A}}+1} \delta_{e_{\mathfrak{A}}}^j(x_1, \ldots, x_{e_{\mathfrak{A}}+1}) \wedge \bigwedge_{l=1}^{e_{\mathfrak{A}}+1} \mathit{dist}_{\leqslant 2 \cdot 7^j}(\overline{a}, x_l). \tag{3.4}$$

$e_{\mathfrak{A}}$ ist also die maximale Anzahl solcher Elemente, die nicht weiter als $2 \cdot 7^j$ von $\overline{a}$ entfernt sind. Da es in der $2 \cdot 7^j$-Umgebung um ein $a_i \in \overline{a}$ keine zwei Elemente mit Abstand größer als $4 \cdot 7^j$ geben kann, folgt $e_{\mathfrak{A}} \leqslant |\overline{a}| \leqslant m - (j+1) \leqslant m$.

Analog zur Definition von $e_{\mathfrak{A}}$ definieren wir $e_{\mathfrak{B}}$ als die entsprechende Zahl in $\mathfrak{B}$. Nach (3.1) gilt $(\tilde{N}_{\mathfrak{A}}(7^{j+1}, \overline{a}), \overline{a}) \equiv_{g(j+1)} (\tilde{N}_{\mathfrak{B}}(7^{j+1}, \overline{b}), \overline{b})$. Weiterhin stellen wir folgende Bedingung an die Funktion g.

> **Bedingung 2.** $g(j+1)$ ist größer als der Quantorenrang der Formeln in (3.3) und (3.4).

Es folgt, daß $(\tilde{N}_{\mathfrak{B}}(7^{j+1}, \overline{b}), \overline{b})$ die Formel in (3.3) erfüllt, die Formel in (3.4) jedoch nicht. Also gilt $e_{\mathfrak{A}} = e_{\mathfrak{B}}$.

Wir setzen $e := e_{\mathfrak{A}} = e_{\mathfrak{B}}$. Wegen (3.3) gilt auch

$$\mathfrak{A} \models \exists x_1 \ldots \exists x_e \delta_e^j(x_1, \ldots, x_e). \tag{3.5}$$

Gilt ferner auch noch

$$\mathfrak{A} \models \exists x_1 \ldots \exists x_e \exists x_{e+1} \delta_e^j(x_1, \ldots, x_{e+1}), \tag{3.6}$$

so setzen wir $e'_{\mathfrak{A}} := e + 1$. Andernfalls setzen wir $e'_{\mathfrak{A}} := e$. Wiederum wird $e'_{\mathfrak{B}}$ analog in $\mathfrak{B}$ definiert und es gilt $e'_{\mathfrak{A}} = e'_{\mathfrak{B}}$. Dies folgt, da die Formeln in (3.5) und (3.6) basis-lokal sind und der Quantorenrang dieser Formeln höchstens dem Quantorenrang der Formeln in (3.3) und (3.4) entspricht. Wegen Bedingung 2 an g ist $g(j+1)$ also größer als dieser Quantorenrang. Nach Voraussetzung des Lemmas erfüllen $\mathfrak{A}$ und $\mathfrak{B}$ dieselben basis-lokalen

Formeln vom Quantorenrang höchstens $M \geqslant g(j+1)$. Also erfüllt $\mathfrak{B}$ ebenfalls die Formel in (3.5) nicht jedoch die in (3.6). Es folgt $e'_{\mathfrak{A}} = e'_{\mathfrak{B}}$. Wir schreiben $e' := e'_{\mathfrak{A}} = e'_{\mathfrak{B}}$.

Wir unterscheiden nun zwei Fälle, je nachdem ob $e = e'$ oder $e + 1 = e'$.

Fall 2.1 Sei $e = e'$, d.h. alle Elemente $a' \in A$, die $\psi_a^{7^j,g(j)}(x)$ erfüllen, haben Abstand von $\overline{a}$ höchstens $6 \cdot 7^j < 7^{j+1}$. Denn gäbe es ein solches Element a' mit Abstand größer als $6 \cdot 7^j$ von $\overline{a}$, so wären die e Elemente in $(N_{\mathfrak{A}}(2 \cdot 7^j, \overline{a}))$ zusammen mit a' insgesamt $e + 1$ Elemente, die die Formel δ^j_{e+1} erfüllen. Dies wäre aber ein Widerspruch zu $e = e'$.

Offensichtlich erfüllt a die Formel $\psi_a^{7^j,g(j)}(x)$. Es gilt also (mit der Voraussetzung des Falles 2) $a \in (N_{\mathfrak{A}}(6 \cdot 7^j, \overline{a}) \backslash (N_{\mathfrak{A}}(2 \cdot 7^j, \overline{a}))$ und somit

$$(\tilde{N}_{\mathfrak{A}}(7^{j+1}, \overline{a}), \overline{a}) \models \exists z \neg \mathit{dist}_{\leqslant 2 \cdot 7^j}(\overline{a}, z) \wedge \mathit{dist}_{\leqslant 6 \cdot 7^j}(\overline{a}, z) \wedge \psi_a^{7^j,g(j)}(z) \wedge \psi_{\overline{a}}^{7^j,g(j)}(\overline{a}). \quad (3.7)$$

Dies liefert die dritte und letzte Bedingung an die Funktion g.

Bedingung 3. $g(j+1)$ ist größer als der Quantorenrang der Formel in (3.7).

Da nach (3.1) $(\tilde{N}_{\mathfrak{A}}(7^{j+1}, \overline{a}), \overline{a}) \equiv_{g(j+1)} (\tilde{N}_{\mathfrak{B}}(7^{j+1}, \overline{b}), \overline{b})$, folgt

$$(\tilde{N}_{\mathfrak{B}}(7^{j+1}, \overline{b}), \overline{b}) \models \exists z \neg \mathit{dist}_{\leqslant 2 \cdot 7^j}(\overline{b}, z) \wedge \mathit{dist}_{\leqslant 6 \cdot 7^j}(\overline{b}, z) \wedge \psi_a^{7^j,g(j)}(z) \wedge \psi_{\overline{a}}^{7^j,g(j)}(\overline{b}).$$

Es existiert also ein $b \in B$ mit $2 \cdot 7^j < \mathrm{Dist}(\overline{b}, b) \leqslant 6 \cdot 7^j$, so daß $(\tilde{N}_{\mathfrak{A}}(7^j, \overline{a}), \overline{a}) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j, \overline{b}), \overline{b})$ und $(\tilde{N}_{\mathfrak{A}}(7^j, a), a) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j, b), b)$. Ferner gilt $N_{\mathfrak{A}}(7^j, \overline{a}) \cap N_{\mathfrak{A}}(7^j, a) = \varnothing$ sowie $N_{\mathfrak{B}}(7^j, \overline{b}) \cap N_{\mathfrak{B}}(7^j, b) = \varnothing$ und somit $(\tilde{N}_{\mathfrak{A}}(7^j, \overline{a}a), \overline{a}a) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j, \overline{b}b), \overline{b}b)$. Nach Definition von I_j gilt daher $\overline{a}a \mapsto \overline{b}b \in I_j$.

Fall 2.2 Sei $e + 1 = e'$. Dann gilt $\mathfrak{B} \models \exists x_1 \ldots \exists x_{e+1} \delta^j_{e+1}(x_1, \ldots, x_{e+1})$. Es gibt also ein $b \in$ mit $\mathfrak{B} \models \psi_a^{7^j,g(j)}[b]$ und $(N_{\mathfrak{B}}(7^j, \overline{b})) \cap N_{\mathfrak{B}}(7^j, b) \models \varnothing$. Insbesondere gilt also $(\tilde{N}_{\mathfrak{A}}(7^j, a), a) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j, b), b)$. Analog zum Fall 2.1 gilt $(\tilde{N}_{\mathfrak{A}}(7^j, \overline{a}a), \overline{a}a) \equiv_{g(j)} (\tilde{N}_{\mathfrak{B}}(7^j, \overline{b}b), \overline{b}b)$ und somit $\overline{a}a \mapsto \overline{b}b \in I_j$.

Dies schließt den Fall 2.2 und damit den Nachweis der Hin-Eigenschaft ab. Die Her-Eigenschaft folgt aus Symmetriegründen. Es gilt also $(I_j) : \mathfrak{A} \cong_m \mathfrak{B}$ und somit $\mathfrak{A} \equiv_m \mathfrak{B}$. Damit ist Lemma A und gleichzeitig der Satz von Gaifman bewiesen. $\square$

3.2 Ehrenfeucht-Fraïssé Spiele für die existentielle Logik zweiter Stufe

In diesem Kapitel werden wir die Methode der Ehrenfeucht-Fraïssé Spiele auf die existentielle Logik zweiter Stufe erweitern. Gemäß Definition 2.6 haben ESO-Formeln $\varphi(\overline{x})$ die Gestalt $\varphi(\overline{x}) := \exists X_1 \ldots \exists X_k \varphi(\overline{x}, X_1, \ldots, X_k)$, mit $\varphi \in \mathrm{FO}[\sigma \dot\cup \{X_1, \ldots, X_k\}]$. Das Ehrenfeucht-Fraïssé-Spiel wird nun entsprechend dem Formelaufbau auf naheliegende Weise für die Logik ESO erweitert.

3.36 Definition. Seien $l, m \in \mathbb{N}_{\geqslant 1}$ und seien $s_1, \ldots, s_l \in \mathbb{N}_{\geqslant 1}$. Sei σ eine Signatur und $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen. Das $((s_1, \ldots, s_l), m)$-Ehrenfeucht-Fraïssé-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$ wird 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-Spiel $G_m((\mathfrak{A}, S_1, \ldots, S_l), (\mathfrak{B}, S'_1, \ldots, S'_l))$.

3.37 Theorem. *Seien $l, m \in \mathbb{N}_{\geqslant 1}$ und seien $s_1, \ldots, s_l \in \mathbb{N}_{\geqslant 1}$. Sei σ eine Signatur und $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen. Spoiler hat genau dann eine Gewinnstrategie im $((s_1, \ldots, s_l), m)$-Ehrenfeucht-Fraïssé-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$, wenn es einen* ESO$[\tau]$*-Satz $\psi := \exists X_1 \ldots \exists X_l \varphi$ mit $\varphi \in$* FO, *$qr(\varphi) \leqslant m$ und $ar(X_i) = s_i$ gibt, so daß $\mathfrak{A} \models \psi$ aber $\mathfrak{B} \not\models \psi$.*

Beweis: Wir zeigen zunächst die Rückrichtung. Sei $\psi := \exists X_1 \ldots \exists X_l \varphi$ 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 $((s_1, \ldots, s_l), m)$-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 $(\mathfrak{A}, S_1, \ldots, S_l) \models \varphi$. In Phase 1 des Spiels wählt Spoiler diese Relationen. Seien $S'_1 \subseteq B^{s_1}, \ldots, S'_l \subseteq B^{s_l}$ der Antwortzug des Duplicator. Da $\mathfrak{B} \not\models \psi$ gilt $(\mathfrak{B}, S'_1, \ldots, S'_l) \not\models \varphi$. Wegen $qr(\varphi) \leqslant m$ gilt also $(\mathfrak{A}, S_1, \ldots, S_l) \not\equiv_m (\mathfrak{B}, S'_1, \ldots, S'_l)$ und somit hat Spoiler in Phase 2 eine Gewinnstrategie für das Spiel $G_m((\mathfrak{A}, S_1, \ldots, S_l), (\mathfrak{B}, S'_1, \ldots, S'_l))$.

Nun zur Hinrichtung. Nach Voraussetzung hat Spoiler eine Gewinnstrategie im $((s_1, \ldots, s_l), m)$-Ehrenfeucht-Fraïssé-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: $(\mathfrak{A}, S_1, \ldots, S_l) \not\equiv_m (\mathfrak{B}, S'_1, \ldots, S'_l)$. Insbesondere gibt es also für jede Wahl von $S'_1, \ldots, S'_l$ einen FO$[\sigma \dot\cup \{X_1, \ldots, X_l\}]$-Satz $\varphi_{S'_1,\ldots,S'_l}$ vom Quantorenrang höchstens m, so daß $(\mathfrak{A}, S_1, \ldots, S_l) \models \varphi_{S'_1,\ldots,S'_l}$ aber $(\mathfrak{B}, S'_1, \ldots, S'_l) \not\models \varphi_{S'_1,\ldots,S'_l}$. Mit

$$\varphi := \bigwedge \{\varphi_{S'_1,\ldots,S'_l} : S'_1 \subseteq B^{s_1}, \ldots, S'_l \subseteq B^{s_l}\}$$

erhält man einen Satz mit $qr(\varphi) \leqslant m$, $(\mathfrak{A}, S_1, \ldots, S_l) \models \varphi$ aber für kein $S'_1 \subseteq B^{s_1}, \ldots, S'_l \subseteq B^{s_l}$ gilt $(\mathfrak{B}, S'_1, \ldots, S'_l) \models \varphi$. Daraus folgt sofort $\mathfrak{A} \models \exists X_1 \ldots X_l \varphi$ und $\mathfrak{B} \not\models \exists X_1 \ldots \exists X_l \varphi$. $\square$

Eine einfache Folgerung des Theorems liefert nun die Methode, um Nicht-Definierbarkeit in ESO zu zeigen.

3.38 Korollar. *Sei σ eine Signatur, $\mathcal{K}$ eine Klasse von σ-Strukturen. Eine Klasse $\mathcal{C} \subseteq \mathcal{K}$ ist genau dann nicht* ESO*-definierbar in $\mathcal{K}$, wenn es für alle $l, m \in \mathbb{N}_{\geqslant 1}$ und alle $s_1, \ldots, s_l \in \mathbb{N}_{\geqslant 1}$ zwei Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ gibt, so daß Duplicator eine Gewinnstrategie im $((s_1, \ldots, s_l), m)$-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$ hat.*

Bemerkung. 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 Σ^1_1-definierbar ist. Nehmen wir nun ein CO-NP-vollständiges Problem $\mathcal{C}$, so ist $\mathcal{C} \in$ NP genau dann, wenn CO-NP $=$ NP. Es gilt also CO-NP $\neq$ NP gdw. $\mathcal{C} \notin$ NP gdw. $\mathcal{C}$ ist nicht ESO-definierbar gdw. für alle $l, m \in \mathbb{N}_{\geqslant 1}$ und alle $s_1, \dots, s_l \in \mathbb{N}_{\geqslant 1}$ gibt es $\mathfrak{A} \in \mathcal{C}$ und $\mathfrak{B} \notin \mathcal{C}$, so daß Duplicator eine Gewinnstrategie im $((s_1, \dots, s_l), m)$-Ehrenfeucht-Fraïssé-Spiel auf $\mathfrak{A}$ und $\mathfrak{B}$ hat.

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.

3.3 Ajtai-Fagin-Spiele

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 quantifiziert werden darf, eine besondere Rolle ein. Zum einen können viele NP-vollständige Probleme, z.B. 3-Färbbarkeit, das Cliquen- oder Independent-Set-Problem und viele weitere, durch ESO-Formeln beschrieben werden, die nur über Mengen quantifizieren. Andererseits wird 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. Man bezeichnet die Klasse aller ESO-Formeln, deren zweitstufige Quantoren nur über Mengen quantifizieren, als *monadische existenzielle Logik zweiter Stufe (monadisches* ESO*)*. Um die folgenden Spiele zu motivieren, betrachten wir zunächst noch eine Anwendung des Satzes von Hanf und zeigen, daß Graphzusammenhang nicht in monadischem ESO definierbar ist. Das heißt, es gibt keinen Satz $\exists X_1 \dots \exists X_k \psi$, mit $\psi \in$ FO und $ar(X_i) = 1$ für alle i, der genau dann in einem Graphen gilt, wenn dieser zusammenhängend ist. Allerdings kann Graphzusammenhang in der universellen, monadischen Logik zweiter Stufe definiert werden. Denn ein Graph $\mathcal{G} := (V, E)$ ist genau dann zusammenhängend, wenn

$$\mathcal{G} \models \forall P([\exists x Px \land \forall x \forall y((Px \land Exy) \to Py)] \to \forall z Pz).$$

3.39 Theorem. *Die Klasse aller endlichen, zusammenhängenden Graphen kann nicht durch einen Satz der monadischen, existentiellen Logik zweiter Stufe definiert werden.*

Beweis: Wir benötigen zunächst einige Vorbemerkungen. Für jedes $l \geqslant 1$ sei $\mathcal{D}_l := (V_l, E_l)$ ein gerichteter Zyklus der Länge $l+1$, also $V_l := \{0, \dots, l\}$ und $E_l := \{(i, i+1) : 0 \leqslant i < l\} \cup \{(l, 0)\}$.

Sei $\sigma := \{E, X_1, \dots, X_k\}$ die Signatur der durch $X_1, \dots, X_k$ gefärbten Graphen, d.h. E ist binär und alle X_i sind unär. Wir betrachten nun gefärbte Zyklen $\mathfrak{A}_l := (\mathcal{D}_l, X_1, \dots, X_k)$, d.h. $X_i \subseteq V_l$ für alle i. Man beachte, daß die Mengen X_i keinen Einfluß auf den Gaifman-Graph der Strukturen $\mathfrak{A}_l$ haben. Der Gaifman-Graph von $\mathfrak{A}_l$ ist also gerade der ungerichtete Zyklus der Länge $l+1$.

Sei nun $m \in \mathbb{N}$. Wir behaupten, daß es ein $l_m \in \mathbb{N}$ gibt, so daß für alle $l > l_m$ folgendes gilt: In jeder σ-Struktur $\mathfrak{A}_l$ gibt es Elemente $a, b \in V_l$ mit disjunkten aber isomorphen 3^m-Umgebungen. Ist $l \geqslant 2 \cdot 3^m + 1$, so hat offensichtlich jede 3^m-Umgebung eines Elementes genau $2 \cdot 3^m + 1$ Elemente (da die unären Mengen ja die Distanzen nicht ändern). Sei i die Anzahl verschiedener möglicher Isomorphietypen von 3^m-Umgebungen. Für $l_m := (i+1)(2 \cdot 3^m + 1)$ folgt die Behauptung sofort.

Sei $\mathfrak{A}_l := (\mathcal{D}_l, X_1, \ldots, X_k)$ ein gefärbter Zyklus, wobei $l > l_m$, und seien $a, b \in V_l$ wie oben zwei Elemente mit disjunkten aber isomorphen 3^m-Umgebungen. Weiterhin sei a' der Vorgänger von a auf dem Zyklus und analog b' der Vorgänger von b. Wir definieren eine neue Struktur $\mathfrak{A}'_l := (\mathcal{D}'_l, X_1, \ldots, X_k)$ durch $\mathcal{D}'_l := (V, E')$ mit $E' := (E_l \setminus \{(a', a), (b', b)\}) \cup \{(b', a), (a', b)\}$. Das Universum V und die Mengen X_i bleiben unverändert. Die Struktur $\mathfrak{A}'_l$ entsteht also aus $\mathfrak{A}_l$, indem die Kante (a', a) nach b „umgebogen" wird und analog für die Kante (b', b). Da a und b isomorphe aber disjunkte 3^m-Umgebungen in $\mathfrak{A}_l$ haben und diese durch die Modifikation in $\mathfrak{A}'_l$ nicht verändert werden, gilt nach dem Satz von Hanf $\mathfrak{A}_l \equiv_m \mathfrak{A}'_l$. Die Anzahl verschiedenen der 3^m-Umgebungen hat sich schließlich nicht verändert.

Mit diesen Vorbereitungen kann nun der Satz bewiesen werden. Dazu nehmen wir an, es gäbe einen monadischen ESO-Satz $\varphi := \exists X_1 \ldots \exists X_k \psi$, mit $\psi \in \mathrm{FO}[\{E, X_1, \ldots, X_k\}]$, der die Klasse aller endlichen, zusammenhängenden Graphen definiert. Ein Graph $\mathcal{G} := (V, E)$ ist also genau dann zusammenhängend, wenn es Mengen $S_1, \ldots, S_k \subseteq V$ gibt, so daß $(\mathcal{G}, S_1, \ldots, S_k) \models \psi$.

Sei m der Quantorenrang von ψ. Wähle nun l_m und, für ein $l > l_m$, $\mathcal{D}_l$ wie oben. Offensichtlich ist $\mathcal{D}_l$ zusammenhängend und somit gibt es Mengen $S_1, \ldots, S_k \subseteq V$, so daß $(\mathcal{D}_l, S_1, \ldots, S_k) \models \psi$. Nach den Vorbemerkungen gilt nun $(\mathcal{D}_l, S_1, \ldots, S_k) \equiv_m (\mathcal{D}'_l, S_1, \ldots, S_k)$ und somit $(\mathcal{D}'_l, S_1, \ldots, S_k) \models \psi$. Also gilt auch $\mathcal{D}'_l \models \varphi$. Da $\mathcal{D}'_l$ nicht zusammenhängend ist, ergibt dies den gewünschten Widerspruch zur Annahme. $\square$

Wie man leicht zeigen kann, ist Graphzusammenhang durch eine Formel der Form $\exists R \psi$ definierbar, wobei R binär ist. Erlaubt man also statt nur unärer auch binäre Relationssymbole, wird die Ausdrucksstärke der Logik echt erhöht.

Kommen wir nun zurück zur Definition der Ajtai-Fagin-Spiele. Dazu betrachten wir noch einmal den letzten Abschnitt des gerade geführten Beweises. Für gegebenen Quantorenrang m und Anzahl k von unären Relationssymbolen X_i wurde der Beweis wie folgt geführt. Zu zeigen war, daß der Satz φ eben nicht Graphzusammenhang definiert. Dazu haben wir im Prinzip zunächst einen geeigneten, zusammenhängenden Zyklus $\mathcal{D}_l$ gewählt. Für beliebige Mengen $X_1, \ldots, X_k$ über dieser Struktur (die im Beweis durch den Satz φ gegeben wurden) haben wir dann einen zweiten, nicht zusammenhängenden Zyklus $\mathcal{D}'_l$ und Mengen $X'_1, \ldots, X'_k$ gewählt (in unserem Fall galt $X'_i = X_i$), so daß Duplicator danach das m-Runden Spiel $G_m((\mathcal{D}_l, X_1, \ldots, X_k), (\mathcal{D}'_l, X'_1, \ldots, X'_k))$ gewann. (Den Nachweis, daß Duplicator das Spiel gewinnt, haben wir in den Vorbemerkungen mit Hilfe des Satzes von Hanf geführt.) Eine Verallgemeinerung dieser Idee führt zum Begriff der Ajtai-Fagin-Spiele.

3.40 Definition (Ajtai-Fagin-Spiel). Sei σ eine Signatur, $\mathcal{K}$ eine Klasse von σ-Strukturen und $\mathcal{C} \subseteq \mathcal{K}$. Seien weiterhin $l, m \in \mathbb{N}_{\geqslant 1}$. Das (l,m)-*Ajtai-Fagin-Spiel für* $\mathcal{C}$ *auf* $\mathcal{K}$ wird wie folgt gespielt.

Phase 1: Duplicator wählt eine Struktur $\mathfrak{A} \in \mathcal{C}$. Danach wählt Spoiler l Mengen $S_1, \ldots, S_l \subseteq A$.

Phase 2: Duplicator wählt eine Struktur $\mathfrak{B} \in \mathcal{K} \backslash \mathcal{C}$ und l Mengen $S'_1, \ldots, S'_l \subseteq B$.

Phase 3: Spoiler und Duplicator spielen das Spiel $G_m((\mathfrak{A}, S_1, \ldots, S_l), (\mathfrak{B}, S'_1, \ldots, S'_l))$.

Die Korrektheit dieser Definition beweist der nächste Satz, den wir ohne Beweis angeben.

3.41 Theorem. *Sei σ eine Signatur, $\mathcal{K}$ eine Klasse von σ-Strukturen und $\mathcal{C} \subseteq \mathcal{K}$. Es gibt genau dann einen Satz ψ in monadischem* ESO$[\sigma]$ *mit $\mathcal{C} = \mathrm{Mod}_{\mathcal{K}}(\psi)$, wenn es $l, m \in \mathbb{N}_{\geqslant 1}$ gibt, so daß Spoiler eine Gewinnstrategie im (l,m)-Ajtai-Fagin-Spiel für $\mathcal{C}$ auf $\mathcal{K}$ hat.*

Damit schließen wir das Kapitel über Ehrenfeucht-Fraïssé-Spiele für Varianten der Logik zweiter Stufe ab. Die für ESO definierten Spiele können selbstverständlich auf naheliegende Weise für die gesamte Logik zweiter Stufe erweitert werden. Das finden von Gewinnstrategien wird dabei natürlich nicht unbedingt einfacher.

3.4 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.

3.4.1 Die infinitäre Logik $\mathrm{L}_{\infty\omega}$

3.42 Definition. Sei σ eine Signatur. Die Logik $L_{\infty\omega}[\sigma]$ ist induktiv wie folgt definiert.

- $L_{\infty\omega}[\sigma]$ enthält alle atomaren FO$[\sigma]$-Formeln,
- Ist $\varphi \in L_{\infty\omega}[\sigma]$ so auch $\neg\varphi$.
- Ist $\varphi \in L_{\infty\omega}[\sigma]$ und ist x eine Variable so ist auch $\exists x\varphi \in L_{\infty\omega}[\sigma]$ und $\forall x\varphi \in L_{\infty\omega}[\sigma]$.
- Ist $\Psi \subseteq L_{\infty\omega}[\sigma]$ eine Menge von Formeln, so ist $\bigwedge \Psi \in L_{\infty\omega}[\sigma]$ und $\bigvee \Psi \in L_{\infty\omega}[\sigma]$. Hierbei kann Ψ auch unendlich sein.

Die Semantik der Logik 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), $\mathfrak{A} \models \bigvee \Psi$ genau dann, wenn es einen Satz $\psi \in \Psi$ gibt mit $\mathfrak{A} \models \psi$. Offensichtlich ist $L_{\infty\omega}$ eine Erweiterung der Logik erster Stufe.

Wir geben zunächst einige Beispiele für $L_{\infty\omega}$-Formeln.

3.43 Beispiel. Sei für jedes $n \in \mathbb{N}_{\geqslant 1}$ φ_n der Satz

$$\varphi_n := \exists x_1 \ldots \exists x_n \bigwedge_{i \neq j} \neg x_i = x_j \wedge \forall y \bigvee_{i=1}^{n} x_i = y$$

der besagt, daß es genau n Elementen in den Modellen von φ_n gibt.

(1) Für jede Signatur σ definiert $\psi := \bigvee\{\varphi_n : n \in \mathbb{N}\} \in L_{\infty\omega}[\sigma]$ die Klasse aller endlichen σ-Strukturen.

(2) Analog definiert der Satz $\psi_{\text{EVEN}} := \bigvee\{\varphi_n : n \in \mathbb{N}$ und n gerade $\}$ die Klasse aller endlichen Strukturen gerader Kardinalität.

Wir wissen bereits, daß die Klasse aller endlicher Strukturen gerader Kardinalität nicht in FO definierbar ist. Die Logik $L_{\infty\omega}$ ist also echt ausdruckstärker als FO, was angesichts der sehr allgemeiner 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.

3.44 Beispiel. Sei σ eine Signatur und $\mathcal{K}$ eine beliebige Klasse endlicher σ-Strukturen. $\mathcal{K}$ wird definiert durch den $L_{\infty\omega}[\sigma]$-Satz $\varphi_{\mathcal{K}} := \bigvee\{\varphi_{\mathfrak{A}} : \mathfrak{A} \in \mathcal{K}\}$, wobei $\varphi_{\mathfrak{A}} := \varphi_{\mathfrak{A}}^{|A|+1}$ die Hintikka-Formel zu $\mathfrak{A}$ gemäß Definition 3.12 ist.

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.

3.4.2 Das k-Variablen Fragment von FO und $L_{\infty\omega}$

3.45 Definition. Sei σ eine Signatur und sei $k \in \mathbb{N}$. Die Klasse $\text{FO}^k[\sigma]$ besteht aus allen FO$[\sigma]$-Formeln, in denen höchstens k verschiedene Variablen vorkommen.

3.46 Beispiel. Für jedes $l \in \mathbb{N}$ gibt es eine $\text{FO}^2[\{<\}]$-Formel $\psi_l(x)$, so daß für jede linear geordnete Struktur $\mathfrak{A} := (A, <^{\mathfrak{A}})$ und jedes $a \in A$ gilt: $\mathfrak{A} \models \psi_l[a]$ genau dann, wenn a das l-te Element bezüglich der Ordnung $<^{\mathfrak{A}}$ ist. Wir definieren ψ_l induktiv durch $\psi_0(x) := \forall y \neg y < x$ und

$$\psi_{l+1}(x) := \forall y \big(y < x \leftrightarrow \bigvee_{0 \leqslant i \leqslant l} \exists x (x = y \wedge \psi_i(x))\big).$$

Die Konstruktion $\exists x(x = y \wedge \psi_i(x))$ wird benutzt, da ψ_l die Variable x und nicht y als freie Variable enthält.

Die Formel besagt also, daß alle Elemente $y < x$ höchstens den Rang l in der Ordnung haben können. Also kann x höchstens den Rang $l + 1$ haben. Andererseits kann x keinen Rang $\leqslant l$ 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 freie 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 $L_{\infty\omega}$.

3.47 Definition. 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 enthalten. Weiter sei $L^\omega_{\infty\omega}[\sigma] := \bigcup_{k\in\mathbb{N}} L^k_{\infty\omega}$.

$L^\omega_{\infty\omega}$ ist also die Klasse aller $L_{\infty\omega}$-Formeln, in denen nur endlich viele Variablen benutzt werden. Als Vorblick auf Kapitel 4 sei erwähnt, daß sich die bisher behandelten Fixpunktlogiken LFP, IFP, PFP sämtlich in $L^\omega_{\infty\omega}$ einbetten lassen, es gilt also PFP $\leqslant L^\omega_{\infty\omega}$. Insbesondere übertragen sich also Nicht-Definierbarkeits-Resultate für $L^\omega_{\infty\omega}$ auch auf die Fixpunktlogiken.

3.48 Beispiel. Für jede Menge $J \subseteq \mathbb{N}_{\geqslant 1}$ gibt es einen $L^3_{\infty\omega}[\{<\}]$-Satz φ_J, so daß

$$\mathrm{Mod}(\varphi_J) = \{\mathfrak{A} := (A, <^{\mathfrak{A}}) : <^{\mathfrak{A}} \text{ ist eine lineare Ordnung auf } A \text{ und } |A| \in J\}.$$

Sei $\psi_{\text{Ord}} \in \text{FO}^3$ ein Satz, der besagt, daß $<$ eine lineare Ordnung ist. Dann ist ψ_J definiert als

$$\psi_J := \psi_{\text{Ord}} \wedge \bigvee\{\exists x \psi_{l-1}(x) \wedge \neg\exists x \psi_l(x) : l \in J\},$$

wobei ψ_l der Satz aus Beispiel 3.46 ist.

Wie dieses Beispiel zeigt, können also schon in $L^3_{\infty\omega}$ Strukturklassen definiert werden, die nicht in FO definierbar sind (mit beliebig vielen Variablen). Wie der nächste Satz allerdings zeigt, können zwei *endliche* Strukturen, die in $L^k_{\infty\omega}$ getrennt werden können, auch schon in FO^k getrennt werden.

3.49 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. $L^k_{\infty\omega}$-Sätze) erfüllen.

3.50 Theorem. *Für alle endlichen σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ gilt $\mathfrak{A} \equiv_{\mathrm{FO}^k} \mathfrak{B}$ genau dann, wenn $\mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$.*

Beweis: Die Rückrichtung ist klar, da FO $\subseteq L^k_{\infty\omega}$. Wir zeigen die Hinrichtung per Induktion über den Aufbau der $L^k_{\infty\omega}$-Formeln. Dazu zeigen wir, daß es zu jeder $L^k_{\infty\omega}[\sigma]$-Formel $\varphi(\overline{x})$ eine FO$^k[\sigma]$-Formel $\tilde{\varphi}(\overline{x})$ gibt, so daß für alle $\overline{a} \in A$, $\overline{b} \in B$ gilt: $\mathfrak{A} \models \varphi[\overline{a}]$ gdw. $\mathfrak{A} \models \tilde{\varphi}[\overline{a}]$ und $\mathfrak{B} \models \varphi[\overline{b}]$ gdw. $\mathfrak{B} \models \tilde{\varphi}[\overline{b}]$.

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 $\overline{a} \in A$ mit $\mathfrak{A} \models \varphi[\overline{a}]$ wähle eine Formel $\psi_{\overline{a}} \in \Psi$, so daß $\mathfrak{A} \models \psi[\overline{a}]$. Analog wähle für jedes $\overline{b} \in B$ mit $\mathfrak{B} \models \varphi[\overline{b}]$ eine Formel $\psi_{\overline{b}} \in \Psi$, so daß $\mathfrak{B} \models \psi_{\overline{b}}[\overline{b}]$. Nun definieren wir

$$\Psi_{\mathfrak{A},\mathfrak{B}} := \{\psi_{\overline{a}} : \overline{a} \in A \text{ und } \mathfrak{A} \models \varphi[\overline{a}]\} \cup \{\psi_{\overline{b}} : \overline{b} \in B \text{ und } \mathfrak{B} \models \varphi[\overline{b}]\}.$$

Nach Konstruktion ist $\Psi_{\mathfrak{A},\mathfrak{B}}$ eine endliche Teilmenge von Ψ. Weiterhin gilt für alle $\overline{a} \in A, \overline{b} \in B$:

$$\mathfrak{A} \models \bigvee \Psi_{\mathfrak{A},\mathfrak{B}} \text{ gdw. } \mathfrak{A} \models \bigvee \Psi[\overline{a}] \text{ gdw. } \mathfrak{A} \models \varphi[\overline{a}]$$

sowie

$$\mathfrak{B} \models \bigvee \Psi_{\mathfrak{A},\mathfrak{B}} \text{ gdw. } \mathfrak{B} \models \bigvee \Psi[\overline{b}] \text{ gdw. } \mathfrak{B} \models \varphi[\overline{b}].$$

Nach der 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 $\overline{a} \in A$ und $\overline{b} \in B$: $\mathfrak{A} \models \varphi[\overline{a}]$ gdw. $\mathfrak{A} \models \tilde{\varphi}[\overline{a}]$ sowie $\mathfrak{B} \models \varphi[\overline{b}]$ gdw. $\mathfrak{B} \models \tilde{\varphi}[\overline{b}]$. □

3.4.3 Pebble-Spiele

Ziel dieses Abschnitts ist es, Ehrenfeucht-Fraïssé-Spiele für die Logiken FO^k und $L^k_{\infty\omega}$ einzuführen. Dazu zunächst ein wenig Notation.

3.51 Notation. Sei σ eine 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 $\overline{a} := a_1 \dots a_k \in (A \dot{\cup} \{*\})^k$ definieren wir den *Träger* Tr($\overline{a}$) *von* $\overline{a}$ als Tr($\overline{a}$) $:= \{i : a_i \neq *\}$.
- Für $i \in \{1, \dots, k\}$, $a \in A$ und $\overline{a} := a_1 \dots a_k \in (A \cup \{*\})^k$ setzen wir $\overline{a}\frac{a}{i} := a_1 \dots, a_{i-1} a a_{i+1} \dots a_k$. Das heißt, wir ersetzen die i-te Stelle von $\overline{a}$ durch a.

3.52 Definition. Sei σ eine Signatur, $k \in \mathbb{N}_{\geqslant 1}$ und $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen. Für $\overline{a} \in (A \cup \{*\})^k$ und $\overline{b} \in (B \cup \{*\})^k$ ist die Abbildung $\overline{a} \mapsto \overline{b}$ ein *k-partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$*, falls

(1) Tr($\overline{a}$) = Tr($\overline{b}$) und

(2) $(\overline{a} \mapsto \overline{b})_{|\text{Tr}(\overline{a})}$ ist ein k-partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

Mit $\text{Part}^k(\mathfrak{A}, \mathfrak{B})$ bezeichnen wir 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.

3.53 Definition (Pebble-Spiele). Sei σ eine Signatur, $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen. Sei $k \in \mathbb{N}_{\geqslant 1}$, $\overline{a} \in (A \dot\cup \{*\})^k$, $\overline{b} \in (B \dot\cup \{*\})^k$ mit $\text{Tr}(\overline{a}) = \text{Tr}(\overline{b})$.

Das k-Pebble-Spiel $G^k_\infty(\mathfrak{A}, \overline{a}, \mathfrak{B}, \overline{b})$ wird zwischen zwei Spielern, Spoiler und Duplicator, gespielt. Den Spielern stehen insgesamt $2 \cdot k$ Spielsteine $\alpha_1, \dots, \alpha_k, \beta_1, \dots, \beta_k$ zur Verfügung, die auf Elemente der Strukturen gelegt werden können. Zu Beginn des Spiels liegt für jedes $i \in \text{Tr}(\overline{a})$ der Stein α_i auf a_i und β_i auf b_i. Die übrigen Steine liegen nebem dem „Spielbrett". In jedem Zug wählt Spoiler eine der Strukturen $\mathfrak{A}$ oder $\mathfrak{B}$, in der er ziehen will. Bei Wahl von $\mathfrak{A}$ nimmt er danach einen der Spielsteine α_i und plaziert ihn auf einem Element $a_i \in A$. Bei Wahl von $\mathfrak{B}$ zieht er entsprechend mit einem der Steine $\beta_1, \dots, \beta_k$ in B. Duplicator antwortet danach mit einem Stein in der anderen Struktur, d.h. zieht Spoiler den Stein α_i in $\mathfrak{A}$ so antwortet Duplicator mit dem Stein β_i und legt ihn auf ein Element $b_i \in B$. Zieht Spoiler in $\mathfrak{B}$, so muß Duplicator in $\mathfrak{A}$ antworten. Beide Spieler dürfen dabei mehrere Steine auf dasselbe Element legen oder Steine vom Spielbrett entfernen bzw. wieder hereinnehmen.

Insgesamt werden in dem Spiel unendliche viele Züge gespielt. Seien nach einem Zug i $a_1, \dots, a_k$ die Elemente (oder $*$) auf denen die Spielsteine $\alpha_1, \dots, \alpha_k$ liegen und entsprechend $b_1, \dots, b_k$ die gewählten Elemente in B. Ist die Abbildung $\overline{a} \mapsto \overline{b}$ kein k-partieller Isomorphismus, so endet das Spiel nach dem Zug i und Spoiler gewinnt. Andernfalls wird das Spiel fortgesetzt. Duplicator gewinnt, wenn unendlich lange gespielt wird, also nach jedem Zug die Abbildung $\overline{a} \mapsto \overline{b}$ ein k-partieller Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$ ist.

Bemerkung:

- Gilt $\overline{a} = \overline{b} = * \cdots *$ so schreiben wir $G^k_\infty(\mathfrak{A}, \mathfrak{B})$ anstelle von $G^k_\infty(\mathfrak{A}, * \cdots *, \mathfrak{B}, * \cdots *)$.
- Liegen nach einem Zug die Spielsteine $\alpha_1, \dots, \alpha_k$ auf $a_1, \dots, a_k$ und β_i auf b_i so schreiben wir auch $\bar{\alpha} \mapsto \bar{\beta}$ anstelle von $\overline{a} \mapsto \overline{b}$.
- Strategien und Gewinnstrategien im k-pebble-Spiel sind analog zum 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. In diesem Falle steht also schon nach einer endlichen Zahl von Zügen fest, wer das Spiel gewinnen kann.

3.54 Beispiel. (a) Sei $\sigma := \varnothing$ und $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen mit $|A|, |B| \geqslant k$. Dann hat Duplicator eine Gewinnstrategie in $G^k_\infty(\mathfrak{A}, \mathfrak{B})$, in dem 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) Seien jetzt $\mathfrak{A}, \mathfrak{B}$ endliche, linear geordnete $\{<\}$-Strukturen. Dann hat Duplicator genau dann eine Gewinnstrategie im Spiel $G^2_\infty(\mathfrak{A}, \mathfrak{B})$, wenn $|A| = |B|$.

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 Spoiler's Wahl isomorphe Elemente wählt.

Zum Beweis der Hinrichtung nehmen wir an, daß $|A| \neq |B|$ und zeigen, daß dann Spoiler eine Gewinnstratgie in $G^2_\infty(\mathfrak{A}, \mathfrak{B})$ hat. O.B.d.A. sei $|A| > |B|$. Spoiler's Strategie besteht darin, im Zug $i \geqslant 1$ den Stein $\alpha_{1+(i \mod 2)}$ auf das Element mit Rang $i-1$ in $\mathfrak{A}$ zu legen.

In den ersten beiden Zügen legt Spoiler also seine beiden Spielsteine α_1, α_2 auf die beiden kleinsten Elemente in $\mathfrak{A}$. In den folgenden Zügen nimmt er jeweils den Stein auf dem kleineren Element und plaziert ihn auf das kleinste noch nicht im Spiel verwendete Element. Nach jedem Zug $i \geqslant 2$ liegen also die Steine α_1, α_2 auf den Elemeten mit Rang $i-1$ und i. 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. Ansonsten wäre die Abbildung $\alpha_1, \alpha_2 \mapsto \beta_1, \beta_2$ kein k-partieller Isomorphismus. Da aber $|B| < |A|$ kann Duplicator dies nach spätestens $|B|$ Zügen 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 den Begriff des m-Isomorphismus und geeigneter Hin-und-Her-Systeme.

3.55 Definition. Sei σ eine 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 $\overline{a} \mapsto \overline{b} \in I$, alle $a \in A$ und $i \in \{1, \dots, k\}$ gibt es ein $b \in B$, so daß $\overline{a}\frac{a}{i} \mapsto \overline{b}\frac{b}{i} \in I$.

k-Her-Eigenschaft: Für alle $\overline{a} \mapsto \overline{b} \in I$, alle $b \in B$ und $i \in \{1, \dots, k\}$ gibt es ein $a \in A$, so daß $\overline{a}\frac{a}{i} \mapsto \overline{b}\frac{b}{i} \in I$.

Ein System mit diesen Eigenschaften nennt man *Hin-und-Her-System*. Wir schreiben $I : \mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$ um anzudeuten, daß I ein Hin-und-Her-System zwischen $\mathfrak{A}$ und $\mathfrak{B}$ ist.

Bemerkung: Die Menge $W^k_\infty(\mathfrak{A}, \mathfrak{B}) := \{\overline{a} \mapsto \overline{b} \in \text{Part}^k(\mathfrak{A}, \mathfrak{B}) :$ Duplicator hat eine Gewinnstrategie in $G^k_\infty(\mathfrak{A}, \overline{a}, \mathfrak{B}, \overline{b})\}$ hat die k-Hin-und-Her-Eigenschaft, ist aber möglicherweise leer.

3.56 Theorem. *Sei σ eine Signatur, $k \in \mathbb{N}_{\geqslant 1}$, $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen, $\overline{a} \in (A \cup \{*\})^k, \overline{b} \in (B \cup \{*\})^k$ mit $\mathrm{Tr}(\overline{a}) = \mathrm{Tr}(\overline{b})$. Dann sind folgende Aussagen äquivalent.*

(i) Duplicator hat eine Gewinnstrategie in $G^k_\infty(\mathfrak{A}, \overline{a}, \mathfrak{B}, \overline{b})$.

(ii) $\overline{a} \mapsto \overline{b} \in W^k_\infty(\mathfrak{A}, \mathfrak{B})$ und $W^k_\infty : \mathfrak{A} \cong^k_{part} \mathfrak{B}$.

(iii) Es gibt ein $I : \mathfrak{A} \cong^k_{part} \mathfrak{B}$ mit $\overline{a} \mapsto \overline{b} \in I$.

(iv) $\overline{a}$ in $\mathfrak{A}$ und $\overline{b}$ in $\mathfrak{B}$ erfüllen dieselben $L^k_{\infty\omega}[\sigma]$-Formeln.

Beweis: Die Implikation $(i) \Rightarrow (ii)$ folgt sofort aus der Definition der Hin-und-Her-Systeme. Die Implikation $(ii) \Rightarrow (iii)$ ist trivial und die Implikation $(iii) \Rightarrow (iv)$ folgt per Induktion über den Aufbau der $L^\omega_{\infty\omega}$-Formeln.

Es bleibt also die Richtung $(iv) \Rightarrow (i)$ zu zeigen. Wir zeigen, daß Duplicator eine Strategie hat, so daß nach jeder Runde die folgende Invariante erhalten bleibt:

$$\begin{array}{c}\text{Sind } \overline{c} \text{ und } \overline{d} \text{ die in } \mathfrak{A} \text{ bzw. } \mathfrak{B} \text{ belegten Elemente,}\\ \text{so erfüllt } \overline{c} \text{ dieselben } L^k_{\infty\omega}[\sigma]\text{-Formeln in } \mathfrak{A} \text{ wie } \overline{d} \text{ in } \mathfrak{B}.\end{array} \tag{*}$$

Offensichtlich gilt $\overline{c} \mapsto \overline{d} \in \mathrm{Part}^k(\mathfrak{A}, \mathfrak{B})$ nach jedem Zug bei dem die Invariante erhalten bleibt.

Nach Voraussetzung des Falls (iv) ist die Invariante zu Beginn des Spiels erfüllt. Gelte nun $(*)$ nach Zug $i \in \mathbb{N}$. Angenommen Spoiler legt im Zug $i+1$ den Spielstein α_i auf ein Element $c \in A$. Sei $M := \{\psi \in L^k_{\infty\omega} : \mathfrak{A}, \overline{c}\frac{c}{i} \models \psi\}$. Nach Konstruktion gilt also $\mathfrak{A}, \overline{c} \models \exists x_i \bigwedge M$ und daher, wegen $(*)$, $\mathfrak{B}, \overline{d} \models \exists x_i \bigwedge M$. Also existiert ein $d \in B$, so daß für alle $\psi \in M$ gilt: $\mathfrak{B}, \overline{d}\frac{d}{i} \models \psi$. Duplicator wählt nun dieses b als Antwortzug. Dann gilt für jede Formel $\chi \in L^k_{\infty\omega}$ entweder $\chi \in M$ oder $\neg\chi \in M$ und daher $\mathfrak{A}, \overline{c}\frac{c}{i} \models \chi$ genau dann, wenn $\mathfrak{B}, \overline{d}\frac{d}{i} \models \chi$. □

Folgendes Korrolar folgt sofort aus dem Theorem zusammen mit Theorem 3.50.

3.57 Korollar. *Sei σ eine Signatur, $k \in \mathbb{N}_{\geqslant 1}$ und $\mathfrak{A}$, $\mathfrak{B}$ σ-Strukturen. Dann sind folgende Aussagen äquivalent:*

(i) Duplicator hat eine Gewinnstrategie in $G^k_\infty(\mathfrak{A}, \mathfrak{B})$.

(ii) $W^k_\infty(\mathfrak{A}, \mathfrak{B}) : \mathfrak{B} \cong^k_{part} \mathfrak{B}$.

(iii) $\mathfrak{A} \cong^k_{part} \mathfrak{B}$.

(iv) $\mathfrak{A} \equiv^{L^k_{\infty\omega}} \mathfrak{B}$.

Sind $\mathfrak{A}$ und $\mathfrak{B}$ endlich, ist folgende Aussage auch noch äquivalent zu den vorherigen:

(v) $\mathfrak{A} \equiv^{\mathrm{FO}^k} \mathfrak{B}$.

3.58 Korollar. *Sei σ eine Signatur, $\mathcal{K}$ eine Klasse von σ-strukturen, $\mathcal{C} \subseteq \mathcal{K}$. Falls die unten stehende Bedingung $(*)$ erfüllt ist, so ist $\mathcal{C}$* nicht *$L^{\omega}_{\infty\omega}[\sigma]$-definierbar in $\mathcal{K}$.*

$$\textit{Für jedes } k \in \mathbb{N}_{\geqslant 1} \textit{ gibt es } \mathfrak{A} \in \mathcal{C} \textit{ und } \mathfrak{B} \in \mathcal{K} \setminus \mathcal{C}, \qquad (*)$$
$$\textit{so daß Duplicator eine Gewinnstrategie in } G^k_{\infty}(\mathfrak{A}, \mathfrak{B}) \textit{ hat.}$$

Beweis: Es sei $(*)$ erfüllt. Angenommen, $\mathcal{C}$ sei $L^{\omega}_{\infty\omega}[\sigma]$-definierbar, etwa durch den Satz $\varphi \in L^k_{\infty\omega}[\sigma]$, für ein $k \in \mathbb{N}$. Es gilt also $\mathcal{C} = \mathrm{Mod}_{\mathcal{K}}(\varphi)$. Nach Voraussetzung $(*)$ gibt es Strukturen $\mathfrak{A} \in \mathcal{C}$ und $\mathfrak{B} \in \mathcal{K} \setminus \mathcal{C}$, so daß $\mathfrak{A} \equiv^{L^k_{\infty\omega}} \mathfrak{B}$. Da $\mathfrak{A} \in \mathcal{C} = \mathrm{Mod}_{\mathcal{K}}(\varphi)$ gilt $\mathfrak{A} \models \varphi$ und somit $\mathfrak{B} \models \varphi$. Dies ist aber ein Widerspruch zu $\mathfrak{B} \notin \mathcal{C}$. □

3.59 Beispiel. Die Klasse

$$\textsc{Even} := \{\mathfrak{A} : \mathfrak{A} \text{ endliche } \varnothing\text{-Struktur}, |A| \text{ gerade}\}$$

ist nicht $L^{\omega}_{\infty\omega}$ definierbar in der Klasse aller endlichen Strukturen.

Wie in Beispiel 3.54 gezeigt, hat Duplicator eine Gewinnstrategie im Spiel $G^k_{\infty}(\mathfrak{A}, \mathfrak{B})$, wenn $|A|, |B| \geqslant k$. Die Behauptung folgt jetzt sofort aus dem vorherigen Theorem.

Ein etwas komplexeres Beispiel liefert der folgende Satz.

3.60 Theorem (de Rougemont, 1987). *Die Klasse*

$$\textsc{Hamilton} := \{\mathcal{G} : \mathcal{G} \textit{ ist ein endlicher Graph, der einen Hamilton-Pfad enthält}\}$$

ist nicht $L^{w}_{\infty\omega}[\sigma]$-definierbar in der Klasse aller endlichen, zusammenhängenden Graphen.

Zur Erinnerung: Ein Hamilton-Pfad in einem Graph ist ein Pfad, der jeden Knoten genau einmal enthält.

Beweis: Nach Korrolar 3.58 reicht es, für jedes k endliche, zusammenhängende Graphen $\mathfrak{A}$ und $\mathfrak{B}$ zu finden, so daß $\mathfrak{A}$ aber nicht $\mathfrak{B}$ einen Hamilton-Pfad enthält und Duplicator eine Gewinnstrategie im Spiel $G^k_{\infty}(\mathfrak{A}, \mathfrak{B})$ hat.

Für $m, n \in \mathbb{N}_{\geqslant 1}$ definieren wir den Graph $H_{m,n} := (V_{m,n}, E_{m,n})$ mit Knotenmenge

$$V_{m,n} := \{v_1, \dots, v_m, w_1, \dots, w_n\}$$

und Kantenmenge

$$\begin{aligned} E_{m,n} := \quad & \{(w_i, w_j : j = (i+1) \mod n\} \cup \\ & \{(v_i, w_j), (w_j, v_i) : 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n\}. \end{aligned}$$

Der Graph $H_{m,n}$ besteht also aus einem gerichteten Kreis $w_1 \dots w_m w_1$ der Länge m sowie einer Menge $\{v_1, \dots, v_n\}$ von Knoten, die mit jedem w_i durch eine ungerichtete Kante verbunden sind, untereinander jedoch keine Kanten haben.

Behauptung 1: $H_{m,n}$ hat genau dann einen Hamilton-Pfad, wenn $n \geqslant m - 1$.

Beweis: Zum Beweis der Rückrichtung sei $n \geqslant m - 1$. Dann ist $p := v_1 w_1 v_2 w_2 \dots v_{m-1}$

$w_{m-1}\, v_m\, w_m \dots w_n$ ein Hamiltonpfad in $H_{m,n}$. Für die Hinrichtung sei $p = u_1 u_2 \dots u_{m+n}$ ein Hamiltonpfad in $H_{m,n}$. Sei $1 \leqslant i_1 < i_2 < \dots < i_m \leqslant n+m$, so daß $\{u_{i_1}, \dots, u_{i_m}\} = \{v_1, \dots, v_m\}$. Gemäß der Definition von $E_{m,n}$ gibt es keine Kanten zwischen Knoten aus $\{v_1, \dots, v_m\}$. Somit muß für alle $j \in \{1, \dots, m-1\}$ gelten: Zwischen u_{i_j} und $u_{i_{j+1}}$ liegt mindestens ein Knoten $u_{i_j+1} \in \{w_1, \dots, w_n\}$. Insbesondere $n := |\{w_1, \dots, w_n\}| \geqslant m-1$. $\square$

Aus Behauptung 1 folgt für $k \in \mathbb{N}_{\geqslant 1}$: $\mathfrak{A} := H_{k+1,k}$ hat einen Hamiltonpfad, $\mathfrak{B} := H_{k+2,k}$ aber nicht.

Behauptung 2: Duplicator hat eine Gewinnstrategie im Spiel $G^k_\infty(H_{k+1,k}, H_{k+2,k})$.

Beweis: Setze $\mathfrak{A} := H_{k+1,k}$, $\mathfrak{B} := H_{k+2,k}$. Seien $A := \{v_1, \dots, v_{k+1}, w_1, \dots, w_k\}$ und $B := \{v'_1, \dots, v'_{k+2}, w'_1, \dots, w'_k\}$ die Knotenmenge von $\mathfrak{A}$ und $\mathfrak{B}$. Gemäß Korollar 3.57 reicht es zu zeigen, daß $\mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$. Wähle $I \subseteq \text{Part}^k(\mathfrak{A}, \mathfrak{B})$ folgendermaßen:

$$
\begin{aligned}
I := \{\overline{a} \mapsto \overline{b} :\ & \overline{a} = a_1 \dots a_k \in (A \dot\cup \{*\})^k, \overline{b} = b_1 \dots b_k \in (B \dot\cup \{*\})^k \\
& \text{für alle } i \in \{1, \dots, k\} \text{ gilt:} \\
& 1)\ a_i = * \iff b_i = * \\
& 2)\ \text{falls } a_i = w_j \text{ für ein } j \in \{1, \dots, k\},\ \text{so } b_i = w'_j \\
& 3)\ a_i \in \{v_1, \dots, v_{k+1}\} \iff b_i \in \{v'_1, \dots, v'_{k+2}\} \\
& 4)\ \text{für alle } i' \in \{1, \dots, k\} \text{ gilt: } a_i = a'_i \iff b_i = b'_i\}
\end{aligned}
$$

Da $\bar{*} \mapsto \bar{*} \in I$ ist offensichtlich $I \neq \varnothing$. Weiterhin ist $I \subseteq \text{Part}^k(\mathfrak{A}, \mathfrak{B})$. Schließlich kann man leicht die Hin- und Her-Eigenschaft für das System I nachweisen. Somit ist also $I : \mathfrak{A} \cong^k_{\text{part}} \mathfrak{B}$. Nach Korollar 3.57 hat also Duplicator eine Gewinnstrategie in $G^k_\infty(\mathfrak{A}, \mathfrak{B})$. $\square$

Insgesamt haben wir also für alle $k \in \mathbb{N}_{\geqslant 1}$ Graphen $\mathfrak{A} := H_{k+1,k} \in$ HAMILTON und $\mathfrak{B} := H_{k+2,k} \in \mathcal{K} \setminus$HAMILTON gefunden, so daß Duplicator $G^k_\infty(\mathfrak{A}, \mathfrak{B})$ gewinnt. $\square$

4 Fixpunktlogiken

In Kapitel 2 haben wir verschiedene Erweiterungen der Logik erster Stufe um Fixpunktkonstrukte 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 das letzte Kapitel 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.

4.1 Fixpunktlogiken und $L^{\omega}_{\infty\omega}$

4.1 Notation. Für den Rest dieses Kapitels vereinbaren wir folgende Notation:

- Fin : Klasse aller endlichen Strukturen
- $\text{Fin}_<$: Klasse aller endlichen linear geordneten Strukturen
- Für Logiken $\mathcal{L}$ und $\mathcal{L}'$ und eine Klasse $\mathcal{K}$ von Strukturen schreiben wir „$\mathcal{L} \leqslant \mathcal{L}'$ *auf* $\mathcal{K}$“, falls es für jede Formel $\varphi \in \mathcal{L}$ eine Formel $\varphi' \in \mathcal{L}'$ gibt, die zu φ auf $\mathcal{K}$ äquivalent ist, d.h. für alle $\mathfrak{A} \in \mathcal{K}$ und $\overline{a} \in A^k$ gilt: $\mathfrak{A} \models \varphi[\overline{a}]$ genau dann, wenn $\mathfrak{A} \models \varphi'[\overline{a}]$.
- Entsprechend schreiben wir $\mathcal{L} = \mathcal{L}'$ auf $\mathcal{K}$, falls $\mathcal{L} \leqslant \mathcal{L}'$ und $\mathcal{L}' \leqslant \mathcal{L}$ und $\mathcal{L} < \mathcal{L}'$ auf $\mathcal{K}$, falls $\mathcal{L} \leqslant \mathcal{L}'$ und $\mathcal{L}' \nleqslant \mathcal{L}$.

4.2 Theorem. *Es gilt* LFP $\leqslant$ IFP $\leqslant$ PFP $< L^{\omega}_{\infty\omega}$ *auf* Fin.

Beweis: LFP $\leqslant$ IFP $\leqslant$ PFP auf Fin wurde schon in Kapitel 2 gezeigt.

Wir zeigen als nächstes, daß $L^{\omega}_{\infty\omega} \nleqslant$ PFP. Sei dazu $J \subseteq \mathbb{N}$ eine unentscheidbare Menge, z.B. die Gödel-Nummern aller Turing-Maschinen, die bei leerer Eingabe halten. Sie φ_J der in Beispiel 3.48 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. Das heißt, daß $\text{Mod}_{\mathcal{K}}(\varphi)$ unentscheidbar ist, wohingegen jede in PFP definierbare Klasse von Strukturen entscheidbar ist. Folglich ist φ_J nicht äquivalent zu einem Satz in PFP.

Es bleibt zu zeigen, daß jeder Satz in $\varphi \in$ PFP äquivalent zu einem Satz in $\varphi^* \in L^{\omega}_{\infty\omega}$ ist. Der Beweis folgt per Induktion über den Formelaufbau. Der einzig interessante Fall ist, wenn φ die Form $\varphi := [\mathbf{pfp}_{R,\overline{x}}\,\psi](\overline{t})$ hat. Dabei sei $|\overline{x}| = ar(R) = |\overline{t}| = r$. Nach Induktionsannahme ist ψ äquivalent zu einer $L^\omega_{\infty\omega}$-Formel $\hat{\psi}$. Sei also $\hat{\psi} \in L^k_{\infty\omega}$ für ein $k \in \mathbb{N}$. Seien $\overline{y} := y_1 \dots y_r$ neue, d.h. in $\hat{\psi}$ nicht vorkommende, Variablen. Wir definieren Formeln $\hat{\psi}^\alpha$, für $\alpha \in \mathbb{N}$, induktiv wie folgt: Sei $\hat{\psi}^0(\overline{x}) := \neg x_1 = x_1$. Die Formel $\hat{\psi}^{\alpha+1}$ entsteht aus $\hat{\psi}^\alpha$ indem jedes Vorkommen eines Atoms $R\overline{u}$, für ein Tupel $\overline{u}$ von Termen, durch die Formel $\exists \overline{y}\big(\bigwedge_{i=1}^r y_i = u_i \wedge \exists \overline{x} \bigwedge_{i=1}^r x_i = y_i \wedge \hat{\psi}^\alpha(\overline{x})\big)$ ersetzt wird. Hierbei ist die Verwendung der neuen Variablen $\overline{y}$ nötig, da die Variablen x_i als Terme in $\overline{u}$ vorkommen können.

Mittels Induktion zeigt man leicht, daß für jede Struktur $\mathfrak{A}$ und alle $\overline{a} \in A^r$ gilt: $\overline{a} \in R^\alpha$ genau dann, wenn $\mathfrak{A} \models \hat{\psi}^\alpha[\overline{a}]$, wobei R^α die α-te Stufe der Fixpunktinduktion über ψ bezeichnet. Somit ist φ äquivalent zur $L^\omega_{\infty\omega}$-Formel

$$\hat{\varphi} := \bigvee_{\alpha \in \mathbb{N}} \big[(\forall \overline{x}\,(\hat{\psi}^\alpha(\overline{x}) \leftrightarrow \hat{\psi}^{\alpha+1}(\overline{x})) \wedge \hat{\psi}^\alpha(\overline{t})\big).$$

Hierbei steht $\hat{\psi}^\alpha(\overline{t})$ wiederum für $\exists \overline{y}\big(\bigwedge_{i=1}^r y_i = t_i \wedge \exists \overline{x} \bigwedge_{i=1}^r x_i = y_i \wedge \hat{\psi}^\alpha(\overline{x})\big)$ $\square$

Aus dem Theorem folgt sofort, daß Klassen wie EVEN oder HAMILTON, für die wir in Kapitel 3 schon gezeigt hatten, daß sie nicht in $L^\omega_{\infty\omega}$-definerbar sind, auch nicht in PFP definiert werden können. Da man selbstverständlich die Klasse EVEN in Polynomialzeit entscheiden kann, folgt daraus sofort das nächste Korollar.

4.3 Korollar. *Es gilt* PFP $<$ PSPACE *auf* Fin *und* LFP $\leqslant$ IFP $<$ PTIME *auf* Fin. *Weiterhin gilt* PTIME $\not\leqslant$ PFP. *Allerdings gilt auch* PFP $\not\leqslant$ PTIME, *es sei denn* PTIME $=$ PSPACE. PTIME *und* PFP *liegen also schief zueinander.*

4.2 Simultane Fixpunkte

4.4 Definition. Sei M eine Menge und sei $k \in \mathbb{N}$. Ferner seien Abbildungen $F_i : \mathrm{Pot}(M^{r_1}) \times \dots \times \mathrm{Pot}(M^{r_k}) \longrightarrow \mathrm{Pot}(M^{r_i})$ gegeben, für natürliche Zahlen $r_1, \dots, r_k \in \mathbb{N}$.

Die Abbildung $\bar{F} := \bar{F}_{(F_1,\dots,F_k)}$ ist definiert als

$$\begin{array}{rccc} \bar{F}: & \mathrm{Pot}(M^{r_1}) \times \dots \times \mathrm{Pot}(M^{r_k}) & \longrightarrow & \mathrm{Pot}(M^{r_1}) \times \dots \times \mathrm{Pot}(M^{r_k}) \\ & (R_1, \dots, R_k) & \mapsto & (F_1(\overline{R}), \dots, F_k(\overline{R})) \end{array}$$

Abbildung wie in der vorherigen Definition werden *simultane Abbildungen über* $F_1, \dots, F_k$ genannt. Eine simultane Abbildung $\bar{F}$ heißt monoton, wenn für alle $\overline{R}$ und $\overline{Q}$ gilt: Ist $\overline{R} \subseteq \overline{Q}$ so auch $\bar{F}(\overline{R}) \subseteq \bar{F}(\overline{Q})$. Hierbei gilt $\overline{R} \subseteq \overline{Q}$, falls $R_i \subseteq Q_i$ für alle i. Analog ist der Begriff einer inflationären simultanen Abbildung definiert.

4.5 Lemma. *Sei* $m \geqslant 1$ *und* $r_1, \dots, r_m \geqslant 0$. *Seien ferner* A *eine Menge und für alle* $1 \leqslant i \leqslant j$,

$$F_i : A^{r_1} \times \dots \times A^{r_m} \longrightarrow A^{r_i}$$

Abbildungen. Sind alle F_i komponentenweise monoton, d.h. gilt für alle $1 \leqslant i \leqslant m$ und alle $R_1 \subseteq A^{r_1}, \dots, R_m \subseteq A^{r_m}$ sowie $R'_j \supseteq R_j$,

$$F_i(R_1, \dots, R_m) \subseteq F_i(R_1, \dots, R_{j-1}, R'_j, R_{j+1}, \dots, R_m),$$

so ist $\bar{F} := (F_1, \dots, F_m)$ monoton.

Beweis: Übung. □

Wir können nun völlig analog zu dem Vorgehen in Kapitel 2 Fixpunkte von simultanen Abbildungen bilden.

Sind wie zuvor $F_1, \dots, F_k$ Abbildungen

$$F_i : \mathrm{Pot}(M^{r_1}) \times \cdots \times \mathrm{Pot}(M^{r_k}) \to \mathrm{Pot}(M^{r_i})$$

gegeben, so definieren wir induktiv eine Sequenz $(R_1^\alpha, \dots, R_k^\alpha)_{\alpha\in\mathbb{N}}$ von Mengen durch $R_i^0 := \varnothing$ und $R_i^{\alpha+1} := F_i(R^\alpha)$. Ist die Menge M endlich, so existiert ein $\alpha \in \mathbb{N}$ mit $\bar{F}^\alpha := \bar{F}^{\alpha+1} =: \bar{F}^\infty$. Analog zum Satz von Knaster und Tarski zeigt man leicht, daß für montone Abbildungen $\bar{F}$ der kleinste Fixpunkt $\mathbf{s\text{-}lfp}(\bar{F})$ immer existiert und es gilt $\mathbf{s\text{-}lfp}(\bar{F}) = F^\infty$.

Wir bezeichnen mit $\mathbf{s\text{-}lfp}(\bar{F})_j$ die j-te Komponente R_j^∞ des kleinsten Fixpunkts von $\bar{F}$.

4.6 Definition. Sei $m \geqslant 1$, σ eine Signatur und $R_1, \dots, R_m$ Relationssymbole der Stelligkeit $r_1, \dots, r_m$ mit $R_i \notin \sigma$, für $1 \leqslant i \leqslant m$. Seien weiterhin $\varphi_j(\overline{x}_j, R_1, \dots, R_m)$ Formeln mit $1 \leqslant j \leqslant m$ und $|\overline{x}_j| = r_j$ gegeben. Auf jeder (endlichen) σ-Struktur $\mathfrak{A}$ definieren die φ_j eine Abbildung

$$\begin{aligned} F_{\varphi_j} :\ & \mathrm{Pot}(A^{r_1}) \times \cdots \times \mathrm{Pot}(A^{r_m}) \to \mathrm{Pot}(A^{r_j}) \\ & (R_1, \dots, R_m) \mapsto \{\overline{a} \in A^{r_i} : (\mathfrak{A}, R_1, \dots, R_m) \models \varphi_j[\overline{a}]\}. \end{aligned}$$

Somit definiert $(\varphi_1, \dots, \varphi_m)$ die Abbildung $\bar{F}_{(\varphi_1,\dots,\varphi_m)} := (F_{\varphi_1}, \dots, F_{\varphi_m})$.

Sind alle φ_j positiv in den Variablen $R_1, \dots, R_m$, so ist $\bar{F}_{\bar{\varphi}}$ monoton (nach Lemma 4.5) und somit existiert der simultane Fixpunkt $\mathbf{s\text{-}lfp}(\bar{F})$, den wir oft als $\mathbf{s\text{-}lfp}(\varphi_1, \dots, \varphi_m)$ schreiben.

4.7 Beispiel. In folgendem 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_1xy und R_2xy zu führen, so daß in R_1 alle Paare (a, b) vorkommen, zwischen denen ein Pfad ungerader Länge und entsprechend in R_2 alle Paare (a, b) vorkommen, zwischen denen ein Pfad gerader Länge existiert. Sei dazu

$$\varphi_1(R_1, R_2, xy) := Exy \lor \exists z(Exz \land R_2zy)$$

und

$$\varphi_2(R_1, R_2, xy) := x = y \lor \exists z(Exz \land R_1zy).$$

In einem Graph $\mathcal{G} := (V, E)$ definiert $\bar{\varphi} = (\varphi_1, \varphi_2)$ eine Abbildung $\bar{F} := (F_{\varphi_1}, F_{\varphi_2})$ mit $F_{\varphi_j}(R_1, R_2) := \{(u, v) \in V^2 : (\mathcal{G}, R_1, R_2) \models \varphi_j[u, v]\}$. Die ersten Induktionsstufen der dadurch induzierten Fixpunktinduktion lauten

$$
\begin{array}{ll}
R_1^0 := \varnothing & R_2^0 := \varnothing \\
R_1^1 := \{(u,v) : \begin{array}{l}\text{es gibt Pfad der Länge 1} \\ \text{von } u \text{ nach } v\end{array}\} & R_2^1 := \{(u,u) : u \in V\} \\
R_1^2 := R_1^1 & R_2^2 := \{(u,v) : \begin{array}{l}\text{es gibt Pfad der Länge} \\ \text{0 oder 2 von } u \text{ nach } v\end{array}\} \\
R_1^3 := \{(u,v) : \begin{array}{l}\text{es gibt Pfad der Länge} \\ \text{1 oder 3 von } u \text{ nach } v\end{array}\} & R_2^3 := R_2^2 \\
\vdots & \vdots
\end{array}
$$

Insgesamt gilt also

$$\textbf{s-lfp}(\bar{F})_1 := \{(u, v) : \text{es gibt einen Pfad ungerader Länge von } u \text{ nach } v\}$$

und

$$\textbf{s-lfp}(\bar{F})_2 := \{(u, v) : \text{es gibt einen Pfad gerader Länge von } u \text{ nach } v\}.$$

4.8 Notation. Für Formeln $\varphi_1(\overline{x}_1), \ldots, \varphi_m(\overline{x}_m)$ und Relationsvariablen $R_1, \ldots, R_m$ schreiben wir

$$S := \begin{cases} R_1\overline{x}_1 & \leftarrow \varphi_1(\overline{x}_1, R_1, \ldots, R_m) \\ & \vdots \\ R_m\overline{x}_m & \leftarrow \varphi_m(\overline{x}_m, R_1, \ldots, R_m) \end{cases}$$

und $\textbf{s-lfp}(S)$ für $\textbf{s-lfp}(\varphi_1, \ldots, \varphi_m)$.

4.9 Definition. Sei σ eine Signatur. Die *simultane kleinste Fixpunktlogik (*S-LFP*)* ist induktiv definiert durch die Regeln für die Logik erster Stufe sowie der Regel

(**S-LFP**): Ist $m \geqslant 1$ und sind $R_1, \ldots, R_m$ Relationsvariablen der Stelligkeiten $r_1, \ldots, r_m$ sowie $R_i \notin \sigma$ und sind weiterhin $\varphi_1, \ldots, \varphi_k \in$ S-LFP Formeln, positiv in allen Variablen $R_1, \ldots, R_m$, und es gilt für alle $1 \leqslant i \leqslant m$, $\{x_{i,1}, \ldots, x_{i,r_i}\} \subseteq \mathit{frei}(\varphi_i)$, für Variablen $\overline{x}_i := x_{i,1}, \ldots, x_{i,r_i}$, so ist

$$[\textbf{lfp } R_i : S](\overline{t}) \in \text{S-LFP},$$

wobei $\overline{t}$ ein r_i-Tupel von Termen und

$$S := \begin{cases} R_1\overline{x}_1 & \leftarrow \varphi_1(\overline{x}_1, R_1, \ldots, R_k) \\ & \vdots \\ R_m\overline{x}_m & \leftarrow \varphi_m(\overline{x}_m, R_1, \ldots, R_k) \end{cases}$$

das entsprechende System von Formeln ist.

Die Semantik ist dabei analog zur Definition für die Logik erster Stufe definiert, wobei für eine Formel $\psi(\overline{x}) := [\mathbf{lfp}\ R_i : S](\overline{x})$, eine σ-Struktur $\mathfrak{A}$ und alle $\overline{a} \in A^{r_i}$ gilt:

$$\mathfrak{A} \models \psi[\overline{a}] \qquad \text{genau dann, wenn} \qquad \overline{a} \in \textbf{s-lfp}(S)_i.$$

4.10 Beispiel. Seien φ_1, φ_2 die Formeln aus Beispiel 4.7. Dann gilt in jedem Graph $\mathcal{G} := (V, E)$ und für alle $u, v \in V$, $(\mathcal{G}, u, v) \models [\mathbf{lfp}\ R_1 : S](x, y)$, mit

$$S := \begin{cases} R_1xy & \leftarrow \varphi_1(x, y, R_1, R_2) \\ R_2xy & \leftarrow \varphi_2(x, y, R_1, R_2) \end{cases}$$

genau dann, wenn es einen Pfad ungerader Länge von u nach v in $\mathcal{G}$ gibt.

Das Verwenden simultaner Fixpunkt erlaubt es oft, Formeln sehr viel übersichtlicher und modularer zu schreiben, als es die einfache kleinste Fixpunktlogik erlaubt. Es stellt sich hierbei natürlich die Frage, ob simultane Fixpunkte die Ausdrucksstärke der resultierenden Logik erhöht. Wie wir als nächstes zeigen werden, ist dies nicht der Fall. Dazu zeigen wir zunächst ein allgemeines Lemma, welches in der Literatur bisweilen auch als das *Bekić-Prinzip* bezeichnet wird.

4.11 Lemma (Auflösen simultaner kleinster Fixpunkte). *Seien M_1, M_2 endliche Mengen und $F_1 : \mathrm{Pot}(M_1) \times \mathrm{Pot}(M_2) \to \mathrm{Pot}(M_1)$ sowie $F_2 : \mathrm{Pot}(M_1) \times \mathrm{Pot}(M_2) \to \mathrm{Pot}(M_2)$ monotone Abbildungen.*

Für jede Menge $X_1 \subseteq M_1$ definieren wir eine Abbildung $F_2^{X_1}$ durch

$$\begin{array}{rccl} F_2^{X_1} & : & \mathrm{Pot}(M_2) & \to & \mathrm{Pot}(M_2) \\ & & X_2 & \mapsto & F_2(X_1, X_2). \end{array}$$

$F_2^{X_1}$ entsteht also aus F_2, indem die erste Komponente X_1 fixiert wird. Sei ferner

$$\begin{array}{rccl} G_1 & : & \mathrm{Pot}(M_1) & \to & \mathrm{Pot}(M_1) \\ & & X_1 & \mapsto & F_1(X_1, \mathbf{lfp}(F_2^{X_1})). \end{array}$$

Dann gilt $\textbf{s-lfp}(F_1, F_2)_1 = \mathbf{lfp}(G_1)$.

Beweis: Sei $(R_1^\infty, R_2^\infty) := \textbf{s-lfp}(F_1, F_2)$ und seien für alle i, (R_1^i, R_2^i) die i-te Stufe der simultanen Fixpunktinduktion für (F_1, F_2).

$\subseteq$: Sei $S_1^\infty = \mathbf{lfp}(G_1)$. Per Induktion nach i zeigen wir, daß für alle $i \in \mathbb{N}$ gilt: $R_1^i \subseteq S_1^\infty$ und $R_2^i \subseteq \mathbf{lfp}(F_2^{S_1^\infty})$. Daraus folgt dann $R_1^\infty \subseteq S_1^\infty$. Die Behauptung ist klar für $i = 0$. Für $i + 1$ gilt

$$\begin{array}{rcll} R_1^{i+1} & = & F_1(R_1^i, R_2^i) & \\ & \subseteq & F_1\big(S_1^\infty, \mathbf{lfp}(F_2^{S_1^\infty})\big) & \text{nach Ind. Vor. und Monotonie der Abbildungen} \\ & = & G_1(S_1^\infty) = S_1^\infty & \text{da } S_1^\infty \text{ der kleinste Fixpunkt von } G_1 \text{ ist} \end{array}$$

und

$$\begin{aligned} R_2^{i+1} &= F_2(R_1^i, R_2^i) \\ &\subseteq F_2\big(S_1^\infty, \mathbf{lfp}(F_2^{S_1^\infty})\big) && \text{nach Ind. Vor. und Monotonie der Abbildungen} \\ &= F_2^{S_1^\infty}(\mathbf{lfp}(F_2^{S_1^\infty})) && \text{nach Definition von } F_2^{S_1^\infty} \\ &= \mathbf{lfp}(F_2^{S_1^\infty}). \end{aligned}$$

$\supseteq$: Aus der Definition von $F_2^{R_1^\infty}$ folgt, daß R_2^∞ ein Fixpunkt von $F_2^{R_1^\infty}$ ist. Somit ist $\mathbf{lfp}(F_2^{R_1^\infty} \subseteq R_2^\infty$ und $G_1(R_1^\infty) = F_1(R_1^\infty, \mathbf{lfp}(F_2^{R_1^\infty})) \subseteq F_1(R_1^\infty, R_2^\infty) = R_1^\infty$. Also gilt $G_1(R_1^\infty) \subseteq R_1^\infty$ und somit $\mathbf{lfp}(G_1) \subseteq R_1^\infty$.

□

Mit Hilfe des Lemmas kann nun leicht folgender Satz bewiesen werden.

4.12 Theorem. 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 einzig interessante Fall sind dabei Formeln $\psi := [\mathbf{lfp}\ R_1 : S](\overline{x})$, wobei nach Induktionsannahme vorausgesetzt werden kann, daß

$$S := \begin{cases} R_1\overline{x}_1 & \leftarrow \varphi_1 \\ & \vdots \\ R_k\overline{x}_k & \leftarrow \varphi_k \end{cases}$$

ein System von LFP-Formeln ist.

Wir zeigen hier für den Fall von $k = 2$, daß ψ äquivalent (auf Fin) zu einer Formel $\Phi(\overline{x}_1) \in$ LFP ist. Setze dazu

$$\Phi(\overline{x}_1) := [\mathbf{lfp}_{R_1\overline{x}_1}\, \varphi_1\big(\overline{x}_1, R_1, R_2\overline{u}/[\mathbf{lfp}_{R_2,\overline{x}_2}\, \varphi_2](\overline{u})\big)](\overline{x}_1),$$

wobei $\varphi_1\big(\overline{x}_1, R_1, R_2\overline{u}/[\mathbf{lfp}_{R_2,\overline{x}_2}\, \varphi_2](\overline{u})\big)$ aus φ_1 entsteht, indem jedes Atom der Form $R_2\overline{u}$ durch $[\mathbf{lfp}_{R_2,\overline{x}_2}\, \varphi_2](\overline{u})$ ersetzt wird.

Wir bezeichnen im folgenden $\varphi_1\big(\overline{x}_1, R-1, R_2\overline{u}/[\mathbf{lfp}_{R_2,\overline{x}_2}\, \varphi_2](\overline{u})\big)$ mit φ^*.

Sei $\mathfrak{A}$ eine σ-Struktur und seien für $i = 1, 2$,

$$F_i : \mathrm{Pot}(A^{r_1}) \times \mathrm{Pot}(A^{r_2}) \to \mathrm{Pot}(A^{r_i})$$

die durch φ_1, φ_2 und G_1 die durch φ^* gegebene Abbildung. Offenbar gilt

$$G_1(R) = F_1(R, \mathbf{lfp}(F_2^R)),$$

wobei F_2^R wie in Lemma 4.11 definiert ist. Aus Lemma 4.11 folgt $\mathbf{s\text{-}lfp}(F_1, F_2) = \mathbf{lfp}(G_1)$ und somit $\Phi(\overline{x}_1) \equiv [\mathbf{lfp}\ R_1 : S](\overline{x}_1)$. □

Bemerkung. Durch die in dem Satz skizzierte Übersetzung von S-LFP-Formeln wird die Stelligkeit der Relationsvariablen nicht verändert. Also sind auch monadisches S-LFP und LFP äquivalent.

Analog zu S-LFP kann man auch *simultanes* IFP *(*S-IFP*)* und *simultanes* PFP *(*S-PFP*)* definieren. Auch hier gilt S-IFP = IFP und S-PFP = PFP, allerdings muß ein anderer Beweis geführt werden.

4.13 Theorem. S-IFP $\equiv$ IFP.

Beweis: Der Beweis erfolgt wiederum über den Formelaufbau. Der einzig interessante Fall sind Formeln $\psi(\overline{x}) := [\mathbf{ifp}\ R_i : S](\overline{x})$, wobei

$$S := \begin{cases} R_1\overline{x}_1 & \leftarrow \varphi_1 \\ & \vdots \\ R_k\overline{x}_k & \leftarrow \varphi_k \end{cases}$$

ein System von IFP-Formeln ist.

Sei $\mathfrak{A}$ eine Struktur. O.B.d.A. nehmen wir an, daß $|\overline{x}_1| = \cdots = |\overline{x}_k|$ und daß $|A| \geqslant k$. Dann ist ψ äquivalent zu

$$\Psi(\overline{x}) := \exists c_1 \dots \exists c_k \bigwedge_{i \neq j} \neg c_i = c_j \wedge [\mathbf{ifp}_{R,\overline{x},c} \bigwedge_{i=1}^{k} (c = c_i \wedge \varphi_i^*(\overline{x}))](\overline{x}, c_1),$$

wobei $\varphi_i^*(\overline{x})$ aus φ_i entsteht, indem jedes Vorkommen eines Atoms $R_j\overline{u}$ durch $R\overline{u}c_j$ ersetzt wird. Die zusätzliche Komponente c in R spielt also die Rolle eines Zählers oder Markers. □

Mit dem gleichen Beweis zeigt man auch S-PFP = PFP. Selbstverständlich funktioniert dieser Beweis auch für die Logik LFP. Zusammen mit Lemma 4.11 folgt daraus folgendes Korollar.

4.14 Korollar. *Jede Formel $\varphi \in$ LFP, in der alle* lfp*-Operatoren nur positiv vorkommen, ist äquivalent zu einer Formel $\psi \in$ LFP mit nur einem* lfp*-Operator.*

Dabei werden verschachtelte Fixpunkt mit Hilfe des Lemmas zu Systemen simultaner Fixpunkte aufgelöst, die mit Hilfe des vorhergehenden Satzes dann zu einem Fixpunkt aufgelöst werden können. Ein wenig Sorgfalt ist bei Booleschen Kombinationen geboten, deren Teilformeln jeweils Fixpunktoperatoren enthalten. Dies sei zur Übung empfohlen.

4.3 Die Stage-Comparison Methode

In diesem Kapitel wird eine Methode, die sogenannte *Stage-Comparison Methode*, eingeführt, 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 der kleinsten und inflationären Fixpunktlogik nachzuweisen.

Zunächst jedoch einige Vorbemerkungen. Sei $\varphi(\overline{x}, R)$ eine Formel aus LFP oder IFP. Offensichtlich gilt $\mathbf{lfp}(\varphi) = \mathbf{lfp}(R\overline{x} \vee \varphi)$ für monotones φ und $\mathbf{ifp}(\varphi) = \mathbf{ifp}(R\overline{x} \vee \varphi)$. Wir können also immer annehmen, daß alle Formeln die in Fixpunktoperatoren auftreten diese Gestalt haben. Dies wird später noch wichtig werden. Weiterhin vereinbaren wir noch folgende Notation.

4.15 Notation. Sei σ eine Signatur und $R \notin \sigma$ ein Relationssymbol. Sei weiterhin $\varphi(\overline{x}, R)$ eine Formel.

- Ist $\psi(\overline{x})$ eine Formel, so schreiben wir $\varphi(\overline{x}, R\overline{u}/\psi(\overline{u}))$ für die Formel, die man aus φ erhält, wenn jedes Vorkommen eines Atoms der Form $R\overline{u}$, für ein Tupel $\overline{u}$ von Termen, durch die Formel $\psi(\overline{u})$ ersetzt wird.

- Sind $\psi_p(\overline{x}), \psi_n(\overline{x})$ Formeln, so schreiben wir $\varphi(\overline{x}, \textit{pos } R\overline{u}/\psi_p(\overline{u}), \textit{neg } R\overline{u}/\psi_n(\overline{u}))$ für die Formel, die man aus φ erhält, wenn man jedes *positive* Vorkommen eines Atoms der Form $R\overline{u}$ durch ψ_p und jedes *negative* Vorkommen durch $\neg\psi_n(\overline{u})$ ersetzt. Man beachte hier, daß auch bei negativen Vorkommen von $R\overline{u}$, z.B. als $\neg R\overline{u}$ nur das Atom $R\overline{u}$ ersetzt wird, nicht aber das gesamte $\neg R\overline{u}$. Aus $\neg R\overline{u}$ würde also $\neg\neg\psi_n(\overline{u})$.

  Im weiteren Verlauf des Kapitels werden wir für $\psi_p(\overline{x})$ und $\psi_n(\overline{x})$ Formeln angeben, die zu $R\overline{x}$ bzw. $\neg R\overline{x}$ äquivalent sind. Dann gilt $\varphi \equiv \varphi(\overline{x}, \textit{pos } R\overline{u}/\psi_p, \textit{neg } R\overline{u}/\psi_n)$.

Wir führen nun den zentralen Begriff der Stage-Comparison-Methode ein, die sogenannten Stage-Comparison-Relationen, die von Yannis Moschovakis in den siebziger Jahren eingeführt wurden, aber auch schon früher in anderer Form im Rahmen der Rekursionstheorie verwendet wurden.

4.16 Definition (Stage-Comparison-Relationen). Sei R ein k-stelliges Relationssymbol, $\varphi(R, \overline{x})$ eine Formel (z.B. aus FO, LFP oder IFP) und $\mathfrak{A}$ eine Struktur.

(i) Der *Rang* $|\overline{a}|_\varphi$ *eines Tupels* $\overline{a} \in A^k$ *bzgl.* φ ist definiert als

$$|\overline{a}|_\varphi := \begin{cases} \min\{i \in \mathbb{N} : \overline{a} \in R^i\} & \text{falls } \overline{a} \in \mathbf{ifp}(\varphi) \\ \infty & \text{sonst.} \end{cases}$$

Ist φ positiv in R kann hier auch der kleinste Fixpunkt verwendet werden, was aber den gleichen Rang ergibt.

(ii) Die *Stage-Comparison-Relationen* $\leqslant_\varphi$ und $\prec_\varphi$ sind wie folgt definiert. Für alle $\overline{a}, \overline{b} \in A^k$ gilt

$$\begin{array}{llll} & \overline{a} \leqslant_\varphi \overline{b} & \text{falls} \quad |\overline{a}|_\varphi \leqslant |\overline{b}| < \infty & \text{also } \overline{a}, \overline{b} \in R^\infty \\ \text{und} & & & \\ & \overline{a} \prec_\varphi \overline{b} & \text{falls} \quad |\overline{a}|_\varphi < |\overline{b}| \text{ und } |\overline{a}|_\varphi \neq \infty & \text{also } \overline{a} \in R^\infty \\ & & & \text{aber eventuell } \overline{b} \notin R^\infty. \end{array}$$

4.17 Theorem. *Sei σ eine Signatur.*

(i) Ist $\varphi(R, \overline{x})$ eine Formel in LFP*, positiv in R, so sind die Relationen $\leqslant_\varphi$ und $\prec_\varphi$ in* LFP *definierbar.*

(ii) Ist $\varphi(R, \overline{x})$ eine Formel in IFP*, so sind die Relationen $\leqslant_\varphi$ und $\prec_\varphi$ in* IFP *definierbar.*

Beweis: Wir beweisen hier nur den Teil (ii). Der Beweis von Teil (i) ist ähnlich und zusätzlich folgt die Behauptung auch aus Theorem 4.20 weiter unten.

Sei $\varphi(R, \overline{x})$ eine IFP-Formel. O.B.d.A. nehmen wir an, daß φ die Form $R\overline{x} \vee \varphi'$ hat. Wir zeigen, daß das Paar $(\leqslant_\varphi, \prec_\varphi)$ durch den simultanen Fixpunkt des Systems

$$S := \begin{cases} \overline{x} \leqslant \overline{x} & \leftarrow \varphi(\overline{x}, R\overline{u}/u \prec \overline{y}) \wedge \varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{y}) \\ \overline{x} \prec \overline{y} & \leftarrow \varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{x}) \wedge \neg\varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{x}) \end{cases}$$

definiert wird. Hierbei sind $\leqslant$ und $\prec$ $2k$-stellige Relationssymbol, die wir der Übersichtlichkeit halber in Infix-Notation schreiben.

Für $\alpha \in \mathbb{N}$ sei $(\leqslant^\alpha, \prec^\alpha)$ die α-te Stufe der (simultanen) Induktion über S.

Behauptung: Sei $\mathfrak{A}$ eine σ-Struktur. Für alle $\alpha \in \mathbb{N}$ und $\overline{a}, \overline{b} \in A^k$ gilt

(a) $(\overline{a}, \overline{b}) \in \leqslant^\alpha$ genau dann, wenn $|\overline{b}|_\varphi \leqslant \alpha$ und $|\overline{a}|_\varphi \leqslant |\overline{b}|_\varphi$.

(b) $(\overline{a}, \overline{b}) \in \prec^\alpha$ genau dann, wenn $|\overline{a}|_\varphi \leqslant \alpha$ und $|\overline{a}|_\varphi < |\overline{b}|_\varphi$.

Aus dem Beweis der Behauptung folgt sofort der Teil (ii) des Theorems.

Wir führen den Beweis der Behauptung per Induktion über α. Für $\alpha = 0$ ist die Behauptung klar, da es keine Elemente vom Rang 0 gibt.

Sei also die Behauptung schon für $\beta < \alpha$ bewiesen und sei $\overline{b}$ ein Tupel vom Rang $\xi \leqslant \alpha$. Dann ist $\xi \geqslant 1$ und $\{\overline{u} : \overline{u} \prec^{\alpha-1} \overline{b}\}$ enthält genau die Tupel $\overline{u}$ mit Rang $< \xi$. Also wird $\varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{y})$ durch $\overline{b}$ erfüllt. Wird nun $\overline{y}$ durch $\overline{b}$ interpretiert, so wird $\varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{y})$ von einem Tupel $\overline{a}$ genau dann erfüllt, wenn $|\overline{a}|_\varphi \leqslant \xi$ und somit $|\overline{a}|_\varphi \leqslant |\overline{b}|_\varphi \leqslant \alpha$.

Ist hingegen $\overline{b}$ ein Tupel mit $|\overline{b}|_\varphi > \alpha$, so kann $\overline{b}$ die Formel $\varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{y})$ nicht erfüllen. Dies zeigt Teil (a) der Behauptung.

Zu Teil (b) sei $\overline{a} \in A^k$. Gilt $|\overline{a}|_\varphi = \xi \leqslant \alpha$, so ist $\{\overline{u} : \overline{u} \prec^{\alpha-1} \overline{a}\} = \{\overline{u} : |\overline{u}|_\varphi < \xi\}$ und $\overline{a}$ erfüllt $\varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{x})$. In diesem Fall erfüllt für die Interpretation $\overline{a}$ für $\overline{x}$ ein Tupel $\overline{b} \in A^k$ die Formel $\neg\varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{x})$ genau dann, wenn $|\overline{b}|_\varphi \not\leqslant \xi$, d.h. $|\overline{b}|_\varphi > \xi$ und somit

$|\overline{b}|_\varphi > |\overline{a}|_\varphi$. Ist hingegen $|\overline{a}|_\varphi > \alpha$, so kann $\overline{a}$ die Formel $\varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{y})$ nicht erfüllen. Dies zeigt Teil (b) der Behauptung.

Also gilt $\overline{a} \leqslant_\varphi \overline{b}$ genau dann, wenn $\mathfrak{A} \models ([\mathbf{ifp}\ \leqslant: S](\overline{x},\overline{y}))[\overline{a},\overline{b}]$ und $\overline{a} \prec_\varphi \overline{b}$ genau dann, wenn $\mathfrak{A} \models ([\mathbf{ifp}\ <: S](\overline{x},\overline{y}))[\overline{a},\overline{b}]$. □

Aus den Stage-Comparison-Relationen kann nun der inflationäre Fixpunkt einer Formel definiert werden. Das zeigt folgendes Lemma, das leicht zu zeigen ist.

4.18 Lemma. *Für jede Formel $\varphi(R,\overline{x}) \in$ IFP gilt*

$$[\mathbf{ifp}_{R,\overline{x}}\, \varphi](\overline{x}) \qquad \equiv \qquad [\mathbf{ifp}\ \leqslant: S](\overline{x},\overline{x}).$$

Als nächstes werden wir zeigen, daß die inflationären Stage-Comparison-Relationen einer beliebigen LFP-Formel $\varphi(R,\overline{x})$, die also nicht in R positiv zu sein braucht, schon in LFP definiert werden können. Mit Hilfe des vorhergehenden Lemmas folgt dann die Äquivalenz von LFP und IFP auf endlichen Strukturen leicht.

4.19 Lemma. *Sei $\varphi(R,\overline{x}) \in$ LFP eine Formel, nicht unbedingt positiv in R. Dann sind die Stage-Comparison-Relationen $\leqslant_\varphi$ und $\prec_\varphi$ in LFP definierbar.*

Beweis: Zunächst betrachten wir noch einmal das System

$$S := \begin{cases} \overline{x} \leqslant \overline{x} & \leftarrow \varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{y}) \wedge \varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{y}) \\ \overline{x} \prec \overline{y} & \leftarrow \varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{x}) \wedge \neg\varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{x}) \end{cases}$$

aus dem Beweis des Theorems 4.17. 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 $\varphi(\overline{x}, R\overline{u}/\overline{u} \prec \overline{y})$ auch die Variable $\prec$ negativ vorkommt. Entsprechend kommt $\prec$ überall da in $\varphi(\overline{y}, R\overline{u}/\overline{u} \prec \overline{x})$ 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: $(R^\alpha)^c = \{\overline{a} : \overline{b} \prec \overline{a}\}$ für ein beliebiges Tupel $\overline{b}$ vom Rang α.

Wir nehmen zunächst an, daß wir die Relationen $\leqslant$ und $\prec$ schon bis zu einer Stufe $0 < \beta$ definiert hätten, d.h. es gilt $\overline{a} \leqslant^\beta \overline{b}$ genau dann, wenn $|\overline{a}|_\varphi \leqslant |\overline{b}|_\varphi \leqslant \beta$ und $\overline{a} \prec^\beta \overline{b}$ genau dann, wenn $|\overline{a}|_\varphi \leqslant \beta$ und $|\overline{a}|_\varphi < |\overline{b}|_\varphi$.

Dann gilt für alle $\overline{a},\overline{b} \in A^k$

$$(\mathfrak{A}, \leqslant^\beta, <^\beta) \models \psi_\leqslant[\overline{a},\overline{b}],$$

wobei

$$\psi_\leqslant(\overline{x},\overline{y}) := \exists\overline{z}\big(\overline{z} \prec \overline{y} \wedge \varphi(\overline{x}, pos\ R\overline{u}/\overline{u} \leqslant \overline{z}, neg\ R\overline{u}/\overline{z} \prec \overline{u}) \wedge \\ \varphi(\overline{y}, pos\ R\overline{u}/\overline{u} \leqslant \overline{z}, neg\ R\overline{u}/\overline{z} \prec \overline{u})\big),$$

genau dann, wenn $1 < |\overline{b}|_\varphi \leqslant \beta + 1$ und $|\overline{a}|_\varphi \leqslant |\overline{b}|_\varphi$.

Gilt $1 < |\overline{b}|_\varphi = \xi \leqslant \beta + 1$ für ein $\overline{b}$ so können wir für $\overline{z}$ ein Tupel mit Rang $\xi - 1$ wählen. Dann erfüllt $\overline{b}$ die Formel $\varphi(\overline{y}, pos\ R\overline{u}/\overline{u} \leqslant \overline{z}, neg\ R\overline{u}/\overline{z} \prec \overline{u})$ und $\overline{a} \in A^k$

erfüllt $\varphi(\overline{x}, pos\ R\overline{u}/\overline{u} \leqslant \overline{z}, neg\ R\overline{u}/\overline{z} \prec \overline{u})$ genau dann, wenn $|\overline{a}|_\varphi \leqslant \xi + 1 = |\overline{b}|_\varphi$. Gilt andererseits $|\overline{b}|_\varphi = \xi > \beta + 1$, so gibt es kein solches $\overline{z}$ und $\overline{b}$ erfüllt $\varphi(\overline{y}, pos\ R\overline{u}/\overline{u} \leqslant \overline{z}, neg\ R\overline{u}/\overline{z} \prec \overline{u})$ nicht.

Analog zeigt man, daß für alle $\overline{a}, \overline{b} \in A^k$

$$(\mathfrak{A}, \leqslant^\beta, <^\beta) \models \psi_\prec[\overline{a}, \overline{b}],$$

wobei

$$\begin{aligned}\psi_\prec(\overline{x}, \overline{y}) := \exists \overline{z}\big(\overline{z} \leqslant \overline{z} \wedge \varphi(\overline{x}, pos\ R\overline{u}/\overline{u} \leqslant \overline{z}, neg\ R\overline{u}/\overline{z} \prec \overline{u}) \wedge\\ \neg\varphi(\overline{y}, pos\ R\overline{u}/\neg\overline{z} \prec \overline{u}, neg\ R\overline{u}/\neg\overline{u} \leqslant \overline{z})\big),\end{aligned}$$

genau dann, wenn $|\overline{a}|_\varphi \leqslant \beta + 1$ und $|\overline{a}|_\varphi < |\overline{b}|_\varphi$.

Wir haben jetzt also Formeln $\psi_\leqslant$ und $\psi_\prec$, die für gegebene Stufe $\beta > 1$ der Induktion über $\leqslant$ und $\prec$, die nächste Stufe $\beta+1$ definieren. Wir brauchen jetzt also nur noch Formeln, die $\leqslant^1, \prec^1$ definieren. Es gilt aber für alle $\overline{a}, \overline{b} \in A^k$

- $(\overline{a}, \overline{b}) \in \leqslant^1$ genau dann, wenn $\mathfrak{A} \models \big(\varphi(\overline{x}, \varnothing) \wedge \varphi(\overline{y}, \varnothing)\big)[\overline{a}, \overline{b}]$ und
- $(\overline{a}, \overline{b}) \in \prec^1$ genau dann, wenn $\mathfrak{A} \models \big(\varphi(\overline{x}, \varnothing) \wedge \neg\varphi(\overline{y}, \varnothing)\big)[\overline{a}, \overline{b}]$.

Hierbei bedeutet $\varphi(\overline{x}, \varnothing)$, daß in φ alle Vorkommen eines Atoms $R\overline{u}$ durch *false* ersetzt wurden und analog für die anderen Formeln.

Fügen wir jetzt beide Teile zusammen so erhalten wir folgendes System

$$T := \begin{cases} \overline{x} \leqslant \overline{y} & \leftarrow \big(\varphi(\overline{x}, \varnothing) \wedge \varphi(\overline{y}, \varnothing)\big) \vee \psi_\leqslant(\overline{x}, \overline{y}) \\ \overline{x} \prec \overline{y} & \leftarrow \big(\varphi(\overline{x}, \varnothing) \wedge \neg\varphi(\overline{y}, \varnothing)\big) \vee \psi_\prec(\overline{x}, \overline{y}). \end{cases}$$

Nach Konstruktion sind alle Formeln in T positiv in $\prec$ und $\leqslant$ und es gilt

$$\begin{array}{lcl} (\mathfrak{A}, \overline{a}, \overline{b}) \models [\mathbf{lfp}\ \leqslant : T](\overline{x}, \overline{y}) & \iff & |\overline{a}|_\varphi \leqslant |\overline{b}|_\varphi \neq \infty \\ (\mathfrak{A}, \overline{a}, \overline{b}) \models [\mathbf{lfp}\ \prec : T](\overline{x}, \overline{y}) & \iff & \overline{a} \in R^\infty \text{ und } |\overline{a}|_\varphi < |\overline{b}|_\varphi \end{array}$$

□

Zusammen mit Lemma 4.18 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.

4.20 Theorem (Gurevich, Shelah, 1986). *Jede Formel $\varphi \in$ IFP ist auf endlichen Strukturen äquivalent zu einer Formel $\varphi^* \in$ LFP.*

Bemerkung. Das Theorem gilt auch im Unendlichen, ist aber aufwendiger zu beweisen.

Aus Lemma 4.19 und Korollar 4.14 folgt leicht folgender Satz.

4.21 Theorem (Immerman). *Auf endlichen Strukturen ist jeder Satz $\varphi \in$ LFP äquivalent zu einem Satz $\psi \in$ LFP mit nur einem Fixpunktoperator.*

Beweis: Zuerst zeigen wir, daß negiert vorkommende Fixpunktoperatoren durch positive ersetzt werden können. Sei dazu $\psi(R, \overline{x}) \in$ LFP eine Formel, positiv in R, und seien $\leqslant$ und $\prec$ die Stage-Comparison-Relationen für ψ, die nach Lemma 4.19 LFP-definierbar sind. Sein nun ψ_{max} die Formel

$$\psi_{\text{max}}(\overline{z}) := \overline{z} \leqslant \overline{z} \wedge \forall \overline{x}(\overline{x} \leqslant \overline{z} \vee \neg\varphi(\overline{x}, pos\ R\overline{u}/\neg\overline{z} \prec \overline{u}, neg\ R\overline{u}/\neg\overline{u} \leqslant \overline{z})).$$

Behauptung: Ist $\mathfrak{A}$ eine endliche Struktur und $(R^\alpha)_{\alpha \leqslant m}$ die Sequenz der Induktionsstufen, d.h. $R^m = R^{m+1}$ aber $R^m \neq R^{m-1}$, falls $m > 0$, und ist $m \geqslant 1$, so gilt für alle $\overline{a} \in A$:

$$\mathfrak{A} \models \psi_{\text{max}}[\overline{a}] \iff \overline{a} \in R^m \backslash R^{m-1}.$$

Beweis der Behauptung. Angenommen, $\mathfrak{A} \models \psi_{\text{max}}[\overline{a}]$, insbesondere gilt also $\overline{a} \leqslant_\psi \overline{a}$. Folglich ist $\overline{a} \in R^\infty$, d.h. es existiert ein $n \leqslant m$ mit $\overline{a} \in R^n \backslash R^{n-1}$. Weiterhin muß für alle $\overline{x}$ gelten, daß $\overline{x} \leqslant \overline{a}$, d.h. $|\overline{x}|_\psi \leqslant |\overline{a}|_\psi = n$ und somit $\overline{x} \in R^n$, oder aber $\neg\varphi(\overline{x}, pos\ R\overline{u}/\neg\overline{z} \prec \overline{u}, neg\ R\overline{u}/\neg\overline{u} \leqslant \overline{z}))$, wobei $\overline{z}$ durch $\overline{a}$ interpretiert wird. Nun ist aber

$$\{\overline{u} : \overline{a} \not\prec \overline{u}\} = \{\overline{u} : |\overline{u}|_\psi \leqslant |\overline{a}|_\psi\} = R^n,$$

und

$$\{\overline{u} : \overline{u} \not\leqslant \overline{a}\} = \{\overline{u} : |\overline{a}|_\psi < |\overline{u}|_\psi\} = (R^n)^c.$$

Wäre nun $n < m$, d.h. gäbe es ein Tupel $\overline{b} \in R^{n+1} \backslash R^n$, so würde weder $\overline{b} \leqslant_\psi \overline{a}$ noch $(\mathfrak{A}, \beta) \models \neg\varphi(\overline{x}, pos\ R\overline{u}/\neg\overline{z} \prec \overline{u}, neg\ R\overline{u}/\overline{z} \prec \overline{u}))$ gelten, wobei β die Interpretation ist, die $\overline{z}$ mit $\overline{a}$ und $\overline{x}$ mit $\overline{b}$ belegt. Es folgt also $n = m$.

Andererseits erfüllt jedes $\overline{a} \in R^m \backslash R^{m-1}$ die Formel. Dies zeigt die Behauptung.

Die Formel $\varphi(\overline{x}) := \neg[\mathbf{lfp}_{R, \overline{x}}\, \psi](\overline{x})$ ist also im Endlichen äquivalent zu

$$\varphi'(\overline{x}) := \forall \overline{y} \neg\varphi(\overline{y}, \varnothing) \vee \exists \overline{z}(\varphi_{\text{max}}(\overline{z}) \wedge \overline{z} \prec \overline{x}).$$

Weiterhin ist die Schachtelungstiefe negierter Fixpunktoperatoren in φ' geringer als in φ. Per Induktion über die Schachtelungstiefe negierter Fixpunktoperatoren folgt, daß jede Formel in LFP im Endlichen zu einer LFP-Formel äquivalent ist, in der alle Fixpunktoperatoren positiv vorkommen. Mit Korollar 4.14 folgt daraus das Theorem. ☐

Mit einem sehr ähnlichen Beweis, kann folgende Variante bewiesen werden, die eine konkrete syntaktische Form der Formeln angibt und machmal sehr nützlich ist.

4.22 Theorem. *Sei σ eine Signatur mit einem Konstantensymbol $c \in \sigma$. Für jede* LFP$[\sigma]$-*Formel φ gibt es eine* FO$[\sigma]$-*Formel χ, so daß φ auf endlichen Strukturen äquivalent zu $[\mathbf{lfp}_{R, \overline{x}}\, \chi](\overline{c})$ ist.*

Im Gegensatz zu den vorherigen Sätzen sind diese beiden Sätze im Unendlichen falsch. Auf unendlichen Strukturen können negierte Fixpunktoperatoren eben nicht durch positive kleinste Fixpunkte ausgedrückt werden.

4.4 Der Satz von Abiteboul und Vianu

Ziel dieses Kapitels ist der Beweis des folgenden Satzes von Abiteboul und Vianu.
Theorem (Abiteboul, Vianu).

$$\text{PFP} = \text{LFP auf Fin} \qquad \Longleftrightarrow \qquad \text{PFP} = \text{LFP auf Fin}_<.$$

Das heißt, zum Trennen von PFP und LFP auf geordneten Strukturen, und damit zum Trennen von PSPACE und PTIME, reicht es bereits, die Logiken auf beliebigen endlichen Strukturen zu trennen. Dies ist potentiell ein enormer Schritt, da auf geordneten Strukturen Methoden wie die Sätze von Hanf und Gaifman trivial werden und somit nicht zur Verfügung stehen.

Technisch gesehen zeigt man den Satz für PFP und IFP anstatt LFP. Die Aussage folgt dann aus dem Theorem von Gurevich und Shelah. Eine Richtung des Satzes ist natürlich trivial, die andere hingegen nicht. Zum Beweis des Satzes brauchen wir daher zunächst einige Vorbemerkungen.

4.4.1 k-Invarianten

4.23 Definition ($\sim^k, A_{/k}$). Sei $k \in \mathbb{N}_{\geqslant 1}$, σ eine relationale Signatur und $\mathfrak{A}$ eine σ-Struktur.

(a) Wir definieren eine Äquivalenzrelation $\sim^k \subseteq A^k \times A^k$ durch $\overline{a} \sim^k \overline{b}$ genau dann, wenn $\overline{a}$ und $\overline{b}$ dieselben $L^k_{\infty\omega}[\sigma]$-Formeln in $\mathfrak{A}$ erfüllen (d.h. wenn Duplicator das Spiel $G^k_\infty(\mathfrak{A}, \overline{a}, \mathfrak{A}, \overline{b})$ gewinnt).

(b) Für $\overline{a} \in A^k$ bezeichne
$$[\overline{a}] := \{\overline{b} \in A^k : \overline{a} \sim^k \overline{b}\}$$
die Äquivalenzklasse von $\overline{a}$ bzgl. $\sim^k$.

(c) $A_{/k} := \{[\overline{a}] : \overline{a} \in A^k\}$ bezeichne die Menge aller Äquivalenzklassen von $\sim^k$ in A^k. Offensichtlich ist $A^k = \bigcup_{M \in A_{/k}} M$.

4.24 Definition ($\sigma_{/k}, \mathfrak{A}_{/k}$: k-Invariante von $\mathfrak{A}$). Sei $k \in \mathbb{N}_{\geqslant 1}$, σ eine relationale Signatur.

(a) Die Signatur $\sigma_{/k}$ besteht aus

- 1-stelligen Relationssymbolen $=_{i,j}$ für alle $i, j \in \{1, \dots, k\}$
- 1-stelligen Relationssymbolen $R_{i_1,\dots,i_r}$ für alle $R \in \sigma$, $r := ar(R)$ und allen $i_1, \dots, i_r \in \{1, \dots, k\}$.
- 2-stelligen Relationssymbolen S_i für alle $i \in \{1, \dots, k\}$.

(b) Die *k-Invariante* $\mathfrak{A}_{/k}$ einer σ-Struktur $\mathfrak{A}$ ist die $\sigma_{/k}$-Struktur mit Universum $A_{/k}$ und Relationen

- $=_{i,j}^{\mathfrak{A}_{/k}} := \{[\overline{a}] : \overline{a} \in A^k, \text{ so daß } a_i = a_j\}$
- $R_{i_1,\ldots,i_r}^{\mathfrak{A}_{/k}} := \{[\overline{a}] : \overline{a} \in A^k, \text{ so daß } (a_{i_1}, \ldots, a_{i_r}) \in R^{\mathfrak{A}}\}$
- $S_i^{\mathfrak{A}_{/k}} := \{([\overline{a}], [\overline{a}']) : \overline{a}, \overline{a}' \in A^k, \text{ so daß es ein } c \in A \text{ gibt mit } \overline{a}' \sim^k \overline{a}\frac{c}{i}\}$

k-Invarianten werden eine zentrale Rolle im Beweis des Satzes von Abiteboul und Vianu spielen. Einen ersten Eindruck von ihrem Nutzen bietet folgender Satz, den wir hier ohne Beweis angeben.

4.25 Theorem. *Sei $k \in \mathbb{N}_{\geqslant 1}$, σ eine relationale Signatur und $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen. Dann gilt*

$$\mathfrak{A} \equiv^{L^k_{\infty\omega}} \mathfrak{B} \qquad \Longleftrightarrow \qquad \mathfrak{A}_{/k} \cong \mathfrak{B}_{/k}.$$

4.4.2 Der Satz von Abiteboul und Vianu

In diesem Abschnitt werden wir nun den eigentlichen Beweis des Satzes von Abiteboul und Vianu präsentieren. Er gliedert sich in mehrere Lemmas, die wir zunächst beweisen werden. Das folgende Lemma nimmt dabei eine zentrale Rolle ein.

4.26 Lemma. *Sei σ eine relationale Signatur und $k \in \mathbb{N}_{\geqslant 1}$. Es gibt eine* IFP$[\sigma_{/k}]$*-Formel $\varphi_<(x, y)$, so daß für alle (endl.) σ-Strukturen $\mathfrak{A}$ gilt:*

$$R_<^{\mathfrak{A}_{/k}} := \{([\overline{a}], [\overline{b}]) \in A_{/k} \times A_{/k} : \mathfrak{A}_{/k} \models \varphi_<[[\overline{a}], [\overline{b}]]\}$$

ist eine lineare Ordnung auf $A_{/k}$.

Beweis: Sei $\mathfrak{A}$ eine σ-Struktur. Für $j \in \mathbb{N}$ sei

$$M_{\mathfrak{A}}^j := \{(\overline{a}, \overline{b}) \in A^k \times A^k : \begin{array}{l}\text{Spoiler hat eine Gewinnstrategie in } G^k_\infty(\mathfrak{A}, \overline{a}, \mathfrak{B}, \overline{b}),\\ \text{mit der er nach höchstens } j \text{ Runden gewonnen hat}\}.\end{array}$$

Sei $M_{\mathfrak{A}}^\infty := \bigcup_{j \in \mathbb{N}} M_{\mathfrak{A}}^j$. Offensichtlich gilt $M_{\mathfrak{A}}^0 \subseteq M_{\mathfrak{A}}^1 \subseteq M_{\mathfrak{A}}^2 \subseteq \ldots$. Da A endlich ist, existiert eine Zahl $l_{\mathfrak{A}} \in \mathbb{N}$, so daß $M_{\mathfrak{A}}^\infty = M_{\mathfrak{A}}^{l_{\mathfrak{A}}}$.

Aus Theorem 3.56 folgt sofort folgender Zusammenhang.

$$\overline{a} \not\sim^k \overline{b} \quad \Longleftrightarrow \quad (\overline{a}, \overline{b}) \in M_{\mathfrak{A}}^\infty \quad \Longleftrightarrow \quad (\overline{a}, \overline{b}) \in M_{\mathfrak{A}}^{l_{\mathfrak{A}}}. \tag{0}$$

Für $j \in \mathbb{N}$ definieren wir partielle Ordnungen $<^j_{\mathfrak{A}_{/k}}$ auf $A_{/k}$, so daß gilt

$$<^0_{\mathfrak{A}_{/k}} \subseteq <^1_{\mathfrak{A}_{/k}} \subseteq <^2_{\mathfrak{A}_{/k}} \subseteq \ldots \tag{1}$$

und

$$\big([\overline{a}] <^j_{\mathfrak{A}_{/k}} [\overline{b}] \text{ oder } [\overline{b}] <^j_{\mathfrak{A}_{/k}} [\overline{a}]\big) \iff (\overline{a}, \overline{b}) \in M_{\mathfrak{A}}^j, \tag{2}$$

für alle $\overline{a}, \overline{b} \in A^k$ und $j \in \mathbb{N}$. Für $j = l$ folgt dann $\overline{a} <^j_{\mathfrak{A}_{/k}} \overline{b}$ oder $\overline{a} <^l_{\mathfrak{A}_{/k}} \overline{b}$ für alle $\overline{a}, \overline{b} \in A^k$ mit $[\overline{a}] \neq [\overline{b}]$. Das heißt, $<^l_{\mathfrak{A}_{/k}}$ ist eine lineare Ordnung auf $A_{/k}$.

Definition von $<^0_{\mathfrak{A}_{/k}}$: Für $\overline{a}, \overline{b} \in A^k$ gilt $(\overline{a}, \overline{b}) \in M^0_{\mathfrak{A}}$ genau dann, wenn es eine atomare Formel σ-Formel, d.h. eine Formel der Gestalt $x_i = x_j$ oder $Rx_{i_1} \dots x_{i_r}$, gibt, bezüglich der sich $\overline{a}$ und $\overline{b}$ unterscheiden.

Wir fixieren eine Aufzählung $\chi_1(v), \dots, \chi_m(v)$ aller atomaren Formeln $\sigma_{/k}$-Formeln der Form $=_{i,j}(v)$ oder $R_{i_1,\dots,i_r}(v)$. Für $(\overline{a}, \overline{b}) \in M^0_{\mathfrak{A}}$ gibt es nun ein minimales $s \in \{1, \dots, m\}$, so daß

$$\mathfrak{A}_{/k} \models \chi_s\big[[\overline{a}]\big] \quad \text{genau dann, wenn} \quad \mathfrak{A}_{/k} \not\models \chi_s\big[[\overline{b}]\big].$$

Falls $\mathfrak{A}_{/k} \models \chi_s\big[[\overline{a}]\big]$, so definieren wir $[\overline{a}] <^0_{\mathfrak{A}_{/k}} [\overline{b}]$. Ansonsten setzen definieren wir $[\overline{b}] <^0_{\mathfrak{A}_{/k}} [\overline{a}]$. Sei nun

$$\psi_0(x, y) := \bigvee_{s=1}^{m} \Big(\chi_s(x) \wedge \neg\chi_s(y) \wedge \bigwedge_{t=1}^{s-1} \big(\chi_t(x) \leftrightarrow \chi_t(y)\big)\Big).$$

Dann ist

$$<^0_{\mathfrak{A}_{/k}} = \{([\overline{a}], [\overline{b}]) : \mathfrak{A}_{/k} \models \psi_0\big[[\overline{a}], [\overline{b}]\big]\}.$$

ψ_0 definiert also eine partielle Ordnung auf $A_{/k}$, die die Bedingung (2) erfüllt.

Definition von $<^{j+1}_{\mathfrak{A}_{/k}}$: Sei nun $<^j_{\mathfrak{A}_{/k}}$ schon definiert. Für alle $(\overline{a}, \overline{b}) \in M^{j+1}_{\mathfrak{A}} \setminus M^j_{\mathfrak{A}}$ hat Spoiler eine Gewinnstrategie in $G^k_\infty(\mathfrak{A}, \overline{a}, \mathfrak{A}, \overline{b})$, mit der er nach $j+1$ Runden gewinnen kann. D.h. es gibt ein minimales $i \in \{1, \dots, k\}$, so daß

- es ein $c \in A$ gibt mit $(\overline{a}\frac{c}{i}, \overline{b}\frac{d}{i}) \in M^j_{\mathfrak{A}}$ für alle $d \in A$, oder
- es ein $d \in A$ gibt mit $(\overline{a}\frac{c}{i}, \overline{b}\frac{d}{i}) \in M^j_{\mathfrak{A}}$ für alle $c \in A$.

Wir definieren $[\overline{a}] <^{j+1}_{\mathfrak{A}_{/k}} [\overline{b}]$, falls es ein $c \in A$ gibt, so daß für alle $d \in A$ folgende Aussage $(*)$ gilt:

$$[\overline{a}\tfrac{c}{i}] <^j_{\mathfrak{A}_{/k}} [\overline{b}\tfrac{d}{i}] \text{ oder } \Big([\overline{b}\tfrac{d}{i}] <^j_{\mathfrak{A}_{/k}} [\overline{a}\tfrac{c}{i}] \text{ und es existiert } c' \in A \text{ so daß } (\overline{b}\tfrac{d}{i}, \overline{a}\tfrac{c'}{i}) \notin M^j_{\mathfrak{A}}\Big).$$

Wie man leicht nachrechnet, liefert dies eine partielle Ordnung $<^{j+1}_{\mathfrak{A}_{/k}}$ auf $A_{/k}$, die die Bedingungen (1) und (2) erfüllt.

Sei

$$\psi(<, x, y) := \psi_0(x, y) \vee (\neg x < y \wedge \neg y < x \wedge \bigvee_{i=1}^{k} (\varphi^*_i \wedge \bigwedge_{j=1}^{i-1} \neg\varphi^*_j),$$

wobei für $1 \leqslant i \leqslant k$, $\varphi^*_i(<, x, y)$ definiert ist als

$$\varphi^*_i := \exists x'\Big(S_i(x, x') \wedge \forall y'\Big(S_i(y, y') \to \big[x' < y' \vee (y' < x' \wedge \exists x'' \neg y' < x'' \wedge \neg x'' < y')\big]\Big)\Big).$$

Die Formel φ_i^* entspricht hier der Bedingung $(*)$ weiter oben. Wie man leicht sieht, ist $<_{\mathfrak{A}}^{j+1}$ die $(j+1)$-te Stufe der inflationären Fixpunktinduktion über ψ.

Insgesamt liefert die Formel $\varphi_<(x,y) := [\mathbf{ifp}_{<,x,y}\,\psi(<,x,y)](x,y)$ die gesuchte lineare Ordnung auf $A_{/k}$. $\square$

4.27 Lemma. *Sei σ eine relationale Signatur, $k \in \mathbb{N}_{\geqslant 1}$. Es gibt eine* IFP$[\sigma]$*-Formel $\varphi_{\sim^k}(x_1, \ldots, x_k, y_1, \ldots, y_k)$, so daß für alle (endl.) σ-Strukturen $\mathfrak{A}$ und alle $\overline{a}, \overline{b} \in A^k$ gilt:*

$$\overline{a} \sim^k \overline{b} \iff \mathfrak{A} \models \varphi_{\sim^k}[\overline{a}, \overline{b}].$$

Beweis: Übung. $\square$

4.28 Lemma. *Sei σ eine relationale Signatur, $k \in \mathbb{N}_{\geqslant 1}$. Für jede* IFP$[\sigma_{/k}]$*-Formel $\psi(x^{(1)}, \ldots, x^{(l)})$ mit l freien Variablen $x^{(1)}, \ldots, x^{(l)}$, für $l \in \mathbb{N}$, gibt es eine* IFP$[\sigma]$*-Formel $\tilde{\psi}(\overline{x}^{(1)}, \ldots, \overline{x}^{(l)})$ mit $k \cdot l$ freien Variablen $\overline{x}^{(j)} := x_1^{(j)}, \ldots, x_k^{(j)}$, für $j = 1, \ldots, l$, so daß für alle σ-Strukturen $\mathfrak{A}$ und alle Elemente $\overline{a}^{(1)} \in A^k, \ldots, \overline{a}^{(l)} \in A^k$ gilt*

$$\mathfrak{A}_{/k} := \psi\big[[\overline{a}^{(1)}], \ldots, [\overline{a}^{(l)}]\big] \iff \mathfrak{A} \models \tilde{\psi}\big[\overline{a}^{(1)}, \ldots, \overline{a}^{(l)}\big].$$

Beweis: Sei ψ eine IFP$[\sigma_{/k}]$-Formel. Die Idee der Übersetzung in eine IFP$[\sigma]$-Formel besteht darin, jede Variable y in ψ durch ein k-Tupel $\overline{y} := y_1, \ldots, y_k$ von Variablen, jede Fixpunktvariable Y der Stelligkeit r durch eine Fixpunktvariable Y' der Stelligkeit $r \cdot k$ zu ersetzen und dann folgende Übersetzung induktiv anzuwenden.

ψ ("spricht über $\mathfrak{A}_{/k}$")		$\tilde{\psi}$ ("spricht über $\mathfrak{A}$")
$y = z$	$\rightsquigarrow$	$\varphi_{\sim^k}(y_1, \ldots, y_k, z_1, \ldots, z_k)$
$R_{i_1,\ldots,i_r}(y)$	$\rightsquigarrow$	$R(y_{i_1}, \ldots, y_{i_r})$
$S_i(y,z)$	$\rightsquigarrow$	$\exists y_i \varphi_{\sim^k}(\overline{y}, \overline{z})$
$Y(y^{(1)}, \ldots, y^{(r)})$	$\rightsquigarrow$	$Y'(\overline{y}^{(1)}, \ldots, \overline{y}^{(r)})$
$\exists y \chi$	$\rightsquigarrow$	$\exists y_1 \ldots \exists y_k \tilde{\chi}$
$\forall y \chi$	$\rightsquigarrow$	$\forall y_1 \ldots \forall y_k \tilde{\chi}$
$[\mathbf{ifp}_{Y,y^{(1)},\ldots,y^{(r)}}\,\chi](z^{(1)}, \ldots, z^{(r)})$	$\rightsquigarrow$	$[\mathbf{ifp}_{Y',\overline{y}^{(1)},\ldots,\overline{y}^{(r)}}\,\tilde{\chi}](\overline{z}^{(1)}, \ldots, \overline{z}^{(r)})$

Der Nachweis, daß die so entstandene Formel $\tilde{\psi}$ die gewünschten Eingeschaften hat, ist leicht per Induktion zu führen. $\square$

Zum Beweis des Satzes von Abiteboul und Vianu brauchen wir als letztes noch bestimmte Normalformen für IFP, LFP und PFP, die in den nächsten beiden Lemmas eingeführt werden.

4.29 Lemma (Eine Normalform für PFP, IFP, LFP). *Sei σ eine relationale Signatur. Für jede* PFP$[\sigma]$*-Formel φ gibt es eine Zahl $k \in \mathbb{N}_{\geqslant 1}$ und eine* PFP$[\sigma]$*-Formel $\hat{\varphi}$, die auf* Fin *äquivalent zu φ ist (insbesondere frei$(\hat{\varphi})$ = frei(φ)) und folgende Eigenschaften hat:*

(Vk): *Es kommen nur die* FO*-Variablen $x_1, \ldots, x_k$ vor*

(Sk): *Alle vorkommenden Fixpunktvariablen haben die Stelligkeit k*

(Ak): *Jedes atomare Vorkommen einer Fixpunktvariablen X ist von der Form $X(x_1, \dots, x_k)$*

(Fk): *Jedes Vorkommen eines Fixpunktoperators ist von der Form $[\mathbf{ifp}_{X,x_1,\dots,x_k}\,\chi](x_1, \dots, x_k)$.*

Die Aussage für IFP *und* LFP *ist entsprechend.*

Beweis: In einem ersten Schritt wird φ zu einer äquivalenten PFP-Formel φ_1 transformiert, die folgende Eigenschaft hat:

(PF): Jedes Vorkommen eines Fixpunktoperators ist von der Form $[\mathbf{pfp}_{X,u_1,\dots,u_r}\,\chi](t_1, \dots, t_r)$, wobei $r := ar(X)$ und $frei(\chi) = \{u_1, \dots, u_r\}$ (d.h. χ hat keine freien Parameter).

Wir erhalten φ_1, indem wir in φ jede Teilformel der Form $[\mathbf{pfp}_{X,u_1,\dots,u_r}\,\chi(u_1, \dots, u_r, v_1, \dots, v_l)](t_1, \dots, t_r)$, mit $frei(\chi) = \{u_1, \dots, u_r, v_1, \dots, v_l\}$, durch

$$[\mathbf{pfp}_{X,u_1,\dots,u_r,v_1',\dots,v_k'}\,\chi'(u_1, \dots, u_r, v_1', \dots, v_l')](t_1, \dots, t_r, v_1, \dots, v_r)$$

ersetzen, wobei χ' aus χ entsteht, indem jedes Vorkommen eines Atoms $X(y_1, \dots, y_r)$ durch $X'(y_1, \dots, y_r, v_1', \dots, v_k')$ ersetzt wird. Dabei sind $v_1', \dots, v_k'$ neue, d.h. in χ sonst nicht vorkommende Variablen. Die so erhaltene Formel φ_1 ist äquivalent zu φ.

Im zweiten Schritt wird φ zu einer äquivalenten PFP-Formel φ_2 transformiert, die die Eigenschaften (PF) und (Vk) hat und bei der für jede Fixpunktvariable X der Stelligkeit $r := ar(X)$ die Eigenschaften (Ar) und (Fr) erfüllt sind.

Sei dazu $h \in \mathbb{N}$ so, daß alle in φ vorkommenden Fixpunktvariablen die Stelligkeit $\leqslant h$ haben und in φ_1 nur die FO-Variablen $x_1, \dots, x_h$ vorkommen. Sei ferner $k := 2h$. Man beachte, daß, nach Wahl von h, die Variablen $x_{h+1}, \dots, x_k$ nicht in φ_1 vorkommen. φ_2 entsteht jetzt aus φ_1, indem in φ_1 jedes Vorkommen einer atomaren Teilformel der Form $X(v_1, \dots, v_r)$ durch die Formel

$$\exists x_{h+1} \dots \exists x_{h+r} \Big(\bigwedge_{i=1}^{r} x_{h+i} = v_i \wedge \exists x_1 \dots x_r \bigwedge_{i=1}^{r} x_i = x_{h+i} \wedge X(x_1, \dots, x_r) \Big)$$

ersetzt wird. Des weiteren wird jede Teilformel der Form $[\mathbf{pfp}_{X,u_1,\dots,u_r}\,\chi(u_1, \dots, u_r)](t_1, \dots, t_r)$ durch die Formel

$$\begin{array}{l} \exists x_{h+1} \dots \exists x_{h+r} \bigwedge x_{h+i} = t_i \wedge \exists x_1 \dots \exists x_r \bigwedge x_i = x_{h+i} \wedge \\ [\mathbf{pfp}_{X,x_1,\dots,x_r} \begin{array}{l} \exists x_{h+1} \dots \exists x_{h+r} \bigwedge_{i=1}^{r} x_{h+i} = x_i \wedge \\ \exists u_1 \dots \exists u_r \bigwedge_{i=1}^{r} u_i = x_{h+i} \chi(u_1, \dots, u_r) \end{array}](x_1, \dots, x_r) \end{array}$$

ersetzt. Offensichtlich hat φ_2 die Eigenschaften (PF), (Vk), (Ar) und (Fr). Ebenso rechnet man leicht nach, daß φ_2 zu φ_1 äquivalent ist.

Im dritten und letzten Schritt wird die Formel φ_2 in eine äquivalente Formel $\hat{\varphi}$ transformiert, die die Eigenschaften (PF), (Vk), (Sk), (Ak) und (Fk) hat. Dazu wird in φ_2

- jede Fixpunktvariable X der Stelligkeit $r := ar(X)$ durch eine Fixpunktvariable X' der Stelligkeit k,
- jedes Vorkommen einer atomaren Teilformel der Form $X(x_1, \ldots, x_r)$ durch
$$\forall x_{r+1} \ldots \forall x_k X' x_1 \ldots x_r x_{r+1} \ldots x_k$$
und
- jede Teilformel der Form $[\mathbf{pfp}_{X,x_1\ldots x_r}\, \chi](x_1, \ldots, x_r)$ durch die Formel
$$\forall x_{r+1} \ldots \forall x_k\, [\mathbf{pfp}_{X',x_1,\ldots,x_r,x_{r+1},\ldots,x_k}\, \chi](x_1, \ldots, x_r, x_{r+1}, \ldots, x_k)$$
ersetzt.

Bei dieser Übersetzung ist es wichtig, daß φ_2 schon die Eigenschaften (Ar) und (Fr) hat.

Aus der Kontruktion folgt, daß $\hat{\varphi}$ die gewünschten Eigenschaften (PF), (Vk), (Sk), (Ak) und (Fk) hat. Ebenso rechnet man leicht nach, daß $\hat{\varphi}$ äquivalent zu φ_2 und damit zu φ ist. □

4.30 Lemma. *Sei σ eine relationale Signatur. Für jede* PFP$[\sigma]$*-Formel φ gibt es eine Zahl $k \in \mathbb{N}_{\geqslant 1}$ und eine* PFP$[\sigma_{/k}]$*-Formel $\varphi^*(x)$ mit höchstens einer freien Variablen x, so daß gilt:*

(1) *φ hat höchstens k freie* FO*-Variablen und*

(2) *für alle (endl.) σ-Strukturen $\mathfrak{A}$ und alle $\overline{a} \in A^k$ gilt $\mathfrak{A} \models \varphi[a_1, \ldots, a_k]$ genau dann, wenn $\mathfrak{A}_{/k} \models \varphi^*\big[[\overline{a}]\big]$.*

Beweis: Sei φ die gegebene PFP$[\sigma]$-Formel. Wir wählen k und $\hat{\varphi}$ gemäß Lemma 4.29, d.h. $\hat{\varphi}$ ist äquivalent zu φ und hat die Eigenschaften (PF), (Vk), (Sk), (Ak) und (Fk). Die gewünschte PFP$[\sigma_{/k}]$-Formel φ^* wird per Induktion über den Aufbau von $\hat{\varphi}$ definiert. Dabei werden FO-Variablentupel $x_1, \ldots, x_k$ durch eine einzelne Variable x repräsentiert und k-stellige Fixpunktvariablen X durch einstellige Variablen X^* ersetzt. Das Schema für die Übersetzung ist wie folgt:

$\hat{\varphi}$ (“spricht über $\mathfrak{A}$”)		φ^* (“spricht über $\mathfrak{A}_{/k}$”)
$x_i = x_j$	$\rightsquigarrow$	$=_{i,j}(x)$
$R(x_{i_1}, \ldots, x_{i_r})$	$\rightsquigarrow$	$R_{i_1,\ldots,i_r}(x)$
$X(x_1, \ldots, x_k)$	$\rightsquigarrow$	$X^*(x)$
$\exists x_i \chi$	$\rightsquigarrow$	$\exists y(S_i(x,y) \wedge (\exists x x = y \wedge \chi^*(x))$
$\forall x_i \chi$	$\rightsquigarrow$	$\forall y(S_i(x,y) \rightarrow (\exists x x = y \wedge \chi^*(x))$
$[\mathbf{pfp}_{X,x_1,\ldots,x_k}\, \chi](x_1, \ldots, x_k)$	$\rightsquigarrow$	$[\mathbf{pfp}_{X^*,x}\, \chi^*(x)](x)$
$\neg\varphi_1, \varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2$	$\rightsquigarrow$	$\neg\varphi_1^*, \varphi_1^* \wedge \varphi_2^*, \varphi_1^* \vee \varphi_2^*$

Auch hier rechnet man leicht nach, daß die Übersetzung korrekt ist, d.h. daß für alle σ-Strukturen $\mathfrak{A}$ und alle $\overline{a} \in A^k$ gilt: $\mathfrak{A} \models \hat{\varphi}[\overline{a}]$ genau dann, wenn $\mathfrak{A}_{/k} \models \varphi^*\big[[\overline{a}]\big]$. □

Nun haben wir alle nötigen Teile zusammen, um den Satz von Abiteboul und Vianu zu beweisen.

4.31 Theorem (Satz von Abiteboul und Vianu, 1991). PFP $=$ IFP *auf* Fin *genau dann, wenn* PFP $=$ IFP *auf* Fin$_<$.

Beweis: Die Hinrichtung ist natürlich klar. Zum Beweis der Rückrichtung nehmen wir an, daß PFP $=$ IFP auf Fin$_<$. Sei σ eine relationale Signatur und $\varphi \in$ PFP$[\sigma]$ ein Satz. Wir müssen zeigen, daß es einen IFP$[\sigma]$-Satz φ' gibt, so daß für alle σ-Strukturen $\mathfrak{A}$ gilt: $\mathfrak{A} \models \varphi'$ genau dann, wenn $\mathfrak{A} \models \varphi$.

Wir wählen dazu $k \in \mathbb{N}_{\geqslant 1}$ und $\varphi^*(x) \in$ PFP$[\sigma_{/k}]$ gemäß Lemma 4.30. D.h. für alle (endl.) σ-Strukturen $\mathfrak{A}$ gilt:

$$\begin{array}{lll} & \mathfrak{A} \models \varphi & \\ \iff & \text{für alle } \overline{a} \in A^k \text{ gilt } \mathfrak{A} \models \varphi[\overline{a}] & \text{da } \varphi \text{ keine freien Variablen enthält} \\ \iff & \text{für alle } \overline{a} \in A^k \text{ gilt } \mathfrak{A}_{/k} \models \varphi^*\big[[\overline{a}]\big] & \text{nach Lemma 4.30} \\ \iff & \mathfrak{A}_{/k} \models \forall x \varphi^*(x). & \end{array}$$

Also gilt

$$\mathfrak{A} \models \varphi \iff \mathfrak{A}_{/k} \models \forall x \varphi^* \iff (\mathfrak{A}_{/k}, <) \models \forall x \varphi^*(x), \tag{1}$$

für jede linear Ordnung $<$ auf $\mathfrak{A}_{/k}$. Die letzte Äquivalenz gilt, da das Symbol $<$ in φ^* gar nicht vorkommt.

Laut Voraussetzung „PFP $=$ IFP auf Fin$_<$" gibt es einen IFP$[\sigma_{/k} \dot\cup \{<\}]$-Satz $\psi^*_<$, so daß für jede lineare Ordnung $<$ auf $\mathfrak{A}_{/k}$ gilt:

$$(\mathfrak{A}_{/k}, <) \models \psi^*_< \iff (\mathfrak{A}_{/k}, <) \models \forall x \varphi^*. \tag{2}$$

Sei $\varphi_<(x,y)$ die IFP$[\sigma_{/k}]$-Formel aus Lemma 4.26 und sei ψ^* der IFP$[\sigma_{/k}]$-Satz, der aus $\psi^*_<$ entsteht, wenn jede atomare Teilformel der Form $u < v$ durch $\varphi_<(u,v)$ ersetzt wird. Dann gilt

$$\mathfrak{A}_{/k} \models \psi^* \iff (\mathfrak{A}_{/k}, R^<_{\mathfrak{A}_{/k}}) \models \psi^*_< \iff \mathfrak{A} \models \varphi. \tag{3}$$

Dabei ist $R^<_{\mathfrak{A}_{/k}}$ die lineare Ordnung aus Lemma 4.26. Die letzte Äquivalenz folgt aus (1) und (2). Gemäß Lemma 4.28 gibt es zum IFP$[\sigma_{/k}]$-Satz ψ^* einen IFP$[\sigma]$-Satz $\tilde{\psi}^*$, so daß gilt:

$$\mathfrak{A} \models \tilde{\psi}^* \iff \mathfrak{A}_{/k} \models \psi^* \iff \mathfrak{A} \models \varphi. \tag{4}$$

$\square$

5 Auswertungsspiele

5.1 Spiele und Strategien

Wir betrachten hier bestimmte Formen von Zwei-Personen Null-Summenspielen mit perfekter Information, bei denen jede Partie von einem der beiden Spieler gewonnen wird. Die Spiele werden von zwei Spielern, Spieler 0 und Spieler 1 genannt, auf gerichteten Graphen gespielt. Dabei ist die Knotenmenge des Graphen in Positionen, an denen Spieler 0 ziehen darf, und solchen, an denen Spieler 1 ziehen darf unterteilt. Das Spiel beginnt in einem vorher festgelegten Startknoten. In jedem Zug wählt der Spieler, in dessen Knotenmenge der aktuelle Knoten liegt, einen Nachfolger des Knotens aus, an dem das Spiel fortgesetzt wird. Auf diese Weise wird ein endlicher oder unendlicher Pfad durch den Graphen erzeugt. Wer eine solche Partie gewinnt, wird dann durch eine Gewinnbedingung entschieden. Formal sind die hier skizzierten Spiele wie folgt definiert.

5.1 Definition. *(i)* Eine *Arena* ist ein Tripel $\mathcal{A} := (V, V_0, E)$, so daß (V, E) ein gerichteter Graph und $V_0 \subseteq V$ eine Menge von Knoten ist. Hierbei sind V die möglichen Spielpositionen, V_0 die Positionen für Spieler 0.

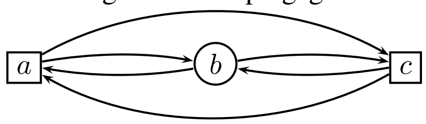
(ii) Ein *Spiel* $\mathcal{G} := (V, V_0, E, v_0, \Omega)$ besteht aus einer Arena $\mathfrak{A} := (V, V_0, E)$, einer Anfangsposition $v_0 \in V$ und einer *Gewinnbedingung* $\Omega \subseteq V^\omega$. Hierbei bezeichnet V^ω die Menge aller unendlichen Pfade durch (V, E).

5.2 Notation.
- Manchmal lassen wir den Startknoten eines Spiels weg und schreiben einfach (V, V_0, E, Ω).
- Wir schreiben V_1 für die Menge $V \setminus V_0$, an denen Spieler 1 ziehen kann.
- Ist $\mathcal{A} := (V, V_0, E)$ eine Arena, so schreiben wir $\mathcal{G} := (\mathcal{A}, v_0, \Omega)$ für das Spiel (V, V_0, E, v_0, Ω).
- Spieler werden wir oft mit $\rho \in \{0, 1\}$ bezeichnen. Mit $\bar{\rho} = 1 - \sigma$ bezeichnen wir dann den Gegenspieler.
- Ist (V, E) ein gerichteter Graph und $v \in V$, so bezeichnet $N(v) := \{u : (v, u) \in E\}$ die Menge aller Nachfolger von v.

5.3 Beispiel. (a) Gegeben sei folgendes Spiel $\mathcal{G} := (V, V_0, E, v_0, \Omega)$, wobei $V := \{a, b, c\}$, $V_0 := \{b\}$, $v_0 := b$ und

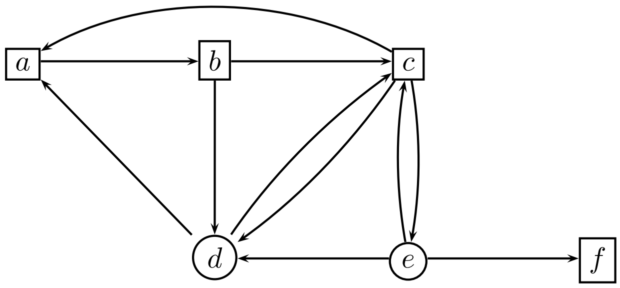
$$\Omega = \{v_0 v_1 \ldots : \begin{array}{l} \text{es gibt unendlich viele } i \in \mathbb{N} \text{ mit } v_i = a \\ \text{und unendlich viele } i \text{ mit } v_i = c \end{array}\}.$$

Die Kantenrelation ist gemäß folgendem Graph gegeben.



Wir werden im folgenden immer Spiele als solche Graphen darstellen, wobei Kästen den Knoten für Spieler 1 und Kreise den Knoten für Spieler 0 entsprechen. Den Anfangsknoten werden wir manchmal doppelt umrahmen, meistens aber weglassen.

(b) Als zweites Beispiel soll das Spiel $\mathcal{G}'$ dienen, bestimmt durch folgenden Graph.



Die Gewinnbedingung ist

$$\Omega := \{v_1 v_2 : \text{es gibt unendlich viele } i \text{ mit } v_i = a\}.$$

5.4 Definition (Partien). Sei $\mathcal{A} := (V, V_0, E)$ eine Arena.

(i) Eine *Partie* auf $\mathcal{A}$ ist ein Pfad im Digraph (V, E).

$$\mathcal{P}(\mathcal{A}) := \{\overline{v} : \overline{v} \text{ ist eine Partie auf } \mathcal{A}\}$$

bezeichnet die Menge aller Partien auf $\mathcal{A}$ und

$$\mathcal{P}(\mathcal{A}, v_0) := \{\overline{v} : \overline{v} \text{ ist eine Partie auf } \mathcal{A}, \text{ die in } v_0 \text{ beginnt } \}$$

bezeichnet die Menge aller Partien mit Anfangsknoten v_0.

(ii) Sei $\mathcal{G} := (\mathcal{A}, v_0, \Omega)$ ein Spiel. Spieler 0 *gewinnt* eine Partie $\overline{v} \in \mathcal{P}(\mathcal{A}, v_0)$ im Spiel $\mathcal{G}$, falls

- $\overline{v} := v_1 \dots v_n$ endlich ist und $v_n \in V_1 \quad (= V \setminus V_0)$ oder
- $\overline{v} := v_0 v_1 v_2 \dots$ unendlich ist und $\overline{v} \in \Omega$.

Ansonsten gewinnt Spieler 1.

Die vorherige Definition beschreibt, welcher der beiden Spieler eine bestimmte Partie gewinnt. Meistens interessiert man sich jedoch weniger für eine einzelne Partie, sondern man möchste wissen, ob einer der beiden Spieler eine Strategie hat, alle Partien zu gewinnen, egal was der Gegenspieler macht. Die dazu nötigen Begriffe liefert die nächste Definition.

5.5 Definition (Strategien und Gewinnstrategien). Sei $\mathcal{G} := (V, V_0, E, v_0, \Omega)$ ein Spiel.

(i) Eine *Strategie* für Spieler $\rho \in \{0, 1\}$ ist eine partielle Abbildung $f : V^* \to V$, so daß für alle Wege $\overline{v} := v_0 \dots v_n$ mit $v_n \in V_\rho$ das Bild $f(\overline{v})$ definiert ist und $f(\overline{v}) \in N(v_n)$.

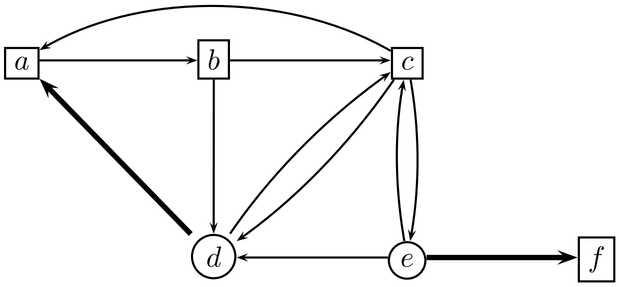
(ii) Ein Weg oder eine Partie $\overline{v} \in V^* \cup V^\omega$ ist *konform mit einer Strategie* f für Spieler ρ, kurz f-konform, wenn für alle $i \geqslant 0$ gilt: Ist $v_i \in V_\rho$, so ist $v_{i+1} = f(v_0 \dots v_i)$.

(iii) Eine Strategie f für Spieler ρ heißt *Gewinnstrategie*, falls Spieler ρ jede f-konforme Partie gewinnt.

(iv) Ein Spieler gewinnt ein Spiel, wenn er eine Gewinnstrategie hat.

5.6 Beispiel (Fortsetzung von Beispiel 5.3). (a) Wie betrachten noch einmal das Spiel Spiel $\mathcal{G} := (V, V_0, E, v_0, \Omega)$ aus Beispiel 5.3 (a). Offenbar hat Spieler 0 eine Gewinnstrategie indem er immer abwechselnd von Knoten b nach a und beim nächsten Mal nach c zieht. Auf diese Weise muß entweder Spieler 1 immer zwischen a und c pendeln, oder aber es wird unendlich oft b durchlaufen und, gemäß der Strategie von Spieler 0, auch unendlich oft a und c.

(b) Auch im Spiel $\mathcal{G}'$ hat Spieler 0 eine Gewinnstrategie von jedem Knoten aus, die durch die fett markierten Pfeile angedeutet wird.



Man beachte den Unterschied zwischen den beiden Gewinnbedingungen von Spieler 0. Im Fall (a) muß sich Spieler 0 jeweils merken, ob er beim letzten Mal nach a oder nach c gegangen ist. Er braucht also Speicher. Im Fall (b) hingegen hängt die Entscheidung für einen Nachfolgeknoten nur vom aktuellen Knoten ab, nicht aber von der Historie des Spiels. Solche Gewinnstrategien werden *speicherfrei* oder *positional* genannt und sind von besonderer Bedeutung. Wie man leicht sieht, hat Spieler 0 in $\mathcal{G}$ keine positionale Gewinnstrategie.

5.7 Definition. Eine (Gewinn-) Strategie für Spieler ρ in einem Spiel $\mathcal{G} := (V, V_0, E, v_0, \Omega)$ heißt *positional* oder *speicherfrei*, falls für alle Pfade $\overline{v} := v_0 \dots v_n$ und $\overline{w} := w_0 \dots w_m$ mit $v_n = w_m \in V_\rho$ gilt $f(\overline{v}) = f(\overline{w})$.

Bemerkung: Speicherfreie Strategien für Spieler ρ können wir als Abbildungen $f : V_\rho \to V$ ansehen, da es auf den Verlauf der Partie nicht ankommt. Ist V endlich, so ist die Größe von f, d.h. die Größe des Graphs von f, polynomial in $|V|$.

5.8 Definition. Ein Spiel $\mathcal{G}$ heißt *(positional) determiniert*, falls einer der Spieler eine (positionale) Gewinnstrategie hat.

In einem determinierten Spiel gibt es also immer einen Spieler, der eine Strategie hat, um alle Partien zu gewinnen. Determiniertheit ist eine für sehr viele Anwendungen solcher Spiele essentielle Eigenschaft, zum Beispiel im Bereich der Controller-Synthese oder des Model-Checkings. Leider kann dies im allgemeinen nicht vorausgesetzt werden, wie folgendes Theorem zeigt.

5.9 Theorem (Gale, Stewart). *Es gibt nicht-determinierte Spiele.*

Glücklicherweise sind aber alle für uns bedeutenden Klassen von Spielen determiniert. Insbesondere sind

- alle endlichen Spiele, also solche, deren Partien sämtlich endlich sind, determiniert. Unter diese Klasse fallen auch die meisten Gesellschaftsspiele, wie z.B. Schach. Im Schachspiel hat also einer der beiden Spieler eine Gewinnstrategie (oder aber beide eine Strategie, um ein Unentschieden zu erzwingen). Wir wissen nur nicht, welcher der beiden.

- alle *Borel-Spiele* determiniert. Borel-Spiele sind eine Klasse von Spielen, bei denen die Gewinnbedingung Ω eine sogenannte Borel-Menge ist. Borel-Mengen sind Mengen einer transfiniten Hierarchie von Mengen, die von dem Mathematiker Borel eingeführt wurden. Die meisten in der Informatik betrachteten Gewinnbedingungen fallen unter diese Klasse von Spielen, sind also determiniert. Der Nachweis der Determiniertheit von Borel-Spielen, von Martin in den siebzigern geführt, ist eine der wichtigen Resultate in der deskriptiven Mengenlehre.

5.2 Spielsemantik der Logik erster Stufe

Neben der üblichen *kompositionalen* Semantik der Logik erster Stufe, die auf den Mathematiker Alfred Tarski zurückgeht, kann man die Semantik von FO auch spieltheoretisch begründen. Ein Satz $\psi \in \text{FO}$ und eine Struktur $\mathfrak{A}$ definieren dieser Semantik gemäß ein *Auswertungsspiel* $\mathcal{G}(\mathcal{A}, \psi)$. Dabei versucht Spieler 0, hier üblicherweise als *Verifiziererin* bezeichnet, zu zeigen, daß $\mathfrak{A} \models \psi$, wohingegen Spieler 1, oft als *Falsifizierer* bezeichnet, versucht, das Gegenteil zu beweisen.

5.10 Definition. Sei σ eine Signatur, $\psi \in \text{FO}[\sigma]$ ein Satz in Negations-Normalform und $\mathfrak{A} := (A, \sigma^{\mathfrak{A}})$ eine σ-Struktur.

(i) $\text{cl}(\psi) := \{\varphi : \varphi \text{ ist eine Unterformel von } \psi\}$ bezeichnet die Menge aller Unterformeln von ψ.

(ii) Das *Auswertungspiel* $\mathcal{G}(\mathfrak{A}, \psi) := (V, V_0, E, v_0, \Omega)$ für $\mathfrak{A}$ und ψ ist wie folgt definiert.

- V ist die Menge aller Paare (β, φ), wobei $\varphi \in \mathrm{cl}(\psi)$ und $\beta : \mathit{frei}(\varphi) \to A$ eine Belegung der freien Variablen von φ ist.
- V_0 ist die Menge aller Paare $(\beta, \varphi) \in V$, so daß φ die Gestalt $\varphi_1 \vee \varphi_2$ oder $\exists x \varphi'$ oder aber φ ein Literal ist und $(\mathfrak{A}, \beta) \not\models \varphi$.
- Es gibt eine Kante zwischen zwei Knoten (β, φ) und (β', φ'), falls
  - $\varphi := \varphi_1 \wedge \varphi_2$ oder $\varphi := \varphi_1 \vee \varphi_2$ und $\beta = \beta'$ sowie $\varphi' = \varphi_1$ oder $\varphi' = \varphi_2$.
  - $\varphi := \forall x \varphi_1$ oder $\varphi := \exists x \varphi_1$ und $\varphi' = \varphi_1$ sowie $\beta' = \beta[x/a]$ für ein $a \in A$, d.h.für alle Variablen $y \neq x$ gilt $\beta'(y) = \beta(y)$ und $\beta'(x) = a$.
- Schließlich ist $v_0 := (\varnothing, \psi)$, wobei $\varnothing$ die leere Interpretation ist.

Die Knoten des Auswertungsspiels $\mathcal{G}(\mathfrak{A}, \psi)$ bestehen also aus Paaren (β, φ), wobei φ eine Unterformel von ψ und β eine Interpretation der freien Variablen von ψ ist. Nach Konstruktion ist das Spiel azyklisch und zusammenhängend, also ein Baum. Die Blätter des Baums sind die Knoten (β, φ), wobei φ ein Literal ist. Hier ist das Spiel beendet und die Verifiziererin gewinnt genau dann, wenn $(\mathfrak{A}, \beta) \models \varphi$. An inneren Knoten zieht die Verifiziererin an Existenzquantoren $\exists x \varphi'$, wobei sie dann ein Element a für x auswählen kann, und an Disjunktionen $\varphi_1 \vee \varphi_2$, bei denen sie eine Teilformel wählen darf. Analog zieht der Falsifizierer bei Konjunktionen und Allquantoren.

Es sollte nun leicht einzusehen sein, daß die Verifiziererin genau dann eine Gewinnstrategie von $(\varnothing, \psi)$ aus hat, wenn $\mathfrak{A} \models \psi$.

5.11 Theorem. *Für alle Sätze $\psi \in \mathrm{FO}[\sigma]$ und alle σ-Strukturen $\mathfrak{A}$ gilt: Die Verifiziererin hat genau dann eine Gewinnstrategie in $\mathcal{G}(\mathfrak{A}, \psi)$, wenn $\mathfrak{A} \models \psi$.*

5.3 Paritätsspiele

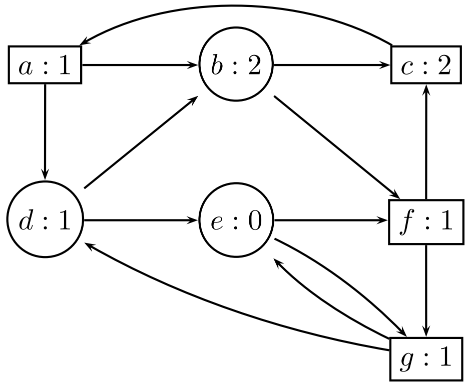
Ähnlich wie im letzten Abschnitt können auch Auswertungsspiele für kleinste Fixpunktlogiken definiert werden. Die entsprechende Klasse von Spielen, die dabei verwendet werden, sind die sogenannten Paritätsspiele, die in diesem Abschnitt eingeführt werden.

5.12 Definition. Ein *Paritätsspiel* $\mathcal{G} := (V, V_0, E, v_0, \pi)$ besteht aus einer Arena $\mathcal{A} := (V, V_0, E)$, einem Startknoten $v_0 \in V$, sowie einer Abbildung $\pi : V \to \mathbb{N}$. π heißt *Prioritätsfunktion* und $\pi(v)$ die *Priorität* des Knotens v. Für eine unendliche Partie $\overline{v} := v_0 v_1 \ldots$ bezeichnet

$$\mathrm{Inf}(\pi(\overline{v})) := \{n \in \mathbb{N} : \text{es gibt unendlich viele } i \geqslant 0 \text{ mit } \pi(v_i) = n\}$$

die Menge aller unendlich oft in $\overline{v}$ vorkommenden Prioritäten. Eine Partie $\overline{v}$ in dem Paritätsspiel $\mathcal{G}$ wird von Spieler 0 gewonnen, wenn $\min \mathrm{Inf}(\pi(\overline{v}))$ gerade ist. Formal induziert das Paritätsspiel $\mathcal{G}$ also das Spiel (V, V_0, E, v_0, Ω) mit $\Omega := \{\overline{v} \in V^\omega : \min \mathrm{Inf}(\pi(\overline{v})) \text{ gerade }\}$.

5.13 Beispiel. Das Paritätsspiel $\mathcal{G}$ sei wie folgt durch einen Spielgraphen gegeben.



Dabei entsprechen Kreise den Positionen für Spieler 0 und Kästen denen für Spieler 1. Die Beschriftung der Knoten, z.B. $a : 1$, gibt an, daß der Knoten a die Priorität 1 hat. Ist der Knotenname unwichtig, schreiben wir nur die Priorität an den Knoten. Offenbar gewinnt Spieler 1 von den Knoten a, b, c, f und Spieler 0 von den Knoten d, e und g.

Die durch die Paritätsbedingung induzierten Gewinnbedingungen Ω sind Borel-Mengen. Wie am Ende von Abschnitt 5.1 bemerkt, folgt daraus die Determiniertheit von Paritätsspielen. Wir werden aber im folgenden noch eine weitaus stärkere Eigenschaft zeigen, nämlich daß Paritätsspiele sogar positionale Gewinnstrategien erlauben.

5.3.1 Positionale Determiniertheit und Komplexität

Ziel dieses Abschnittes ist der Nachweis, daß in jedem Paritätsspiel und von jedem Knoten eines solchen Spiels, einer der beiden Spieler eine positionale Gewinnstrategie hat. Dazu werden zunächst noch einige wichtige Begriffe eingeführt. Da die folgenden Begriffe nicht nur für Paritätsspiele nützlich sind, führen wir sie direkt für allgemeine Spiele ein. Für den Rest dieses Abschnitts nehmen wir an, daß jeder Knoten eines Spiels mindestens einen Nachfolger hat. Es gibt also keine endlichen Partien.

5.14 Definition (Attraktoren und Fallen). Sei $\mathcal{G} := (V, V_0, E, \Omega)$ ein Spiel und sei $U \subseteq V$ eine Knotenmenge. Für $\rho \in \{0, 1\}$ ist die *Attraktormenge* $\mathrm{Attr}_\rho(\mathcal{G}, U)$ *von Spieler* ρ *für* U *in* $\mathcal{G}$ definiert als die Menge aller Knoten $v \in V$, von denen aus Spieler ρ eine Strategie hat, um das Spiel in endlich vielen Zügen nach U zu zwingen.

Dual dazu ist eine Menge U eine *Falle für Spieler* ρ, kurz eine ρ-Falle, falls Spieler $\bar{\rho}$ von jedem Knoten aus U eine Strategie hat, um das Spiel unendlich lang in U zu halten.

Das folgende Lemma zeigt, wie die Attraktormenge einer Menge U sowie eine positionale Strategie berechnet werden kann, die es erlaubt, das Spiel von jedem Knoten der Attraktormenge nach U zu zwingen.

5.15 Lemma (Attraktorlemma). *Sei* $\mathcal{G} := (V, V_0, E, \Omega)$ *ein Spielgraph und* $U \subseteq V$. *Sei* $\rho \in \{0, 1\}$ *und sei* $(U_j)_{j \geqslant 0}$ *die Sequenz von Mengen induktiv definiert durch* $U_0 = U$ *und*

$$\begin{aligned} U_{j+1} := U_j \quad & \cup \quad \{u \in V_\rho : \exists v (Euv \wedge v \in U_j)\} \\ & \cup \quad \{u \in V_{\bar{\rho}} : \forall v (Euv \rightarrow v \in U_j)\}. \end{aligned}$$

Dann ist $\mathrm{Attr}_\rho(\mathcal{G}, U) = \bigcup_{j \geqslant 0} U_j$. *Weiterhin erhält man eine positionale Strategie für Spieler* ρ, *die es ihm erlaubt, das Spiel mindestens einmal nach* U *zu zwingen, indem man für jeden* V_0*-Knoten in* $U_{j+1} \backslash U_j$ *zu einem beliebigen Knoten in* U_j *zieht.*

Ist $\mathcal{G}$ *endlich, ist* $\mathrm{Attr}_\rho(\mathcal{G}, U) = U_j$ *für das erste* $j \geqslant 0$ *mit* $U_j = U_{j+1}$ *und somit berechenbar.*

Es ist klar, daß Spieler ρ von keinem Knoten außerhalb von $\mathrm{Attr}_\rho(\mathcal{G}, U)$ einen Zug nach U erzwingen kann. Andernfalls wäre der Knoten bereits Teil der Attraktormenge. Das heißt, das Komplement einer Attraktormenge ist eine Falle für Spieler ρ.

5.16 Bemerkung. Das Komplement einer Attraktormenge W für Spieler ρ innerhalb eines Spiel $\mathcal{G}$ ist eine ρ-Falle und induziert ein Teilspiel, das mit $\mathcal{G} \backslash W$ bezeichnet wird.

Teilspiel heißt hierbei, daß jeder Knoten weiterhin einen Nachfolger hat. Als letzte Vorbereitung für den Beweis der positionalen Determiniertheit beweisen wir noch folgendes Lemma, welches es erlaubt, positionale Gewinnstrategien für Knoten in Gewinnmengen zu vereinigen.

5.17 Lemma. *Sei* $\mathcal{G} := (V, V_0, E, \pi)$ *ein Paritätsspiel und sei* W_0 *die Menge der Knoten, von denen aus Spieler* 0 *eine positionale Gewinnstrategie hat. Dann gilt* $W_0 = \mathrm{Attr}_0(\mathcal{G}, W_0)$ *und es gibt eine positionale Strategie für Spieler* 0, *mit der er von jedem Knoten aus* W_0 *gewinnt.*

Beweis: Wir zeigen das Lemma hier nur für endliche Spiele. Zunächst ist klar, daß

$$\mathrm{Attr}_0(\mathcal{G}, W_0) = W_0.$$

Sei nun für jeden Knoten $u \in W_0$ f_u eine positionale Gewinnstrategie von u aus. Sei $\mathcal{G}' = (W_0, E') := <W_0>^{\mathcal{G}}$ der von W_0 induzierte Teilgraph des Spiels $\mathcal{G}$. Jede Strategie f_0 induziert wiederum einen Teilgraph $\mathcal{G}_u$ von $\mathcal{G}'$, indem man alle Kanten aus $\mathcal{G}'$ streicht, die nicht zur Strategie passen, d.h. alle Kanten $(v, w) \in E'$ mit $v \in V_0$ aber $w \neq f_u(v)$, und dann alle Knoten entfernt, die nicht länger von u aus erreichbar sind. Sei $\leqslant$ eine beliebig gewählte lineare Ordnung auf W_0.

Wir definieren eine Strategie $f : W_0 \to W_0$ für Spieler 0 wie folgt. Für jeden Knoten $v \in W_0$ sei $\gamma(v)$ der bezüglich $\leqslant$ kleinste Knoten $u \in W_0$, so daß v in $\mathcal{G}_u$ vorkommt. Für alle $v \in W_0$ definieren wir $f(v) := f_{\gamma(v)}(v)$ und behaupten, daß dies eine positionale Gewinnstrategie für Spieler 0 in $\mathcal{G}$ von jedem Knoten aus W_0 ist. Sei dazu $v_0 \in W_0$ und $\overline{v} := v_0 v_1 \dots$ eine f-konforme Partie. Da für jeden Knoten $v_i \in V_0$ Spieler 0 gemäß einer seiner Gewinnstrategien spielt und W_0 seine eigene Attraktormenge ist, kann $\overline{v}$ nie die Menge W_0 verlassen. Für $i < j$ und $v_i, v_j \in V_0$ gilt weiterhin $\gamma(v_i) \geqslant \gamma(v_j)$, d.h. die auftretenden Werte $\gamma(v_i)$ werden nicht größer. Daraus folgt, daß es einen Knoten u geben muß, so daß ab einem $i \in \mathbb{N}$ für alle Knoten $v_j \in V_0$ mit $j > i$ gilt $\gamma(v_j) = u$. Das bedeutet, daß ab einem bestimmten Punkt der Partie Spieler 0 nur noch einer Gewinnstrategie f_u folgt. Also wird $\overline{v}$ von Spieler 0 gewonnen. $\square$

Derselbe Beweis funktioniert auch für Spiele mit unendlichen Spielgraphen. Anstelle einer linearen Ordnung wählt man dann eine Wohlordnung, d.h. eine lineare Ordnung ohne unendlich absteigende Ketten.

5.18 Theorem. *Paritätsspiele sind positional determiniert, d.h. von jedem Knoten eines Paritätsspiels hat genau einer der beiden Spieler eine positionale Gewinnstrategie.*

Beweis: Sei $\mathcal{G} := (V, V_0, E, \pi)$ ein Paritätsspiel. Der Beweis der Behauptung wird per Induktion über die Anzahl n der auftretenden Prioritäten geführt. Für $n = 1$ ist nichts zu beweisen. Sei also $n > 1$. Wir nehmen an, daß die kleinste auftretende Priorität ungerade ist, ansonsten vertauschen wir die Rolle der Spieler im folgenden Beweis.

Sei P die Menge aller Knoten mit minimaler Priorität und sei W_0 die Menge aller Knoten, von denen aus Spieler 0 eine positionale Gewinnstrategie hat. Zu zeigen ist, daß Spieler 1 von jedem Knoten $v \in V \setminus W_0$ eine positionale Gewinnstrategie besitzt.

Bezeichnet $W = V \setminus W_0$ das Komplement von W_0 in $\mathcal{G}$, so ist nach Bemerkung 5.16 klar, daß W eine 0-Falle und $\mathcal{G} \setminus W_0$ ein Teilspiel ist. Ist nun $W \cap P = \varnothing$, so folgt die Behauptung sofort aus der Induktionsvoraussetzung. Wir nehmen also an, daß $W \cap P \neq \varnothing$, d.h. es gibt Knoten minimaler Priorität in W. Sei jetzt

$$Y \quad = \quad \mathrm{Attr}_1(\mathcal{G} \setminus W_0, W \cap P)$$

die Attraktormenge dieser Knoten minimaler Priorität in W. Wiederum bildet die Menge $Z = W \setminus Y$ eine 1-Falle und somit ein Teilspiel $(\mathcal{G} \setminus W_0) \setminus Y$ von $\mathcal{G} \setminus W_0$. In dem Spiel $(\mathcal{G} \setminus W_0) \setminus Y$ kommen keine Knoten minimaler Priorität mehr vor. Laut Induktionsvoraussetzung zerfällt Z also in zwei disjunkte Mengen Z_0, Z_1, so daß Spieler ρ eine positionale Gewinnstrategie von jedem Knoten in Z_ρ aus hat. Da Z aber eine 1-Falle ist, d.h. Spieler 1 kann das Spiel nicht aus Z heraus zwingen, ist jede Gewinnstrategie in $(\mathcal{G} \setminus W_0) \setminus Y$ für Spieler 0 von einem Knoten $v \in Z_0$ aus auch eine Gewinnstrategie von v aus in $\mathcal{G}$. Somit wäre v bereits in W_0. Es folgt also $Z_0 = \varnothing$. Ähnlich wie in Lemma 5.17 können wir jetzt die einzelnen Gewinnstrategien von Spieler 1 für die Knoten in Z zu einer einzigen Strategie f kombinieren.

Diese Strategie wird nun zu einer positionalen Strategie g für Spieler 1 auf allen Knoten in W erweitert. Für Knoten $v \in Y$ benutzen wir die in Lemma 5.15 konstruierte Strategie. Wird ein Knoten $v \in P \cap W$ erreicht, kann Spieler 1 auf jeden Fall wieder nach W ziehen, falls $v \in V_1$, oder alle Nachfolger von v liegen in W, falls $v \in V_0$. Andernfalls würde das Spiel von v in W_0 weitergehen und v wäre somit selbst in W_0. Für Knoten in Z benutzen wir die Strategie f.

Für jede g-konforme Partie von einem Knoten $v \in W$ aus gibt es jetzt nur zwei Möglichkeiten. Entweder bleibt sie ab einem bestimmten Punkt in Z oder aber sie kehrt unendlich oft nach Y zurück. Im ersten Fall ist die Partie ab einem Punkt f-konform und somit für Spieler 1 gewinnend. Andernfalls stellt die Attraktor-Strategie sicher, daß das Spiel unendlich oft einen Knoten aus $P \cap W$ durchläuft. Da dies die Knoten kleinster Priorität sind und diese nach Voraussetzung ungerade ist, gewinnt also Spieler 1 auch diese Partien. □

5.19 Notation. Für ein Paritätsspiel $\mathcal{G} := (V, V_0, E, \pi)$ und einen Spieler $\rho \in \{0, 1\}$ bezeichnet

$$W_\rho(\mathcal{G}) := \{v \in V : \text{Spieler } \rho \text{ gewinnt } \mathcal{G} \text{ von } v \text{ aus }\}$$

die *Gewinnmenge* von Spieler ρ in $\mathcal{G}$.

5.20 Theorem. *Das Problem*

PARITY:	
Eingabe:	*Paritätsspiel* $\mathcal{G} := (V, V_0, E, \pi)$ *Knoten* $v_0 \in V$
Problem:	*Entscheide, ob* v_0 *eine Gewinnposition für Spieler* 0 *ist*

ist in NP $\cap$ CO-NP.

Beweis:

Wir zeigen zunächst, daß PARITY $\in$ NP. Sei also $\mathcal{G} := (V, V_0, E, \pi)$ ein Paritätsspiel und sei $n := |V|$. O.B.d.A. nehmen wir an, daß alle in $\mathcal{G}$ auftretenden Prioritäten kleiner oder gleich n sind. (Man überlege sich, daß dies keine Einschränkung der Allgemeinheit ist.) Ein Knoten v liegt in der Gewinnmenge für Spieler 0 genau dann, wenn Spieler 0 eine positionale Gewinnstrategie $f : V \to V$ von v aus hat. f, d.h. der Graph von f, ist also polynomial groß in n und kann somit geraten werden. Aus $\mathcal{G}$ und f konstruieren wir das Spiel $\mathcal{G}_f = (V, V_0, E', \pi)$ mit

$$E' = \{(v, w) \in E : v \in V_1\} \cup \{(v, f(v)) : v \in V_0\}.$$

Wir entfernen also aus $\mathcal{G}$ alle Kanten von einem Knoten für Spieler 0 aus, die nicht seiner Gewinnstrategie entsprechen. In $\mathcal{G}_f$ hat nur noch Spieler 1 eine echte Auswahl, Spieler 0 hingegen hat nur jeweils eine Möglichkeit zu ziehen. Weiterhin sind offensichtlich alle Partien in $\mathcal{G}_f$ f-konform. Kann Spieler 1 also eine Partie in $\mathcal{G}_f$ gewinnen, so ist f keine Gewinnstrategie für 0 in $\mathcal{G}$.

Ein Zyklus $\mathcal{C}$ in $\mathcal{G}_f$ heißt *ungerade*, wenn die kleinste auf $\mathcal{C}$ vorkommende Priorität ungerade ist. Spieler 1 gewinnt nun $\mathcal{G}_f$ genau dann, wenn es einen ungeraden Zyklus $\mathcal{C}$ in $\mathcal{G}_f$ gibt, der von v_0 aus erreichbar ist. Dies kann in Polynomialzeit überprüft werden.

Insgesamt erhalten wir also folgenden NP-Algorithmus:

(1) Rate Strategie $f : V \to V$ für Spieler 0.

(2) Konstruiere das Spiel $\mathcal{G}_f$.

(3) Überprüfe, ob Spieler 1 das Spiel $\mathcal{G}_f$ gewinnen kann.

Zum Nachweis von PARITY $\in$ CO-NP wird genauso verfahren, nur raten wir eine Strategie für Spieler 1 und suchen dann nach geraden Zyklen. Dabei nutzen wir aus, daß Spieler 0 genau dann *keine* Gewinnstrategie hat, wenn Spieler 1 eine Gewinnstrategie hat. □

Bemerkung. Die besten deterministischen Algorithmen für das Paritätsspielproblem haben eine Laufzeit von $\mathcal{O}\big(d \cdot |E| \cdot \big(\frac{|V|}{\lfloor \frac{d}{2} \rfloor}\big)^{\lfloor \frac{d}{2} \rfloor}\big)$, wobei d die Anzahl der verschiedenen Prioritäten ist.

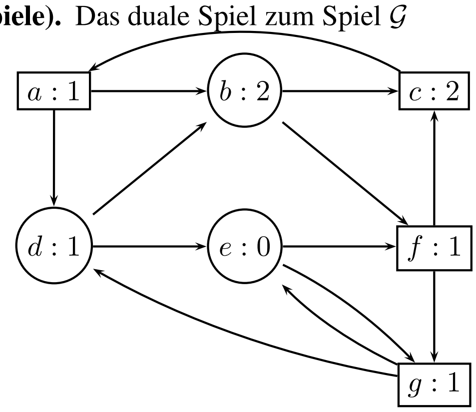
Weiter oben wurde schon erwähnt, daß Paritätsspiele die Auswertungsspiele für kleinste Fixpunktlogiken sind. Als Vorbereitung auf den Beweis dieses Zusammenhangs, führen wir zunächst noch einige Begriffe und Methoden ein, die dann im nächsten Abschnitt benutzt werden.

5.21 Definition (Spiel-Dualisierung). Sei $\mathcal{G} := (V, V_0, E, \pi)$ ein Paritätsspiel. Das *duale Spiel* $\mathcal{G}^d$ *zu* $\mathcal{G}$ ist definiert als $\mathcal{G}^d = (V, V \setminus V_0, E, \pi')$, wobei für alle $v \in V$

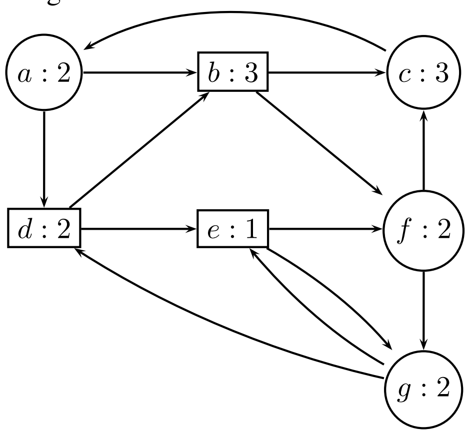
$$\pi'(v) := \begin{cases} \pi(v) + 1 & \text{falls 0 die kleinste Priorität in } \mathcal{G} \text{ ist} \\ \pi(v) - 1 & \text{sonst.} \end{cases}$$

Das duale Spiel entsteht also aus $\mathcal{G}$, indem die Positionen der einzelnen Spieler vertauscht werden und die Prioritäten eines Knotens jeweils für den anderen Spieler „günstig“ gemacht werden.

5.22 Beispiel (Duale Spiele). Das duale Spiel zum Spiel $\mathcal{G}$



aus Beispiel 5.3 ist wie folgt definiert.



Wie man leicht sieht, gilt folgender Zusammenhang zwischen dualen Spielen.

5.23 Lemma. *Für alle Paritätsspiele* $\mathcal{G} := (V, V_0, E, \pi)$ *gilt* $W_\rho(\mathcal{G}) = W_{\bar{\rho}}(\mathcal{G}^d)$.

5.3.2 Abwicklung von Paritätsspielen

Sei $\mathcal{G} := (V, V_0, E, \pi)$ ein endliches Paritätsspiel, d.h. V endlich. Wir nehmen an, daß die kleinste in $\mathcal{G}$ auftretende Priorität gerade ist (ansonsten wechseln wir zum dualen Spiel) und daß alle Knoten 1. einen Nachfolger haben und 2. alle Knoten mit minimaler Priorität genau einen Nachfolger haben. Dies ist keine Einschränkung der Allgemeinheit, da jedes Paritätspiel leicht so transformiert werden kann, daß nur die Knoten mit maximaler Priorität mehr als einen Nachfolger haben.

Sei T die Menge der Knoten in V mit minimaler Priorität.Wir definieren induktiv für alle $\alpha \in \mathbb{N}$ eine Partitionierung T_0^α, T_1^α von T, d.h. $T = T_0^\alpha \dot\cup T_1^\alpha$ und $T_0^\alpha \cap T_1^\alpha = \varnothing$, sowie Mengen W_0^α, W_1^α und ein Spiel $\mathcal{G}^\alpha$ wie folgt.

Für $\alpha = 0$ sei

- $T_0^0 := T$ und
- $\mathcal{G}^0 := (V, V_0 \backslash T, E', \pi)$ mit $E' = E \backslash \{(u, v) \in E : u \in T\}$.

Weiterhin gilt für alle $\alpha \geqslant 0$,

$$W_\rho^\alpha := \{v : \text{Spieler } \rho \text{ gewinnt das Spiel } \mathcal{G}^\alpha \text{ von Position } \alpha \text{ aus } \}.$$

Schließlich definieren wir für $\alpha > 0$

- $T_0^{\alpha+1} := \{v \in T : s_{\mathcal{G}}(v) \in W_0^\alpha\}$. Hierbei bezeichnet $s_{\mathcal{G}}(v)$ den (eindeutig bestimmen) Nachfolger von v im Spiel $\mathcal{G}$.
- $\mathcal{G}^{\alpha+1} := (V, (V_0 \backslash T) \cup (T \backslash T_0^{\alpha+1}), E', \pi)$. E' ist wie oben definiert.

Die Spiele $\mathcal{G}^\alpha$ unterscheiden sich also von $\mathcal{G}$ zunächst einmal dadurch, daß alle von Knoten mit minimaler Priorität ausgehenden Kanten gelöscht werden. Innerhalb der einzelnen $\mathcal{G}^\alpha$ ändert sich dann nur noch, wer an den einzelnen Knoten minimaler Priorität gewinnt.

Aus der Determiniertheit von Paritätsspielen folgt $W_0^\alpha \dot\cup W_1^\alpha = V$. Ferner gilt $W_0^\alpha \supseteq W_0^{\alpha+1}$ sowie $W_1^\alpha \subseteq W_1^{\alpha+1}$ für alle $\alpha \in \mathbb{N}$. Da V endlich ist, existiert ein $\alpha \in \mathbb{N}$ (sogar: $\alpha \leqslant |V|$), so daß $W_0^\alpha = W_0^{\alpha+1} =: W_0^\infty$ und $W_1^\alpha = W_1^{\alpha+1} =: W_1^\alpha$.

5.24 Lemma (Abwicklungslemma). *Sei* $\mathcal{G}$ *wie oben ein endliches Paritätsspiel,* $W_0 := W_0(\mathcal{G})$ *die Gewinnmenge für Spieler* 0 *und* $W_1 := W_1(\mathcal{G})$ *die Gewinnmenge für Spieler* 1. *Dann gilt* $W_0 = W_0^\infty$ *und* $W_1 = W_1^\infty$.

Beweis: Wir definieren Strategien f für Spieler 0 und g für Spieler 1, so daß Spieler ρ von allen $v \in W_\rho^\infty$ gewinnt.

Strategie für Spieler 0**:** Sei f^α eine Gewinnstrategie für Spieler 0 in $\mathcal{G}^\alpha = \mathcal{G}^{\alpha+1}$ für alle $v \in W_0^\alpha$. Da alle Knoten $v \in T$ eindeutige Nachfolger in $\mathcal{G}$ haben, läßt sich f^α kanonisch zu einer Strategie f in $\mathcal{G}$ erweitern.

Behauptung. f ist Gewinnstrategie für Spieler 0 in $\mathcal{G}$ von allen $v \in W_0^\infty$ aus.

Beweis. Sei $\overline{v} := v_0 v_1 v_2 \dots$ eine f-konforme Partie in $\mathcal{G}$ mit $v_0 \in W_0^\infty$. Dann gilt $v_i \in W_0^\infty$ für alle i. Denn ist $v_i \in W_0^\infty \backslash T$, so ist $v_{i+1} \in W_0^\infty$, da f Gewinnstrategie in $\mathcal{G}^\alpha$ ist und die Spiele $\mathcal{G}$ und $\mathcal{G}^\alpha$ auf solchen Knoten übereinstimmen. Ist hingegen $v_i \in W_0^\alpha \cap T = W_0^{\alpha+1} \cap T = W_0^\infty \cap T$, so ist $s_\mathcal{G}(v) \in W_0^\alpha$ nach Konstruktion von W_0^α.

Als nächstes zeigen wir, daß jede Partie, die immer in W_0^α bleibt, von Spieler 0 gewonnen wird.

- Gilt für ein $i \in \mathbb{N}$ und alle $j \geqslant i$ $v_j \notin T$, so ist $v_i v_{i+1} \dots$ eine f-konforme Partie, die immer in dem Teil von $\mathcal{G}^\alpha$ bleibt, der mit $\mathcal{G}$ übereinstimmt. Somit ist die kleinste auf $\overline{v}$ unendlich oft vorkommende Priorität gerade und Spieler 0 gewinnt die Partie auch in $\mathcal{G}$.
- Andernfalls werden in $\overline{v}$ unendlich oft Knoten aus T durchlaufen. Da dies die Knoten mit minimaler und gerader Priorität sind, ist $\overline{v}$ gewinnend für Spieler 0.

Also ist f eine Gewinnstrategie für Spieler 0 von allen $v \in W_0^\infty$ aus.

Für Spieler 1**:** Für jedes $v \in W_1^\infty$ sei $\gamma(v) = \min\{\beta : v \in W_1^\beta\}$. Für jedes $\beta \in \mathbb{N}$ sei g^β eine Gewinnstrategie für Spieler 1 in $\mathcal{G}^\beta$ von jedem $v \in W_1^\beta$ aus. Wir definieren eine neue Strategie g durch

$$g(v) := \begin{cases} g^{\gamma(v)}(v) & \text{für alle } v \in V_1 \backslash T \\ s_\mathcal{G}(v) & \text{für } v \in V_1 \cap T. \end{cases}$$

Sei $\overline{v} := v_0 v_1 v_2 \dots$ eine g-konforme Partie in $\mathcal{G}$.

Behauptung: Ist $v_i \in W_1^\infty$, so

(1) $v_{i+1} \in W_1^\infty$

(2) $\gamma(v_i) \geqslant \gamma(v_{i+1})$ und

(3) $\gamma(v_i) > \gamma(v_{i+1})$ falls $v_i \in T$.

Beweis. Ist $v_i \in W_1^\infty \backslash T$ und $\gamma(v_i) = \beta$, also insbesondere $v_i \in W_1^\beta$, dann ist $v_{i+1} \in W_1^\beta$, da für $v_i \in V_1$ Spieler 1 gemäß g^β spielt und für $v_i \in V_0$ Spieler 0 nicht aus W_1^β herausziehen kann. Es folgt $\gamma(v_{i+1}) \leqslant \gamma(v_i)$.

Ist hingegen $v_i \in W_1^\infty \cap T$ und $\gamma(v_i) = \beta$, also $v_i \in T_1^\beta$ und $\beta \geqslant 1$, so ist nach Konstruktion von W_1^β $v_{i+1} = s_\mathcal{G}(v_i) \in W_1^{\beta-1}$. Also $\gamma(v_{i+1}) < \gamma(v_i)$.

Die Eigenschaften (1) bis (3) implizieren, daß die Partie immer in W_1^∞ bleibt und die Werte $\gamma(v_i)$ nicht größer werden. Außerdem wird $\gamma(v_i)$ echt kleiner, wenn $v_i \in T$. Also können nur endlich viele Knoten aus T durchlaufen werden und es existiert ein $i \in \mathbb{N}$ mit $v_j \notin T$ für alle $j \geqslant i$. Da $\overline{v}$ g-konform ist, ist also die kleinste unendliche oft in $v_i v_{i+1} \dots$ gesehene Priorität ungerade. Spieler 1 gewinnt also $\overline{v}$.

□

5.4 Auswertungsspiele für die kleinste Fixpunktlogik

In diesem Abschnitt werden Paritätsspiele als Auswertungsspiele für die kleinste Fixpunktlogik verwendet. Dazu ist es zweckmäßig, neben den üblichen kleinsten Fixpunktoperatoren auch Operatoren zur Definition größter Fixpunkte zuzulassen. Wir erinnern zunächst noch einmal an die wesentlichen Begriffe und führen dann die entsprechende Erweiterung von LFP ein.

Erinnerung. Sei M eine endliche Menge und $F : M \to M$ monoton. Mit

$$\mathbf{gfp}(F) := \bigcup \{M : F(M) = M\}$$

bezeichnen wir den größten Fixpunkt von F. Für $F^0 := M$ und $F^{\alpha+1} := F(F^\alpha)$ gibt es ein $\alpha \in \mathbb{N}$ mit $F^\alpha = F^{\alpha+1} =: F^\infty$ und $F^\infty = \mathbf{gfp}(F)$. Weiterhin gilt $\mathbf{lfp}(F) = (\mathbf{gfp}(F^d))^c$, wobei F^d die duale Abbildung $F^d(X) = (F(X^c))^c$ bezeichnet.

Wir erweitern die Syntax der kleinsten Fixpunktlogik LFP durch die Regel

(GFP) Ist $\varphi(R, \overline{x}) \in$ LFP eine Formel, R ein Relationssymbol der Stelligkeit k, $|\overline{x}| = k$ und kommt R nur positiv in φ vor, so ist

$$[\mathbf{gfp}_{R,\overline{x}}\, \varphi](\overline{t}) \in \text{LFP}$$

für ein k-Tupel $\overline{t}$ von Termen.

Wegen der Dualität kleinster und größter Fixpunkte gilt für alle Strukturen $\mathfrak{A}$ und alle $\overline{a} \in A^k$

$$(\mathfrak{A}, \overline{a}) \models \neg[\mathbf{lfp}_{R,\overline{x}}\, \varphi](\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \overline{a}) \models [\mathbf{gfp}_{R,\overline{x}}\, \neg\varphi(R\overline{u}/\neg R\overline{u})](\overline{x}).$$

Als Konsequenz kann jede LFP-Formel mit Hilfe der de Morgan'schen Gesetze und dieser Dualitätsregel in Negations-Normalform gebracht werden. Wir nehmen daher für den Rest dieses Kapitels an, daß alle Formeln folgende Eigenschaften erfüllen.

- Alle Formeln sind in NNF.
- Die Fixpunktoperatoren haben keine Parameter.
- Keine Fixpunktvariable wird doppelt definiert, d.h. für keine Variable R kommt mehr als eine Formel der Gestalt $[\mathbf{lfp}_{R,\overline{x}}\, \varphi](\overline{x})$ oder $[\mathbf{gfp}_{R,\overline{x}}\, \varphi](\overline{x})$ vor.

- Wird eine Fixpunktvariable R durch eine Formel der Gestalt $[\mathbf{lfp}_{R,\overline{x}}\,\varphi](\overline{x})$ oder $[\mathbf{gfp}_{R,\overline{x}}\,\varphi](\overline{x})$ definiert, so haben alle Atome, in denen R vorkommt, die Form $R\overline{x}$.

Formeln in dieser Gestalt nennen wir *wohl geformt*.

5.25 Definition. Sei $\varphi \in \text{LFP}$ wohl geformt und R, R' in φ vorkommende Fixpunktvariablen.

(i) Die eindeutig bestimmt Unterformel $[\mathbf{fp}_{R,\overline{x}}\,\varphi'](\overline{t})$ von φ wird die R *definierende Unterformel* genannt und mit $\mathcal{D}_\varphi(R)$ bezeichnet. Hierbei steht **fp** für **lfp** oder **gfp**.

(ii) R *ist abhängig von* R', falls R' frei in $\mathcal{D}_\varphi(R)$ vorkommt. Insbesondere ist $\mathcal{D}_\varphi(R)$ also eine Unterformel von $\mathcal{D}_\varphi(R')$. Wir schreiben $R \sqsubseteq R'$, falls R von R' abhängt.

(iii) Die *Abhängigkeitsordnung* $\sqsubseteq_\varphi$ ist definiert als reflexive, transitive Hülle von $\sqsubseteq_1$.

(iv) Die *Alternierungstiefe ad*$_\varphi(R)$ *von* R ist die maximale Anzahl von Wechseln zwischen kleinsten und größten Fixpunkten auf $\sqsubseteq_\varphi$-Pfaden, die bei R beginnen.

(v) Die *Alternierungstiefe ad*(φ) *von* φ ist die maximale Alternierungstiefe einer Fixpunktvariablen in φ.

5.26 Beispiel. Die Formel

$$\varphi := [\mathbf{lfp}_{R,x}\,\forall y(Exy \rightarrow [\mathbf{gfp}_{Q,z}\,y < z \vee \exists z'(Qz' \wedge Ezz')](x)](x)$$

hat Alternierungstiefe 1. Die Formel

$$\psi := [\mathbf{lfp}_{R,x}\,\forall y(Exy \rightarrow [\mathbf{gfp}_{Q,z}\,Rz \vee \exists z'(Qz' \wedge Ezz')](x)](x)$$

hingegen hat Alternierungstiefe 2.

Mit Hilfe dieser Vorbemerkungen können wir nun zu gegebener Formel $\varphi \in \text{LFP}$ und Struktur $\mathfrak{A}$ das dazu passende Auswertungsspiel $\mathcal{G}(\mathfrak{A}, \psi)$ definieren. Dabei handelt es sich um ein Paritätsspiel. Die Arena des Spiels ist genauso definiert wie für die Logik erster Stufe, nur daß die zusätzlichen Unterformeln der Gestalt $[\mathbf{fp}_{R,\overline{x}}\,\varphi](\overline{x})$ und Atome $R\overline{x}$ mit Fixpunktvariablen behandelt werden müssen. Dabei gibt es eine Kante von $[\mathbf{fp}_{R,\overline{x}}\,\varphi](\overline{x})$ zu φ und von $R\overline{x}$ zu $[\mathbf{fp}_{R,\overline{x}}\,\varphi](\overline{x})$. Offensichtlich ist es egal, welcher Spieler an diesem Positionen zieht, da es keine echte Wahl gibt. Wir geben als nächstes die formale Definition der Spiele und versuchen anschließend, die dahinter stehende Idee zu erläutern.

5.27 Definition. Sei σ eine Signatur, $\psi \in \text{LFP}[\sigma]$ ein Satz in Negations-Normalform und $\mathfrak{A} := (A, \sigma^{\mathfrak{A}})$ eine σ-Struktur.

Das *Auswertungsspiel* $\mathcal{G}(\mathfrak{A}, \psi) := (V, V_0, E, v_0, \pi)$ für $\mathfrak{A}$ und ψ ist ein Paritätsspiel, daß wie folgt definiert wird.

- V ist die Menge aller Paare (β, φ), wobei $\varphi \in \text{cl}(\psi)$ und β : *frei*$(\varphi) \rightarrow A$ eine Belegung der freien Variablen von φ ist.

- V_0 ist die Menge aller Paare $(\beta, \varphi) \in V$, so daß φ die Gestalt $\varphi_1 \vee \varphi_2$ oder $\exists x\varphi'$ oder aber φ ein Literal $P\overline{t}$ oder $\neg P\overline{t}$, mit $P \in \sigma$, ist und $(\mathfrak{A}, \beta) \not\models \varphi$.
- Es gibt eine Kante zwischen zwei Knoten (β, φ) und (β', φ'), falls
  - $\varphi := \varphi_1 \wedge \varphi_2$ oder $\varphi := \varphi_1 \vee \varphi_2$ und $\beta = \beta'$ sowie $\varphi' = \varphi_1$ oder $\varphi' = \varphi_2$.
  - $\varphi := \forall x\varphi_1$ oder $\varphi := \exists x\varphi_1$ und $\varphi' = \varphi_1$ sowie $\beta' = \beta[x/a]$ für ein $a \in A$, d.h.für alle Variablen $y \neq x$ gilt $\beta'(y) = \beta(y)$ und $\beta'(x) = a$.
  - $\varphi := R\overline{x}$ mit $R \notin \sigma$, $\varphi' = D_\psi(R)$ und $\beta = \beta'$
  - $\varphi := [\mathbf{fp}_{R,\overline{x}}\, \varphi_1](\overline{x})$, $\varphi' = \varphi_1$ und $\beta = \beta'$.
- $v_0 := (\varnothing, \psi)$, wobei $\varnothing$ die leere Interpretation ist.
- Für alle Knoten $v = (\beta, \varphi)$ mit $\varphi := R\overline{x}$ für eine Fixpunktvariable R gilt $\pi(v) = \mathit{ad}_\psi(R)$. Alle anderen Knoten bekommen die maximale Priorität, d.h. $\mathit{ad}(\psi)$.

Sei $\psi := [\mathbf{lfp}_{R,\overline{x}}\, \varphi](\overline{x})$, mit $\varphi \in$ FO, $\mathfrak{A}$ eine Struktur und $\overline{a} \in A^k$ eine Belegung der Variablen $\overline{x}$. Versucht die Verifiziererin zu zeigen, daß $\mathfrak{A} \models \psi[\overline{a}]$, d.h. versucht sie das Spiel von der Position $(\overline{x} \mapsto \overline{a}, \psi)$ zu gewinnen, so muß sie im Prinzip zeigen, daß $\overline{a} \in R^\alpha$ für eine Induktionsstufe α der Fixpunktinduktion über φ. Da $\varphi \in$ FO eine Formel der Logik erster Stufe ist, hat sie dazu zwei Möglichkeiten. Entweder sie zwingt das Spiel - wie in Abschnitt 5.2 erläutert - in ein für sie gewinnendes Blatt, d.h. reduziert die Formel auf ein Literal, das in der Struktur gilt, oder aber das Spiel läuft in einen Knoten $(\beta', R\overline{x})$. Von dort aus geht es dann mit der Formel ψ weiter, jetzt aber mit der Belegung β'. Man sagt, daß hier der Fixpunkt *regeneriert* wird. Sei $\beta'(\overline{x}) = \overline{c}$. Das bedeutet, daß die Verifiziererin nun zeigen muß, daß $\overline{c} \in R^\beta$ für ein $\beta < \alpha$. In diesem Falle geht das Spiel wie beschrieben weiter. Da β echt kleiner α sein muß, darf der Verifiziererin nur endlich oft erlaubt werden, durch ein Atom $R\overline{x}$ zu laufen. Irgendwann ist sie bei $\beta = 1$ angekommen und muß hier wirklich den Nachweis führen, daß ein Elementtuple $\overline{d} \in R^1$. Daher erhalten die Fixpunktvariablen, die durch kleinste Fixpunkte definiert werden, ungerade Prioritäten.

Dual dazu erhalten größte Fixpunktoperatoren gerade Prioritäten, da hier der Falsifizierer nachweisen muß, daß ein Elementtupel irgendwann nicht mehr im Fixpunkt auftaucht. Schließlich müssen weiter innen in der Formel auftretende Fixpunkte größere Prioritäten bekommen als die sie umschließenden Fixpunktoperatoren. Dies führt zu oben stehender Definition der Auswertungsspiele. Wir können nun den Nachweis führen, daß das Spiel $\mathcal{G}(\mathfrak{A}, \psi)$ von der Verifiziererin genau dann gewonnen wird, wenn $\mathfrak{A} \models \psi$.

5.28 Theorem. *Sei $\psi(\overline{x})$ eine wohl-geformte* LFP$[\sigma]$*-Formel, $\mathfrak{A} := (A, \sigma)$ eine endliche σ-Struktur und $\beta : \mathit{frei}(\psi) \to A$. Dann gilt $(\mathfrak{A}, \beta) \models \psi$ genau dann, wenn die Verifiziererin eine Gewinnstrategie im Spiel $\mathcal{G}(\mathfrak{A}, \psi)$ vom Knoten (β, ψ) aus hat.*

Beweis: Der Beweis wird per Induktion über den Formelaufbau geführt. Der einzig interessante Fall betrifft Fixpunktformeln der Gestalt $\psi(\overline{x}) := [\mathbf{gfp}_{R,\overline{x}}\, \varphi](\overline{x})$.

In dem Spiel $\mathcal{G}(\mathfrak{A}, \psi)$ sind die Knoten mit minimaler Priorität genau die Fixpunktatome $(\beta', R\overline{x})$. Weiterhin folgt aus der Induktionsannahme, daß für jede Belegung $\beta' : \mathit{frei}(\varphi) \to A$ und jede Interpretation $P \subseteq A^k$ von R die Verifiziererin genau dann eine Gewinnstrategie im Spiel $\mathcal{G}((\mathfrak{A}, P), \varphi)$ vom Knoten (β', φ) aus hat, wenn $((\mathfrak{A}, P), \beta') \models \varphi$. Schließlich gilt, daß $(\mathfrak{A}, \beta) \models [\mathbf{gfp}_{R,\overline{x}}\, \varphi](\overline{x})$ genau dann, wenn für alle $\alpha \in \mathbb{N}$ gilt: $((\mathfrak{A}, R^\alpha), \beta) \models \varphi$.

Per Induktion zeigt man leicht, daß die Spiele $\mathcal{G}((\mathfrak{A}, R^\alpha), \varphi)$ gerade den Spielen $\mathcal{G}^\alpha$ der Abwicklung von $\mathcal{G}(\mathcal{A}, \psi)$ entsprechen. Nach dem Abwicklungslemma folgt somit, daß die Verifiziererin das Spiel $\mathcal{G}(\mathfrak{A}, \psi)$ genau dann gewinnt, wenn sie alle Spiele $\mathcal{G}^\alpha$ gewinnt, also genau dann, wenn $((\mathfrak{A}, R^\alpha), \beta) \models \psi$, also genau dann, wenn $(\mathfrak{A}, \beta) \models \psi$. $\square$

Die Konstruktion des Spiels $\mathcal{G}(\mathfrak{A}, \psi)$ aus gegebenem Satz ψ und Struktur $\mathfrak{A}$ kann offenbar in Polynomialzeit ausgeführt werden. Weiterhin ist die Größe des Spielgraphen beschränkt durch $\mathcal{O}(|\mathrm{cl}(\psi)| \cdot |A|^{\mathit{weite}(\psi)})$.

5.29 Korollar. *Das Auswertungsproblem für* LFP*-Formeln beschränkter Weite ist in* NP $\cap$ CO-NP*, falls keine Parameter in den Fixpunktformeln erlaubt werden.*

Ist weiterhin auch die Alternierungstiefe (nicht die Anzahl) der Fixpunktoperatoren beschränkt, so ist das Problem sogar in PTIME.

5.30 Korollar. *Das Auswertungsproblem für* LFP*-Formeln beschränkter Weite und beschränkter Alternierungstiefe ist in* PTIME*, falls keine Parameter in den Fixpunktformeln erlaubt werden.*

Ohne Beweis geben wir noch an, daß der Gewinner eines Paritätsspiels in M-LFP definiert werden kann.

5.31 Theorem. *Sei $\mathcal{G} := (V, V_0, E, \pi)$ ein Paritätsspiel. Dann hat Spieler 0 eine Gewinnstrategie vom Knoten $v \in V$ aus genau dann, wenn $\mathcal{G}, v \models \varphi_0$, wobei*

$$\mathbf{gfp}_{R_0,x}\, \mathbf{lfp}_{R_1,x} \dots \mathbf{fp}_{R_k,x} \left(\begin{array}{l} \big(V_0x \wedge \bigvee_{i=0}^{k} \pi(x) = i \wedge \exists y(Exy \wedge R_iy)\big) \vee \\ \big(\neg V_0x \wedge \bigvee_{i=0}^{k} \pi(x) = i \wedge \forall y(Exy \to R_iy)\big) \end{array} \right).$$

Hierbei steht $\mathbf{fp}$ *für* $\mathbf{lfp}$*, falls k ungerade ist und ansonsten für* $\mathbf{gfp}$.

Das Theorem zeigt, daß Paritätsspiele *genau* die Auswertungsspiele für kleinste Fixpunktlogiken sind. Denn natürlich könnte man das Auswertungsproblem für LFP auf das Strategieproblem für noch viel allgemeinere, also stärkere, Spiele reduzieren. Das Theorem besagt aber, daß wir auch umgekehrt das Paritätsspielproblem auf das Auswertungsproblem für M-LFP reduzieren können. Die Paritätsspiele haben also gerade die richtige Ausdrucksstärke.