Tafelmitschriften zur Vorlesung „Logik“ im Wintersemester 2019/20

Dr. Jean Christoph Jung, Prof. Dr. Thomas Schneider
AG Theorie der Künstlichen Intelligenz
Fachbereich 3
Universität Bremen

Stand: 15. Dezember 2019

Inhaltsverzeichnis

Teil I.

Aussagenlogik

T1.1 Modellierung des Zeitplanungsproblems

Wir verwenden die Variablen

$$x_{ij}^{\alpha} \quad \text{mit} \quad \alpha \in \{\mathsf{M},\mathsf{S},\mathsf{K}\},\ i \in \{\mathsf{I},\mathsf{II},\mathsf{III}\},\ j \in \{\mathsf{a},\mathsf{b}\}.$$

Die Variable x_{ij}^{α} repräsentiert die Aussage

„Lehrerin α unterrichtet in Stunde i die Klasse j."

Wir modellieren das Problem wie folgt:

- Jede Stunde wird von einer passenden Lehrerin unterrichtet:

  $$\varphi_1 = \bigwedge_{ij \in \{\mathsf{Ia},\mathsf{IIIa},\mathsf{IIIb}\}} (x_{ij}^{\mathsf{M}} \vee x_{ij}^{\mathsf{K}}) \quad \wedge \quad \bigwedge_{ij \in \{\mathsf{Ib},\mathsf{IIa},\mathsf{IIb}\}} (x_{ij}^{\mathsf{S}} \vee x_{ij}^{\mathsf{K}})$$

  Die Konjunktion in der 1. Hälfte iteriert über alle Deutschstunden (also $\mathsf{Ia},\mathsf{IIIa},\mathsf{IIIb}$) und besagt, dass diese nur von Lehrerinnen Müller oder Körner unterrichtet werden können. Analog für die 2. Hälfte.

- Jede Lehrerin unterrichtet ≥ 2 Stunden:

  $$\varphi_2 = \bigwedge_{\alpha \in \{\mathsf{M},\mathsf{S},\mathsf{K}\}} \bigvee_{\substack{ij,i'j' \in \{\mathsf{Ia},\mathsf{Ib},\mathsf{IIa},\mathsf{IIb},\mathsf{IIIa},\mathsf{IIIb}\} \\ ij \neq i'j'}} (x_{ij}^{\alpha} \wedge x_{i'j'}^{\alpha})$$

  Diese Formel ist eine Konjunktion aus drei Teilen, einem pro Lehrerin α. Jeder Teil ist wiederum eine Disjunktion, die besagt, dass es ein Paar von *verschiedenen* Stunden (ij und $i'j'$) gibt, das die jeweilige Lehrerin α unterrichtet.

- Keine Lehrerin unterrichtet zwei Klassen gleichzeitig:

  $$\varphi_3 = \bigwedge_{\alpha \in \{\mathsf{M},\mathsf{S},\mathsf{K}\}} \bigwedge_{i \in \{\mathsf{I},\mathsf{II},\mathsf{III}\}} \neg(x_{i\mathsf{a}}^{\alpha} \wedge x_{i\mathsf{b}}^{\alpha})$$

  Diese Formel ist eine Konjunktion über alle Kombinationen aus Lehrerin α und Stunde i; jeder Bestandteil dieser Konjunktion besagt, dass Lehrerin α in Stunde i nicht gleichzeitig Klasse a und b unterrichten kann.

- In jeder Unterrichtsstunde wird jede Klasse von höchstens einer Lehrerin unterrichtet:

  $$\varphi_4 = \bigwedge_{i \in \{\mathsf{I},\mathsf{II},\mathsf{III}\}} \bigwedge_{j \in \{\mathsf{a},\mathsf{b}\}} \Big(\neg(x_{ij}^{\mathsf{M}} \wedge x_{ij}^{\mathsf{S}}) \quad \wedge \quad \neg(x_{ij}^{\mathsf{M}} \wedge x_{ij}^{\mathsf{K}}) \quad \wedge \quad \neg(x_{ij}^{\mathsf{S}} \wedge x_{ij}^{\mathsf{K}})\Big)$$

  Diese Formel ist eine Konjunktion über alle Kombinationen aus Stunde i und Klasse j; jeder Bestandteil dieser Konjunktion besagt, dass keine zwei Lehrerinnen aus $\{\mathsf{M},\mathsf{S},\mathsf{K}\}$ in ij unterrichten können.

Man sieht nun leicht, dass die Belegungen V mit

$$V \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$$

genau den Lösungen des Zeitplanungsproblems entsprechen.

T1.2 Beispiel fürs Ersetzungslemma

Seien $\varphi = x \wedge y$ and $\psi = \neg(\neg x \vee \neg y)$,
und sei $\vartheta = z \vee \neg(\underbrace{x \wedge y}_{\varphi})$.

Dann ist $\vartheta' = z \vee \neg \underbrace{\neg(\neg x \vee \neg y)}_{\psi}$.

Wir wissen aus dem vorhergenden Beispiel, dass $\varphi \equiv \psi$.

Also liefert das Ersetzungslemma $\vartheta \equiv \vartheta'$.

T1.3 Beweis des Ersetzungslemmas

Lemma 1.8 (Ersetzungslemma). Seien φ and ψ äquivalente Formeln, ϑ eine Formel mit $\varphi \in \mathsf{TF}(\vartheta)$ und ϑ' eine Formel, die sich aus ϑ ergibt, indem ein beliebiges Vorkommen von φ durch ψ ersetzt wird. Dann gilt $\vartheta \equiv \vartheta'$.

Beweis.

Induktionsanfang.

Wenn ϑ atomar ist, muss $\vartheta = \varphi$ sein. Dann ist $\vartheta' = \psi$, also $\vartheta \equiv \vartheta'$.

Induktionsschritt.

Wenn $\vartheta = \varphi$, dann können wir wie im Induktionsanfang argumentieren.

Anderenfalls unterscheiden wir drei Fälle.

1. $\vartheta = \neg\vartheta_1$.

   Dann hat ϑ' die Form $\neg\vartheta_1'$, wobei man ϑ_1' aus ϑ_1 erhält, indem man ein Vorkommen von φ durch ψ ersetzt. Nach Induktionsvoraussetzung (IV) gilt $\vartheta_1 \equiv \vartheta_1'$. Mit der Semantik der Negation folgt daraus $\vartheta \equiv \vartheta'$.

2. $\vartheta = \vartheta_1 \vee \vartheta_2$.

   Dann wird φ entweder in ϑ_1 oder in ϑ_2 durch ψ ersetzt. Wir betrachten nur den ersten Fall: Dann ist $\vartheta' = \vartheta_1' \vee \vartheta_2$, mit ϑ_1' wie in 1. Nach IV gilt $\vartheta_1 \equiv \vartheta_1'$. Mit der Semantik der Disjunktion folgt daraus $\vartheta \equiv \vartheta'$.

3. $\vartheta = \vartheta_1 \wedge \vartheta_2$.

   Analog zu 2. ❑

T1.4 Beispiel für die Wandlung Formel → Boolesche Funktion

Sei $\varphi = (x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)$ mit $|\mathsf{Var}(\varphi)| = 2$.
Aus der Verknüpfungstafel kann man direkt $f_\varphi(x_1, x_2)$ ablesen:

$V(x_1)$	$V(x_2)$	$V(\varphi)$		
0	0	1	d. h.	$f_\varphi(0,0) = 1$
0	1	0		$f_\varphi(0,1) = 0$
1	0	0		$f_\varphi(1,0) = 0$
1	1	1		$f_\varphi(1,1) = 1$

T1.5 Beweis des Theorems über funktionale Vollständigkeit

Theorem 1.10 (funktionale Vollständigkeit). Zu jeder Booleschen Funktion $f \in \mathcal{B}$ gibt es eine Formel φ mit $f_\varphi = f$.

Beweis. Wenn $f \in \mathcal{B}^0$, dann wird f durch die Formeln 0 oder 1 dargestellt (die Booleschen Konstanten sind ja auch Formeln).
Für $f \in \mathcal{B}^n$ mit $n > 0$ argumentieren wir wie folgt. Jede Eingabe für f hat die Form $t = (w_1, \ldots, w_n) \in \{0,1\}^n$ und kann als Formel

$$\psi_t = \ell_1 \wedge \cdots \wedge \ell_n$$

dargestellt werden, wobei

$$\ell_i = \begin{cases} x_i & \text{wenn } w_i = 1 \\ \neg x_i & \text{wenn } w_i = 0. \end{cases}$$

Aus $t = (0,1,0,1)$ beispielsweise wird:

$$\psi_t = \neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4$$

Setzt man nun

$$\varphi := \bigvee_{\substack{t=(w_1,\ldots,w_n)\in\{0,1\}^n \\ f(t)=1 \text{ (!)}}} \psi_t,$$

so gilt $f_\varphi = f$. ❑

Ist also z. B. $n = 2$ und die Funktion f mit

w_1	w_2	$f(w_1, w_2)$	
0	0	0	
0	1	1	◀
1	0	1	◀
1	1	0	

gegeben, dann ist $\varphi = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$.

T1.6 Beweis des Theorems über KNF-/DNF-Umwandlung

Theorem 1.12 (KNF-/DNF-Umwandlung). Jede Formel lässt sich effektiv in eine äquivalente Formel in KNF und DNF wandeln.

Beweis. Sei φ eine Formel. Um eine äquivalente Formel in **DNF** zu erhalten, wende die Konstruktion aus dem Beweis des letzten Satzes auf f_φ an. Diese Konstruktion ist offensichtlich effektiv.

Aus der effektiven Konstruierbarkeit der DNF folgt auch die der **KNF:**

$$\begin{aligned}
\varphi &\equiv \neg\neg\varphi && \text{(bekannte Äquivalenz: Doppelnegation)} \\
&\equiv \neg \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \ell_{ij} && \text{(Wandeln von } \neg\varphi \text{ in DNF; Ersetzungslemma)} \\
&\equiv \bigwedge_{i=1}^{n} \neg \bigwedge_{j=1}^{m_i} \ell_{ij} && \text{(De Morgan, Ersetzungslemma)} \\
&\equiv \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} \neg\ell_{ij} && \text{(De Morgan, Ersetzungslemma)}
\end{aligned}$$

Die entstandene Formel ist in KNF. ❑

T1.x Existenz kleiner Formeln

Im Wintersemester 2016/17 tauchte die Frage auf, ob jede Boolesche Funktion durch eine (wie auch immer geartete) polynomiell große aussagenlogische Formeln dargestellt werden kann. Diese Frage ist zu *verneinen,* denn:

> Es gibt eine eine Familie von Booleschen Funktionen $f_1, f_2, f_3, \ldots$ mit $f_n \in \mathcal{B}^n$ für alle $n \geq 0$, so dass für alle $n \geq 0$ und alle Formeln φ mit $f_n = f_\varphi$ gilt:
>
> $$|\varphi| > 2^{\mathcal{O}(n)}$$

Um dies zu begründen, benötigt man zwei Überlegungen:

1. Jede aussagenlogische Formel φ ist als Schaltkreis C_φ mit $\neg$-, $\wedge$- und $\vee$-Gattern darstellbar, so dass $|C_\varphi|$ nur polynomiell größer als $|\varphi|$ ist.
2. Nach dem Theorem von Shannon (1949) gibt es eine Familie von Booleschen Funktionen $f_1, f_2, f_3, \ldots$ mit $f_n \in \mathcal{B}^n$ für alle $n \geq 0$, so dass jeder Schaltkreis, der f_n berechnet, eine Größe $> \frac{2^n}{10n}$ hat.

Letzteres ist ein wichtiges Resultat in der Schaltkreiskomplexität. Es wird durch ein Abzählargument bewiesen: Wir wissen bereits, dass es 2^{2^n} Boolesche Funktionen der Stelligkeit n gibt. Dann bleibt zu zeigen, dass es $< 2^{2^n}$ Schaltkreise der Größe $\frac{2^n}{10n}$ gibt. Dies geschieht im Wesentlichen dadurch, dass man Schaltkreise als Zeichenketten

über einem festen Alphabet kodiert und dann kombinatorisch die Anzahl der möglichen Zeichenketten einschränkt. Mit komplexeren Abzählargumenten kann man sogar noch schärfere Schranken beweisen. Details zum Theorem von Shannon sowie die Definition von Schaltkreisen sind nachzulesen in einem dedizierten Kapitel von [AB09].

T1.7 Beispiel für n-ären Junktor

Betrachte $f \in \mathcal{B}^3$ wie folgt:

$$f(0, x, y) = x \vee y$$
$$f(1, x, y) = x \wedge y$$

Die Verknüpfungstafel liefert die Semantik für einen dreistelligen Junktor f:

x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	1
	$\vdots$		
1	1	1	1

Eine Formel über der Junktormenge $\{\neg, \wedge, \vee, f\}$ wäre z. B.:

$$x \vee \neg f(x \vee y,\ f(y, y, z),\ \neg z)$$

T1.8 Funktionale Vollständigkeit des Junktors Nand

Da wir bereits gezeigt haben, dass die Menge $\{\neg, \wedge\}$ funktional vollständig ist, genügt es zu zeigen:

- Mit $\mid$ kann man $\neg$ ausdrücken:

$$\begin{aligned} \varphi \mid \varphi &\equiv \neg(\varphi \wedge \varphi) && \text{(Definition von } \mid) \\ &\equiv \neg\varphi && \text{(Idempotenz)} \end{aligned}$$

- Mit $\mid$ kann man $\wedge$ ausdrücken:

$$\begin{aligned} (\varphi \mid \psi) \mid (\varphi \mid \psi) &\equiv \neg(\varphi \mid \psi) && \text{(gerade gezeigt)} \\ &\equiv \neg\neg(\varphi \wedge \psi) && \text{(Definition von } \mid) \\ &\equiv \varphi \wedge \psi && \text{(Doppelnegation)} \end{aligned}$$

T1.9 (co)NP-Vollständigkeit von Erfüllbarkeit bzw. Gültigkeit

Theorem 1.17 (Komplexität). Das Erfüllbarkeitsproblem der Aussagenlogik ist NP-vollständig. Dies gilt auch für Formeln in KNF, sogar bei max. 3 Literalen pro Konjunkt. Das Gültigkeitsproblem der Aussagenlogik ist co-NP-vollständig. Dies gilt auch für Formeln in DNF, sogar bei max. 3 Literalen pro Disjunkt.

Beweis.

Erfüllbarkeit ist in NP. Sei φ die Eingabeformel mit $|\mathsf{Var}(\varphi)| = n$. Eine Turingmaschine kann mit n *nichtdeterministischen* Übergängen eine Belegung V für φ aufs Band schreiben (also diese Belegung „raten“). Die weitere Berechnung prüft deterministisch, ob $V \models \varphi$ gilt (geht in Linearzeit!); sie ist erfolgreich, wenn dies der Fall ist. Dann gilt:

$$\begin{array}{lll} M \text{ akzeptiert } \varphi & \text{gdw.} & \text{es erfolgreiche Berechnung von } M \text{ auf } \varphi \text{ gibt} \\ & \text{gdw.} & \text{es Belegung } V \text{ gibt mit } V \models \varphi \\ & \text{gdw.} & \varphi \text{ erfüllbar} \end{array}$$

Erfüllbarkeit ist NP-hart. Dies folgt aus dem Theorem von Cook & Levin (siehe Skript „Theoretische Informatik 2“ [Sch19] oder die Literaturverweise dort). Der Beweis verwendet nur Formeln in KNF. Durch eine geeignete Reduktion kann man auch zeigen, dass 3SAT (d. h. Erfüllbarkeit der Einschränkung auf KNFs mit genau 3 Literalen pro Konjunkt) bereits NP-hart ist; siehe ebenfalls Theorie 2. Im Gegensatz dazu ist 2SAT in Polyzeit lösbar!

Gültigkeit ist coNP-vollständig. Die Dualität von Erfüllbarkeit und Gültigkeit (vorhergehendes Lemma) liefert eine Polynomialzeitreduktion des Gültigkeitsproblems auf das Erfüllbarkeitsproblem bzw. zurück. Daraus folgen dann coNP-Zugehörigkeit bzw. -Härte des Gültigkeitsproblems. ❏

T1.10 Beispiele für den Horn-Erfüllbarkeitsalgorithmus

Beispiel 1. Betrachte die Horn-Formel auf der vorhergehenden Folie (mit den Aussagenvariablen Regen, Schnee, Niederschlag, Temp$\leq$0, Temp$<$0).

Dann berechnet der Algorithmus

- in Zeile 1: $V = \{\mathsf{Regen}, \mathsf{Schnee}\}$
- in Zeilen 2–4: $V = \{\mathsf{Regen}, \mathsf{Schnee}\} \cup \{\mathsf{Temp}{\leq}0, \mathsf{Temp}{<}0\}$

Wegen des Konjunkts $\mathsf{Temp}{\leq}0 \wedge \mathsf{Temp}{<}0 \to 0$ wird in Zeile 6 „unerfüllbar“ zurückgegeben.

Beispiel 2. Sei $\varphi = x \wedge y \wedge (x \to z) \wedge (y \wedge z \to u) \wedge (u \wedge w \to 0)$. Dann berechnet der Algorithmus

- in Zeile 1: $V = \{x, y\}$
- in Zeilen 2–4: $V = \{x, y\} \cup \{z, u\}$ (2 Iterationen!)

Da der einzige Constraint in φ das Konjunkt $u \wedge w \to 0$ ist und nicht gleichzeitig u und w in V sind, wird in Zeile 8 „erfüllbar“ zurückgegeben.

T1.11 Korrektheit und Zeitbedarf des Horn-Algorithmus

Lemma 1.23. Der Algorithmus für Erfüllbarkeit von Horn-Formeln ist korrekt und läuft in polynomieller Zeit.

Beweis.

Zeitbedarf. Offensichtlich terminiert der Algorithmus auf jeder Eingabe φ nach maximal $|\mathsf{Var}(\varphi)|$ Durchläufen der while-Schleife, also in polynomieller Zeit.

Korrektheit. Wir müssen zeigen: der Algorithmus antwortet bei Eingabe φ „erfüllbar“ gdw. φ erfüllbar ist.

($\Rightarrow$) Angenommen, der Algorithmus antwortet bei Eingabe φ „erfüllbar“. Sei V die dabei berechnete Menge, betrachtet als Belegung. Wir zeigen: $V \models \varphi$. Dazu genügt es zu zeigen, dass für jedes Konjunkt C von φ gilt: $V \models C$. Sei also C ein Konjunkt von φ. Wir unterscheiden drei Fälle.

1. $C = x$ (Fakt).

   Dann ist $x \in V$ wegen Zeile 1 des Algorithmus, also $V \models x$.

2. $C = x_1 \wedge \cdots \wedge x_k \to x$ (Regel).

   Wenn $\{x_1, \ldots, x_k\} \not\subseteq V$, dann $V \models C$. Anderenfalls muss $x \in V$ sein (wegen Zeilen 2–4) und damit ebenfalls $V \models C$.

3. $C = x_1 \wedge \cdots \wedge x_k \to 0$ (Constraint).

   Da der Algorithmus „erfüllbar“ antwortet, muss wegen Zeile 5 $\{x_1, \ldots, x_k\} \not\subseteq V$ gelten, also $V \models C$.

($\Leftarrow$) Angenommen φ sei erfüllbar. Man zeigt leicht per Induktion über die Anzahl der Schleifendurchläufe in Zeilen 2–4:

$$V \subseteq \widehat{V} \quad \text{für alle Modelle } \widehat{V} \text{ von } \varphi \qquad (*)$$

Um nun zu zeigen, dass der Algorithmus „erfüllbar“ antwortet, betrachten wir ein beliebiges Konjunkt $x_1 \wedge \cdots \wedge x_k \to 0$ von φ. Für dieses müssen wir zeigen: $\{x_1, \ldots, x_k\} \not\subseteq V$. Angenommen $\{x_1, \ldots, x_k\} \subseteq V$. Da φ nach Voraussetzung erfüllbar ist, gibt es ein Modell $\widehat{V}$ von φ. Mit $(*)$ gilt $\widehat{V} \models x_1 \wedge \cdots \wedge x_k$, also $\widehat{V} \not\models x_1 \wedge \cdots \wedge x_k \to 0$, was im Widerspruch zu $\widehat{V} \models \varphi$ steht. ❑

T1.12 Nicht-Horn-Ausdrückbarkeit der Disjunktion

Lemma 1.25 (Nicht-Horn-Ausdrückbarkeit). Keine Horn-Formel ist äquivalent zu $x \vee y$.

Beweis. Angenommen, es gäbe eine zu $x \vee y$ äquivalente Horn-Formel φ. Dann hätte φ und damit auch $x \vee y$ ein minimales Modell V. Betrachte nun folgende Modelle von $x \vee y$:

$$V_x = \{x\} \qquad V_y = \{y\}$$

Offenbar sind beides Modelle von $x \vee y$. Für das minimale Modell V müsste dann gelten: $V \subseteq V_x \cap V_y$, wegen Bedingung 2 für minimale Modelle. Also $V = \emptyset$, was aber kein Modell von $x \vee y$ ist. Damit haben wir einen Widerspruch zur Annahme gefolgert; diese muss also falsch sein. ❏

T1.13 Beweis Resolutionslemma

Lemma 1.28 (Resolutionslemma). Sei M eine Klauselmenge, $C_1, C_2 \in M$ und C die Resolvente von C_1 und C_2. Dann gilt: $M \equiv M \cup \{C\}$

Beweis. Es ist zu zeigen, dass für alle Belegungen V gilt:

$$V \models M \qquad \text{gdw.} \qquad V \models M \cup \{C\} \tag{$*$}$$

Die Richtung „$\Leftarrow$“ von $(*)$ ist dabei trivial, denn wenn V alle Formeln in $M \cup \{C\}$ erfüllt, dann auch alle Formeln in M (Teilmenge von $M \cup \{C\}$!).

Für die Richtung „$\Rightarrow$“ nehmen wir an, es gelte $V \models M$. Da C die Resolvente von $C_1, C_2 \in M$ ist, gilt

$$C \;=\; (C_1 \setminus \{\ell\}) \;\cup\; (C_2 \setminus \{\overline{\ell}\}).$$

Wir unterscheiden zwei Fälle.

1. $V(\ell) = 1$.

   Wegen $V \models C_2$ gilt dann auch $V \models C_2 \setminus \{\overline{\ell}\}$ (denn das „negative“ Disjunkt $\overline{\ell}$ kann nichts zur „positiven“ Disjunktion C_2 beigetragen haben); also $V \models C$.

2. $V(\ell) = 0$.

   Wegen $V \models C_1$ gilt dann auch $V \models C_1 \setminus \{\ell\}$ (wie oben); also $V \models C$. ❏

T1.14 Beispiel für wiederholte Resolventenbildung

Sei $\varphi = x_1 \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \neg x_3$.
Dann ist $M = M(\varphi) = \{\, \{x_1\},\, \{\neg x_1, x_2\},\, \{\neg x_2, x_3\},\, \{\neg x_3\} \,\}$ und somit $\mathsf{Res}^0(M) = M$. Wenn wir jeweils systematisch alle Paare von bisher erzeugten Klauseln durchgehen, so erhalten wir:

$$\begin{aligned}
\mathsf{Res}^1(M) &= \mathsf{Res}^0(M) \cup \{\, \{x_2\},\, \{\neg x_1, x_3\},\, \{\neg x_2\} \,\} \\
\mathsf{Res}^2(M) &= \mathsf{Res}^1(M) \cup \{\, \{x_3\},\, \{\neg x_1\},\, \square \,\} \\
\mathsf{Res}^3(M) &= \mathsf{Res}^2(M)
\end{aligned}$$

Also ist:

$$\begin{aligned}
\mathsf{Res}^*(M) &= \mathsf{Res}^3(M) \\
&= \{\{x_1\}, \{\neg x_1, x_2\}, \{\neg x_2, x_3\}, \{\neg x_3\}, \{x_2\}, \{\neg x_1, x_3\}, \{\neg x_2\}, \{x_3\}, \{\neg x_1\}, \square\}
\end{aligned}$$

T1.15 Beweis Resolutionssatz, Korrektheit

Behauptung:
Wenn M eine endliche Klauselmenge ist und $\square \in \mathsf{Res}^*(M)$, dann ist M unerfüllbar.

Beweis. Da $\square \in \mathsf{Res}^*(M)$, ist $\mathsf{Res}^*(M)$ unerfüllbar. Es genügt zu zeigen: $\mathsf{Res}^*(M) \equiv M$
Mittels Resolutionslemma (Lemma 1.28) und per Induktion über i zeigt man leicht:

$$\mathsf{Res}^i(M) \equiv M \quad \text{für alle } i \geq 0$$

Wegen Lemma 1.30 (2) ist $\mathsf{Res}^*(M) = \mathsf{Res}^i(M)$ für ein $i \geq 0$. Also $\mathsf{Res}^*(M) \equiv M$. ❏

T1.16 Beweis Resolutionssatz, Vollständigkeit

Lemma 1.32. Wenn eine endliche Klauselmenge M *un*erfüllbar ist, dann gilt:

(1) M^+ und M^- sind *un*erfüllbar;
(2) $\square \in \mathsf{Res}^*(M)$ oder $\{\neg x\} \in \mathsf{Res}^*(M)$;
(3) $\square \in \mathsf{Res}^*(M)$ oder $\{x\} \in \mathsf{Res}^*(M)$;
(4) $\square \in \mathsf{Res}^*(M)$.

Beweis.

(1) Angenommen, M^+ sei erfüllbar und $V \models M^+$. Erweitere V durch $V(x) = 1$. Man prüft leicht, dass $V \models M$ (klauselweise, Fallunterscheidung ob $x \in C$ und Anwendung der Definition von M^+). Dies widerspricht der Annahme, dass M unerfüllbar ist. Also ist M^+ unerfüllbar.

Die Unerfüllbarkeit von M^- wird analog begründet.

(2)–(4) Wir gehen per Induktion über $|\mathsf{Var}(M)|$ vor (denn jeder Resolutionsschritt entspricht dem Eliminieren einer Variable).

Induktionsanfang.

Wenn $|\mathsf{Var}(M)| = 0$, dann $M = \emptyset$ oder $M = \{\square\}$. Der Fall $M = \emptyset$ wird dadurch ausgeschlossen, dass $\emptyset$ erfüllbar ist,[1] was der Voraussetzung widerspricht. Also $M = \{\square\}$ und damit $\square \in \mathsf{Res}^*(M)$, wodurch (2)–(4) gezeigt sind.

Induktionsschritt.

(2) Wenn $|\mathsf{Var}(M)| = n+1$, dann $|\mathsf{Var}(M^+)| = n$, denn x wurde gelöscht. Also kann man auf M^+ die Induktionsvoraussetzung anwenden: weil wegen (1) M^+ unerfüllbar ist, gilt $\square \in \mathsf{Res}^*(M^+)$. Also gibt es Klauseln $C_1, \ldots, C_m$, so dass $C_m = \square$ und für alle $i \leq m$ gilt:

[1] Zur Erinnerung: $V \models M$, wenn $V \models C$ für alle $C \in M$. Wenn also $M = \emptyset$, dann ist *jede* Belegung V Modell von M. Also ist $M = \emptyset$ erfüllbar.

(a) $C_i \in M^+$ oder

(b) C_i ist Resolvente von C_j und C_k für gewisse $j, k < i$.

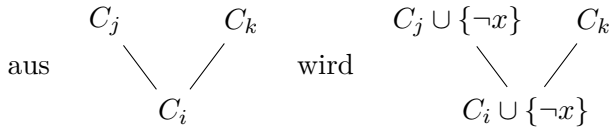
Nun können zwei Fälle eintreten:

(i) Alle Klauseln C_i der Form (a) sind auch in M (d. h. in keiner der „Originalklauseln“ in M kam $\neg x$ vor). Dann prüft man leicht, dass $C_1, \ldots, C_m \in \mathsf{Res}^*(M)$, also insbesondere $\square = C_m \in \mathsf{Res}^*(M)$.

(ii) Für mindestens eine Klausel C_i der Form (a) ist $C_i \cup \{\neg x\} \in M$. Wir erhalten durch Wiedereinfügen von $\neg x$ eine Folge von Klauseln

$$C'_1, \ldots, C'_m \in \mathsf{Res}^*(M),$$

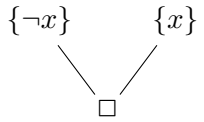
die bezeugt, dass $\{\neg x\} \in \mathsf{Res}^*(M)$:



Wenn also das Literal $\neg x$ an einer Stelle eingeführt wird, dann wird es auch an die jeweilige Resolvente weitergegeben. Am Ende wird aus $C_m = \square$ dann $C'_m = \{\neg x\}$, also $\{\neg x\} \in \mathsf{Res}^*(M)$.

(3) Wird analog zu (2) bewiesen, unter Verwendung von M^- statt M^+.

(4) Aus (2) und (3) folgt, dass $\square \in \mathsf{Res}^*(M)$ oder $\{x\}, \{\neg x\} \in \mathsf{Res}^*(M)$. Aus letzterem folgt aber mit



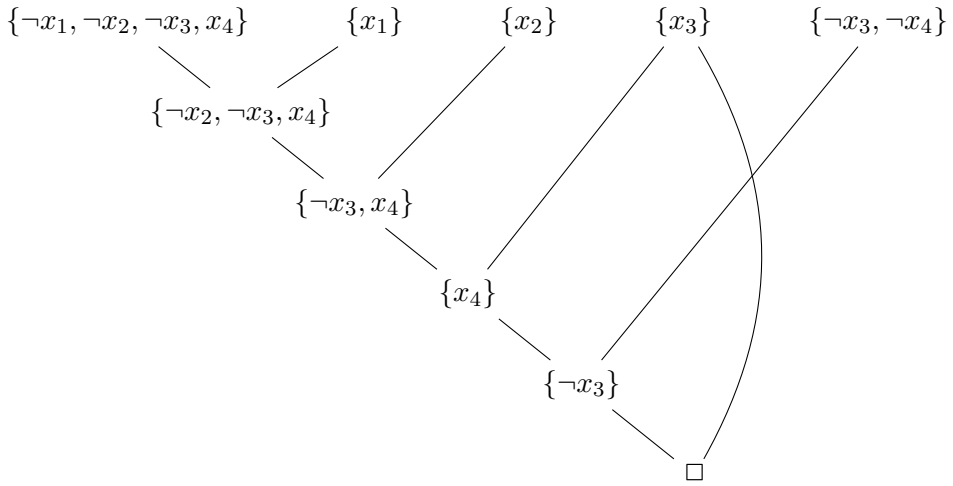
auch $\square \in \mathsf{Res}^*(M)$. ❏

T1.17 Beispiel Einheitsresolution

Gegeben sei die Klauselmenge

$$M = \{\{\neg x_1, \neg x_2, \neg x_3, x_4\}, \{x_1\}, \{x_2\}, \{x_3\}, \{\neg x_3, \neg x_4\}\}.$$

Ein möglicher Resolutionsbeweis mittels Einheitsresolution ist folgender.



Natürlich gibt es auch hier wieder mehrere Resolutionsbeweise; entsprechend enthält $\mathsf{ERes}^*(M)$ mehr Klauseln als die oben gezeigten. Um $\mathsf{ERes}^*(M)$ zu bestimmen, gehe vor wie in T1.14, aber wende *nur Einheitsresolution* an.

T1.18 Beweis Resolutionssatz für Einheitsresolution

Theorem 1.37 (Resolutionssatz für Einheitsresolution).
Eine endliche Menge M von Hornklauseln ist unerfüllbar gdw. $\square \in \mathsf{ERes}^*(M)$.

Beweis.

Richtung „⇐“. (Korrektheit)

Wie im Resolutionssatz (T1.15), denn man zeigt genauso: $\mathsf{ERes}^*(M) \equiv M$.

Richtung „⇒“. (Vollständigkeit)

Wir verwenden die Korrektheit unseres ursprünglichen Algorithmus für Erfüllbarkeit von Horn-Formeln (Folie 43). Setze in Analogie zu diesem Algorithmus:

$$V^0 = \{x \mid M \text{ enthält } \{x\}\}$$
$$V^{i+1} = V^i \cup \{x \mid \text{es gibt } x_1, \ldots, x_k \in V_i \text{ mit } \{\neg x_1, \ldots, \neg x_k, x\} \in M\}$$
$$V^* = \bigcup_{i \geq 0} V^i$$

Mit der Korrektheit des ursprünglichen Algorithmus gilt:

$$M \text{ ist unerfüllbar} \quad \text{gdw.} \quad \text{es } x_1, \ldots, x_k \in V^* \text{ gibt mit } \{\neg x_1, \ldots, \neg x_k\} \in M \quad (*)$$

Wir zeigen zunächst, dass für alle $i \geq 0$ gilt:

$$x \in V^i \quad \text{impliziert} \quad \{x\} \in \mathsf{ERes}^*(M) \qquad (**)$$

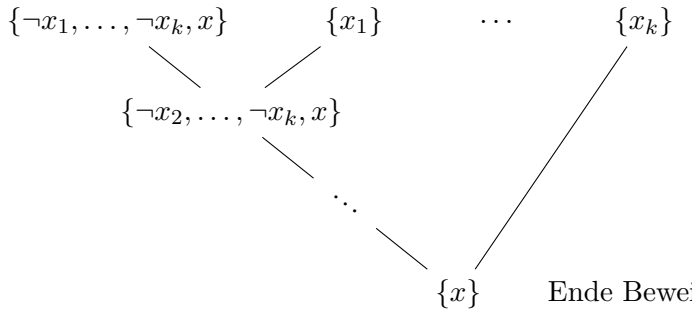
Dazu gehen wir per Induktion über i vor.

Induktionsanfang.

Wenn $x \in V^0$, dann $\{x\} \in M$ nach Definition von V^0. Da $M \subseteq \mathsf{ERes}^*(M)$, gilt auch $\{x\} \in \mathsf{ERes}^*(M)$.

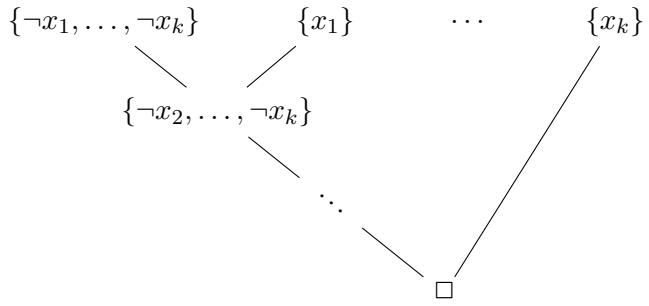
Induktionsschritt.

Sei $x \in V^{i+1}$. Wenn $x \in V^i$, dann folgt die Behauptung bereits nach Induktionsvoraussetzung. Wenn $x \notin V^i$, dann gibt es nach Definition von V^{i+1} Variablen $x_1, \ldots, x_k \in V^i$ mit $\{\neg x_1, \ldots, \neg x_k, x\} \in M$. Nach Induktionsvoraussetzung sind $\{x_1\}, \ldots, \{x_k\} \in \mathsf{ERes}^*(M)$. Nun bezeugt die folgende Ableitung von $\{x\}$ mittels Einheitsresolution, dass $\{x\} \in \mathsf{ERes}^*(M)$.



Ende Beweis von $(**)$.

Nun können wir mit $(*)$ und $(**)$ die gewünschte Implikation beweisen. Sei also M unerfüllbar. Mit $(*)$ und $(**)$ gibt es $\{\neg x_1, \ldots \neg x_k\} \in M$ mit $\{x_1\}, \ldots, \{x_k\} \in \mathsf{ERes}^*(M)$. Folgende Ableitung mittels Einheitsresolution zeigt $\square \in \mathsf{ERes}^*(M)$.



❑

T1.19 Beispiel DPLL-Algorithmus

Sei $M = \{\{x_1, \neg x_2, x_3\}, \{\neg x_1, x_2, \neg x_3\}, \{\neg x_1, x_3, \neg x_4\}, \{\neg x_1, x_3\}\}$.

Aufruf `DPLL`(M)

- *Einheitsklauseln vorhanden?* Nein.
- $\square$ *vorhanden?* Nein.
- *Pure Literale vorhanden?* Ja, und zwar $\neg x_4$ (und kein weiteres).
  Also lösche die vorletzte Klausel, d. h.
  $$M = \{\{x_1, \neg x_2, x_3\}, \{\neg x_1, x_2, \neg x_3\}, \{\neg x_1, x_3\}\}.$$
- $M = \emptyset$ *?* Nein.
- *Nichtdeterministische Verzweigung*
  Wähle $\ell = x_1$ (d. h. „probiere“ $V(x_1) = 1$).

  Aufruf `DPLL`$(\{\{x_1, \neg x_2, x_3\}, \{\neg x_1, x_2, \neg x_3\}, \{\neg x_1, x_3\}, \{x_1\}\})$
  - *Einheitsklauseln vorhanden?* Ja, und zwar $\{x_1\}$.
    - *Unit Subsumption:* Lösche alle Klauseln, die x_1 enthalten.
      $$M = \{\{\neg x_1, x_2, \neg x_3\}, \{\neg x_1, x_3\}\}$$
    - *Unit Resolution:* Lösche $\neg x_1$ aus den übrigen Klauseln.
      $$M = \{\{x_2, \neg x_3\}, \{x_3\}\}$$
  - *Einheitsklauseln vorhanden?* Ja, und zwar $\{x_3\}$.
    - *Unit Subsumption:* Lösche alle Klauseln, die x_3 enthalten.
    - *Unit Resolution:* Lösche $\neg x_3$ aus den übrigen Klauseln.
      $$M = \{\{x_2\}\}$$
  - *Einheitsklauseln vorhanden?* Ja, und zwar $\{x_2\}$.
    - *Unit Subsumption:* Lösche alle Klauseln, die x_2 enthalten.
      $$M = \emptyset$$
    - *Unit Resolution:* Keine Veränderung, da M leer ist.
  - $\square$ *vorhanden?* Nein.
  - *Pure Literale vorhanden?* Nein.
  - $M = \emptyset$ *?* Ja $\Rightarrow$ **Ausgabe „erfüllbar“**

  $\Rightarrow$ **Ausgabe „erfüllbar“**

T1.20 Beispiel Hilbert-Kalkül

Die Formel $x \to x$ ist herleitbar:

(a)	$x \to ((y \to x) \to x)$	Instanz von Axiom 1 mit $\varphi = x$ und $\psi = (y \to x)$
(b)	$\Big(x \to ((y \to x) \to x)\Big)$ $\to \Big((x \to (y \to x)) \to (x \to x)\Big)$	Instanz von Axiom 2 mit $\varphi = x,\ \psi = (y \to x),\ \vartheta = x$
(c)	$(x \to (y \to x)) \to (x \to x)$	Modus Ponens (MP) angewandt auf (a), (b)
(d)	$x \to (y \to x)$	Instanz von Axiom 1 mit $\varphi = x$ und $\psi = y$
(e)	$x \to x$	MP angewandt auf (c), (d)

T1.21 Beweis 4-Farben-Satz (unendliche Variante)

Theorem 1.44 (4-Farben-Satz, beliebige Graphen).
Jeder (möglicherweise unendliche) planare Graph ist 4-färbbar.

Beweis. Sei $G = (V, E)$ ein (möglicherweise unendlicher) planarer Graph. Definiere folgende Formelmenge.

$$\begin{aligned}\Gamma = &\{x_{v1} \vee x_{v2} \vee x_{v3} \vee x_{v4} \mid v \in V\} \\ &\cup \{\neg(x_{vi} \wedge x_{vj} \mid v \in V,\ 1 \le i < j \le 4\} \\ &\cup \{\neg(x_{vi} \wedge x_{wi} \mid \{v, w\} \in E,\ 1 \le i \le 4\}\end{aligned}$$

Wir haben also für jeden Knoten $v \in V$ vier Aussagenvariablen x_{vi}, $1 \le i \le 4$, eingeführt. Intuitiv steht x_{vi} für: „Knoten v ist mit Farbe i gefärbt“. Die Formeln in Γ drücken nun aus, dass es sich um eine zulässige Färbung handelt:

- Jeder Knoten ist mit mindestens einer Farbe gefärbt (1. Zeile).
- Jeder Knoten ist mit höchstens einer Farbe gefärbt (2. Zeile).
- Keine zwei benachbarten Knoten sind mit derselben Farbe gefärbt (3. Zeile).

Insbesondere entspricht jede erfüllende Belegung für Γ einer zulässigen 4-Färbung von G und umgekehrt. Es genügt also zu zeigen, dass Γ erfüllbar ist. Wegen des Kompaktheitssatzes genügt es zu zeigen:

Behauptung. Jede endliche Teilmenge $\Delta \subseteq \Gamma$ ist erfüllbar.

Beweis der Behauptung. Sei $\Delta \subseteq \Gamma$ endlich. Sei $V' \subseteq V$ die Menge der Knoten $v \in V$, sodass x_{vi} in Δ vorkommt für mindestens ein i. Wir betrachten den endlichen Teilgraphen G' von G, den man durch Einschränkung auf die Knotenmenge V' erhält. G' ist ein

endlicher planarer Graph, also laut 4-Farben-Satz (endliche Variante) 4-färbbar, und jede 4-Färbung liefert eine erfüllende Belegung für Δ.

Wegen der Behauptung und dem Kompaktheitssatz ist Γ erfüllbar. Jede erfüllende Belegung liefert eine 4-Färbung von G. ❑

T1.22 Beweis Kompaktheitssatz

Theorem 1.41 (Kompaktheitssatz).
Für alle (potentiell unendlichen) Mengen $\Gamma \subseteq \mathsf{AL}$ gilt:

Γ ist erfüllbar gdw. jede endliche Teilmenge von Γ erfüllbar ist.

Beweis. Die Richtung „$\Rightarrow$“ ist trivial, wenn man sich die Definition der Erfüllbarkeit für Formel*mengen* in Erinnerung ruft (Folie 72).

Um „$\Leftarrow$“ zu zeigen, nennen wir fortan eine Menge $\Gamma \subseteq \mathsf{AL}$ *Limit-erfüllbar*, wenn alle endlichen Teilmengen von Γ erfüllbar sind. Wir müssen also zeigen:

Wenn Γ Limit-erfüllbar ist, dann ist Γ erfüllbar.

Sei also Γ Limit-erfüllbar. Da die Menge Var aller Aussagenvariablen abzählbar ist, gibt es eine Aufzählung

$$\varphi_1, \varphi_2, \varphi_3, \ldots$$

von AL.[2] Wir wollen nun Γ in dem Sinne „vervollständigen“, dass für jede Formel φ_i entweder φ_i oder $\neg\varphi_i$ enthalten ist. Dies tun wir schrittweise, indem wir eine aufsteigende Folge

$$\Gamma = \Gamma_0 \subseteq \Gamma_1 \subseteq \Gamma_2 \subseteq \ldots$$

von Limit-erfüllbaren Formelmengen Γ_i wie folgt konstruieren. Um zu beschreiben, wie man Γ_{i+1} aus Γ_i erhält, beobachten wir zunächst:

Behauptung. $\Gamma_i \cup \{\varphi_{i+1}\}$ oder $\Gamma_i \cup \{\neg\varphi_{i+1}\}$ ist Limit-erfüllbar.

Beweis der Behauptung. Per Kontraposition: wir nehmen an, dass beide Mengen nicht Limit-erfüllbar seien. Also gibt es endliche Mengen $\Delta, \Delta' \subseteq \Gamma_i$, so dass $\Delta \cup \{\varphi_{i+1}\}$ und $\Delta' \cup \{\neg\varphi_{i+1}\}$ unerfüllbar sind. Dann ist $\Delta \cup \Delta' \subseteq \Gamma_i$ unerfüllbar. Dies steht jedoch im Widerspruch zur Limit-Erfüllbarkeit von Γ_i.

Nun setzen wir:

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{\varphi_{i+1}\} & \text{falls } \Gamma_i \cup \{\varphi_{i+1}\} \text{ Limit-erfüllbar ist} \\ \Gamma_i \cup \{\neg\varphi_{i+1}\} & \text{sonst} \end{cases}$$

[2] Da sowohl die Formellänge als auch die Anzahl der Variablen unbegrenzt wachsen können, darf man die Formeln nicht lexikographisch aufzählen, sondern muss *Dove tailing* verwenden.

Sei weiterhin:

$$\Gamma_\omega = \bigcup_{i \geq 0} \Gamma_i$$

Da alle Γ_i Limit-erfüllbar sind, ist dies auch Γ_ω:

Behauptung.

$$\Gamma_\omega \text{ ist Limit-erfüllbar.} \qquad (*)$$

Beweis der Behauptung. Angenommen, dies sei nicht der Fall. Dann gibt es eine endliche Teilmenge $\Delta \subseteq \Gamma_\omega$, die unerfüllbar ist. Da Δ endlich ist, gibt es ein $\ell \geq 0$ mit $\Delta \subseteq \Gamma_\ell$. Dann wäre aber Γ_ℓ auch nicht Limit-erfüllbar, was der obigen Definition der Γ_ℓ widerspricht.

Wir verwenden nun die Definition und Limit-Erfüllbarkeit von Γ_ω, um eine erfüllende Belegung für Γ_ω zu konstruieren. Diese ist wegen $\Gamma \subseteq \Gamma_\omega$ dann auch die gewünschte erfüllende Belegung für Γ. Wir definieren V wie folgt.

$$V(x) = \begin{cases} 1 & \text{wenn } x \in \Gamma_\omega \\ 0 & \text{sonst (d. h. } \neg x \in \Gamma_\omega) \end{cases}$$

Wir brauchen nur noch zu zeigen: $V \models \Gamma_\omega$. Dazu zeigen wir per Induktion über die Struktur von φ:

$$V \models \varphi \quad \text{gdw.} \quad \varphi \in \Gamma_\omega \qquad \text{für alle } \varphi \in \mathsf{AL}$$

Induktionsanfang. Wenn φ Variable ist, dann folgt die Behauptung aus der Definition von V. Die Konstante 1 (bzw. 0) ist nach Konstruktion in Γ_ω enthalten (bzw. nicht enthalten).

Induktionsschritt. Hier genügt es zwei Fälle zu unterscheiden:

- $\varphi = \neg\psi$.
  Dann gilt:

$$\begin{aligned} V \models \varphi \quad &\text{gdw.} \quad V \not\models \psi && \text{(Semantik von } \neg) \\ &\text{gdw.} \quad \psi \notin \Gamma_\omega && \text{(Induktionsvoraussetzung)} \\ &\text{gdw.} \quad \varphi \in \Gamma_\omega && \text{(Konstruktion von } \Gamma_\omega) \end{aligned}$$

- $\varphi = \psi \wedge \vartheta$.
  Für die Richtung „⇒“ nehmen wir $V \models \varphi$ an, d. h. $V \models \psi$ und $V \models \vartheta$. Nach Induktionsvoraussetzung gilt dann auch $\psi \in \Gamma_\omega$ und $\vartheta \in \Gamma_\omega$. Nach Konstruktion von Γ_ω muss eine der beiden Formeln $\psi \wedge \vartheta$ und $\neg(\psi \wedge \vartheta)$ in Γ_ω enthalten sein. Wenn $\neg(\psi \wedge \vartheta)$ enthalten wäre, dann wäre $\{\psi, \vartheta, \neg(\psi \wedge \vartheta)\}$ eine unerfüllbare Teilmenge von Γ_ω, was im Widerspruch zu $(*)$ stünde. Also $\psi \wedge \vartheta \in \Gamma_\omega$, wie gewünscht.

  Für die Richtung „⇐“ nehmen wir $\psi \wedge \vartheta \in \Gamma_\omega$ an. Dann sind auch $\psi, \vartheta \in \Gamma_\omega$: wäre z. B. $\psi \notin \Gamma_\omega$, dann wäre $\neg\psi \in \Gamma_\omega$, und damit wäre $\{\psi \wedge \vartheta, \neg\psi\} \subseteq \Gamma_\omega$ eine unerfüllbare Teilmenge, im Widerspruch zu $(*)$ – und analog für ϑ. Nach Induktionsvoraussetzung folgt nun $V \models \psi$ und $V \models \vartheta$, also $V \models \psi \wedge \vartheta$. ❑

Teil II.

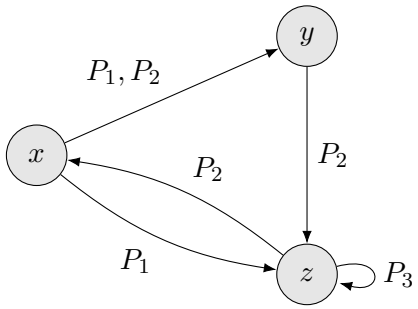
Prädikatenlogik Grundlagen

T2.1 Beispiel: Strukturen als kantenbeschriftete Graphen

Sei $\mathfrak{A} = (A, P_1^{\mathfrak{A}}, P_2^{\mathfrak{A}}, P_3^{\mathfrak{A}})$ mit

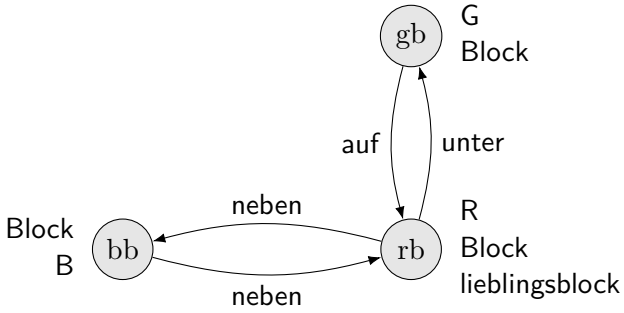
$$\begin{aligned} A &= \{x, y, z\} \\ P_1^{\mathfrak{A}} &= \{(x, y),\ (x, z)\} \\ P_2^{\mathfrak{A}} &= \{(x, y),\ (y, z),\ (z, x)\} \\ P_3^{\mathfrak{A}} &= \{(z, z)\} \end{aligned}$$

Diese Struktur entspricht dem folgenden Graphen.



T2.2 Beispiel: Struktur für die drei Blöcke R, G, B

Da die gegebene Struktur nur unäre und binäre Relationssymbole enthält, kann man sie leicht als Graph darstellen:



T2.3 Beispiel: Struktur für Film-Datenbank

Die Struktur ist $\mathfrak{A} = (A, \mathsf{Film}^{\mathfrak{A}}, \mathsf{Schauspieler\_in}^{\mathfrak{A}})$ mit

$$\begin{aligned} A &= \{\, \text{Die Vögel},\ \text{Marnie},\ \text{Goldfinger},\ 1963,\ 1964,\\ &\qquad \text{Hitchcock},\ \text{Hamilton},\ \text{Connery},\ \text{Hedren} \,\} \\ \mathsf{Film}^{\mathfrak{A}} &= \{\, (\text{Die Vögel}, 1963, \text{Hitchcock}),\\ &\qquad (\text{Marnie}, 1963, \text{Hitchcock}),\\ &\qquad (\text{Goldfinger}, 1964, \text{Hamilton}) \,\} \\ \mathsf{Schauspieler\_in}^{\mathfrak{A}} &= \{\, (\text{Connery}, \text{Marnie}),\\ &\qquad (\text{Connery}, \text{Goldfinger}),\\ &\qquad (\text{Hedren}, \text{Marnie}) \,\} \end{aligned}$$

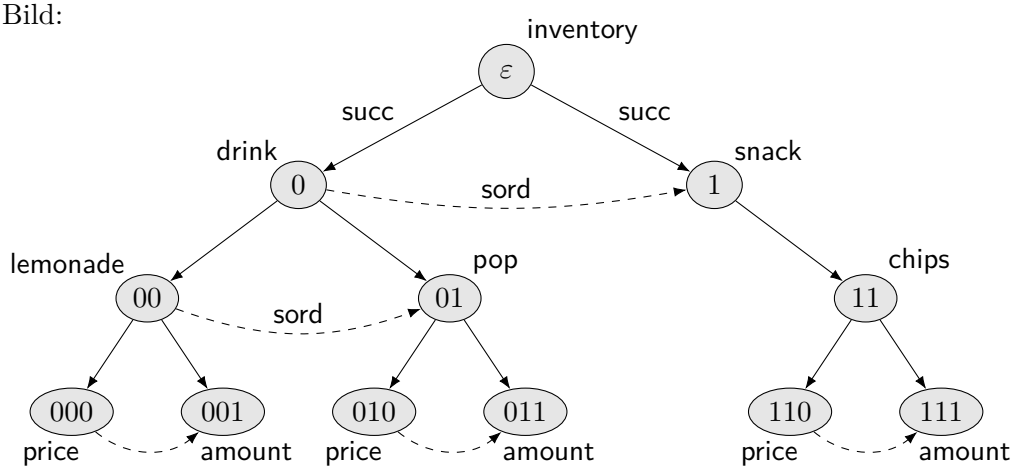
Da $\mathsf{Film}^{\mathfrak{A}}$ eine ternäre Relation ist, kann man $\mathfrak{A}$ höchstens als Hypergraphen darstellen. Das ist aber kaum übersichtlicher als die Tupel-Schreibweise.

T2.4 Beispiel: Struktur für XML-Dokument

Die Struktur ist $\mathfrak{A} = (A, \mathsf{succ}^{\mathfrak{A}}, \mathsf{sord}^{\mathfrak{A}}, \mathsf{inventory}^{\mathfrak{A}}, \mathsf{drink}^{\mathfrak{A}}, \ldots, \mathsf{amount}^{\mathfrak{A}})$ mit

$$\begin{aligned} A &= \{\varepsilon, 0, 1, 00, 01, 11, 000, 001, 010, 011, 110, 111\} \\ \mathsf{succ}^{\mathfrak{A}} &= \{(w, w0) \mid w0 \in A\} \cup \{(w, w1) \mid w1 \in A\} \\ \mathsf{sord}^{\mathfrak{A}} &= \{(w0, w1) \mid w0, w1 \in A\} \\ \mathsf{inventory}^{\mathfrak{A}} &= \{\varepsilon\} \\ &\vdots \\ \mathsf{amount}^{\mathfrak{A}} &= \{001, 011, 111\} \end{aligned}$$

Als Bild:

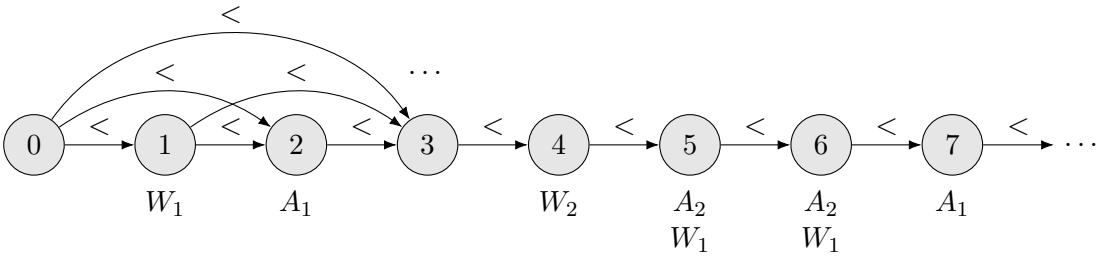


T2.5 Beispiel: Ordnungen als Strukturen

Betrachte zwei kommunizierende Prozesse. Wir verwenden 4 unäre Relationssymbole:

- W_i: Prozess i wartet auf Eintritt in den kritischen Abschnitt
- A_i: Prozess i ist im kritischen Abschnitt

mit $i \in \{1, 2\}$.
Jede Struktur $\mathfrak{A} = (\mathbb{N}, <, W_1^{\mathfrak{A}}, W_2^{\mathfrak{A}}, A_1^{\mathfrak{A}}, A_2^{\mathfrak{A}})$ modelliert dann einen zeitlichen Verlauf der beiden Prozesse, z. B.:

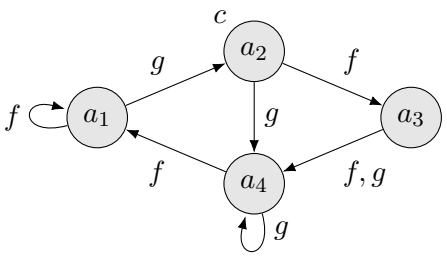


Diese Struktur modelliert einen möglichen Verlauf, und zwar denjenigen, in dem

- Prozess 1 in Zeitpunkt 1 auf den kritischen Bereich wartet, diesen in Zeitpunkt 2 betritt und in Zeitpunkt 3 wieder verlässt,
- Prozess 2 in Zeitpunkt 4 auf den kritischen Bereich wartet, diesen in Zeitpunkt 5 betritt und erst in Zeitpunkt 7 wieder verlässt,
- Prozess 1 in Zeitpunkten 5 und 6 auf den kritischen Bereich wartet, diesen (wegen Prozess 2) erst in Zeitpunkt 7 betritt usw.

T2.6 Beispiel: Zuweisung

Betrachte folgende Struktur $\mathfrak{A}$ mit unären Funktionssymbolen f, g und Konstante c.



Dann muss für *jede* Zuweisung β gelten:

$$\beta(c) = a_2 \quad (\text{da } c^{\mathfrak{A}} = a_2)$$
$$\beta(g(f(c))) = a_4 \quad (\text{da } g^{\mathfrak{A}}(f^{\mathfrak{A}}(c)) = a_4)$$

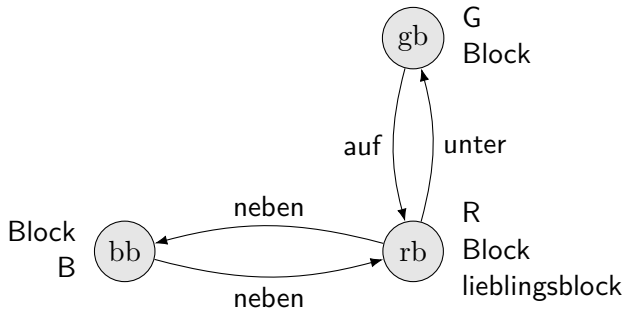
Außerdem gilt:

- Wenn $\beta(x) = a_1$, dann $\beta(g(f(x))) = a_2$.
- Wenn $\beta(x) = a_3$, dann $\beta(g(f(x))) = a_4$.

(Beachte jeweils: in $g(f(\cdot))$ wird zuerst f angewendet und dann g.)

T2.7 Beispiel: Erfülltheitsrelation

Betrachte die Struktur $\mathfrak{A}$ aus **Beispiel 1** (T2.2):



Für alle Zuweisungen β gilt:

- $\mathfrak{A}, \beta \models \exists x\, \mathsf{R}(x) \wedge \exists x\, \mathsf{G}(x) \wedge \exists x\, \mathsf{B}(x)$,
  denn $\mathfrak{A}, \beta \models \exists x\, \mathsf{R}(x)$ und $\mathfrak{A}, \beta \models \exists x\, \mathsf{G}(x)$ und $\mathfrak{A}, \beta \models \exists x\, \mathsf{B}(x)$,
  denn $\mathfrak{A}, \beta[x/\mathrm{rb}] \models \mathsf{R}(x)$ und $\mathfrak{A}, \beta[x/\mathrm{gb}] \models \mathsf{G}(x)$ und $\mathfrak{A}, \beta[x/\mathrm{bb}] \models \mathsf{B}(x)$
- $\mathfrak{A}, \beta \not\models \exists x\, \mathsf{neben}(x, x)$
- $\mathfrak{A}, \beta \models \exists x \Big(\mathsf{R}(x) \wedge \forall y \, (\mathsf{auf}(y, x) \rightarrow \mathsf{G}(y)) \Big)$
- $\mathfrak{A}, \beta \models \forall x \Big((\mathsf{R}(x) \wedge \mathsf{B}(x)) \rightarrow \mathsf{G}(x) \Big)$

Betrachte β mit $\beta(x) = \mathrm{bb}$, $\beta(y) = \mathrm{gb}$ und $\beta(z) = \mathrm{gb}$. Dann gilt:

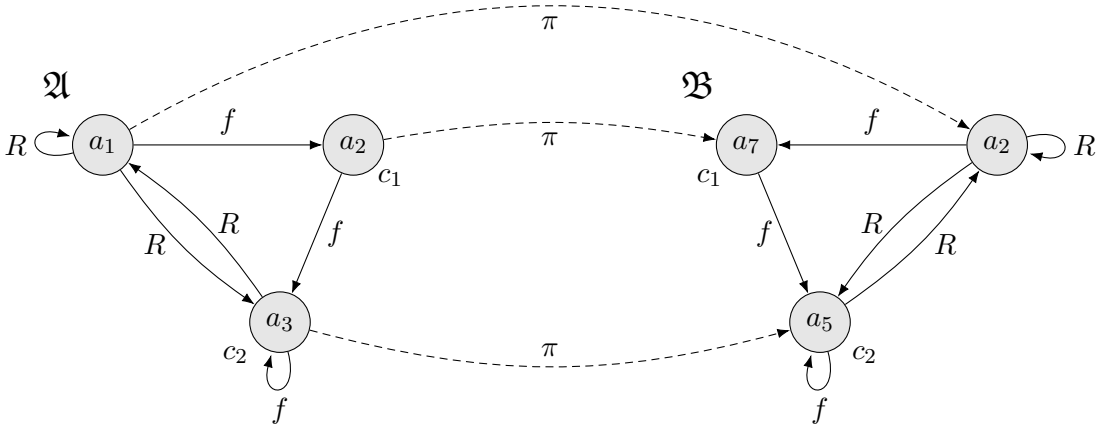
- $\mathfrak{A}, \beta \models x \neq \mathsf{lieblingsblock}$
- $\mathfrak{A}, \beta \models \exists z \Big(\mathsf{neben}(x, z) \wedge \mathsf{unter}(z, y) \Big)$

Nun betrachte die Struktur $\mathfrak{N} = (\mathbb{N}, +, \cdot, 0, 1)$ aus **Beispiel 4**. Dann gilt:

- $\mathfrak{N}, \beta \models \forall x \forall y \, (x + y = y + x)$ für *alle* β
- $\mathfrak{N}, \beta \models \underbrace{\forall y \forall z \, (y \cdot z = \underline{x} \rightarrow y = 1 \vee z = 1) \wedge x \neq 0 \wedge x \neq 1}_{\text{Abkürzung: } \mathbf{Prim}(\boldsymbol{x})}$
  genau dann, wenn $\beta(x)$ eine Primzahl ist
- $\mathfrak{N}, \beta \models \underbrace{\exists z \, (z \neq 0 \wedge \underline{x} + z = \underline{y})}_{\text{Abkürzung: } \boldsymbol{y > x}}$ genau dann, wenn $\beta(y) > \beta(x)$
- $\mathfrak{N}, \beta \models \forall x \Big(\mathsf{Prim}(x) \rightarrow \exists y \, (y > x \wedge \mathsf{Prim}(y)) \Big)$
  (Satz von Euklid: es gibt unendlich viele Primzahlen)
- Es ist unbekannt, ob $\mathfrak{N}, \beta \models \forall x \, \exists y \, (y > x \wedge \mathsf{Prim}(y) \wedge \mathsf{Prim}(\underbrace{y + 2}_{\text{Abkürz. für } y+1+1}))$. Dies ist ein schweres offenes Problem der Zahlentheorie: gibt es unendlich viele Primzahl*zwillinge*?

T2.8 Beispiel: Isomorphismus

Betrachte die Signatur, die aus den Konstantensymbolen c_1, c_2, einem einstelligen Funktionssymbol f und einem zweistelligen Relationssymbol R besteht. Dann ist π (gestrichelte Pfeile) ein Isomorphismus zwischen den folgenden beiden Strukturen $\mathfrak{A}$ und $\mathfrak{B}$.



T2.9 Beweis des Isomorphielemmas

Lemma 2.10 (Isomorphielemma). Seien $\mathfrak{A}$, $\mathfrak{B}$ Strukturen und $\pi : A \to B$ ein Isomorphismus.

Dann gilt für alle Formeln $\varphi(x_1, \ldots, x_n)$ und alle $a_1, \ldots, a_n \in A$:

$$\mathfrak{A} \models \varphi[a_1, \ldots, a_n] \text{ gdw. } \mathfrak{B} \models \varphi[\pi(a_1), \ldots, \pi(a_n)]$$

Beweis. Sei $\pi : A \to B$ ein Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$. Wir schreiben die zu zeigende Aussage zunächst so auf, dass sie die Zuweisung β explizit macht, was für den Beweis bequemer ist:

Es ist zu zeigen:

> Für alle FO-Formeln φ und für alle Interpretationen $\mathfrak{I}_{\mathfrak{A}} = (\mathfrak{A}, \beta)$ und $\mathfrak{I}_{\mathfrak{B}} = (\mathfrak{B}, \beta')$ mit $\beta'(x) = \pi(\beta(x))$ für alle $x \in \mathsf{Frei}(\varphi)$ gilt:
>
> $$\mathfrak{I}_{\mathfrak{A}} \models \varphi \qquad \text{gdw.} \qquad \mathfrak{I}_{\mathfrak{B}} \models \varphi \qquad (*)$$

Da Syntax und Semantik der Prädikatenlogik in zwei Schritten aufgebaut ist (Terme und Formeln), bietet es sich an, $(*)$ ebenfalls in zwei Schritten zu beweisen.

Schritt 1. Man zeigt zunächst leicht per Induktion über die Struktur von t, dass für alle Terme mit Variablen aus $\mathsf{Frei}(\varphi)$ gilt:

$$\beta'(t) = \pi(\beta(t)) \qquad (**)$$

Induktionsanfang.

- Für $t = x$ folgt $(**)$ aus der Voraussetzung $\beta'(x) = \pi(\beta(x))$ für alle $x \in \mathsf{Frei}(\varphi)$.
- Für $t = c$ (Konstante) gilt $(**)$ wegen

$$\begin{aligned} \beta'(c) &= c^{\mathfrak{B}} && \text{(Definition Zuweisung)} \\ &= \pi(c^{\mathfrak{A}}) && \text{(Definition Isomorphismus)} \\ &= \pi(\beta(c)) && \text{(Definition Zuweisung)} \end{aligned}$$

Induktionsschritt. Sei $t = f(t_1, \ldots, t_n)$. Dann gilt:

$$\begin{aligned} \beta'(t) &= \beta'(f(t_1, \ldots, t_n)) \\ &= f^{\mathfrak{B}}(\beta'(t_1), \ldots, \beta'(t_n)) && \text{(Definition Zuweisung)} \\ &= f^{\mathfrak{B}}(\pi(\beta(t_1)), \ldots, \pi(\beta(t_n))) && \text{(Induktionsvoraussetzung)} \\ &= \pi(f^{\mathfrak{A}}(\beta(t_1), \ldots, \beta(t_n))) && \text{(Definition Isomorphismus)} \\ &= \pi(\beta(f(t_1, \ldots, t_n)) && \text{(Definition Zuweisung)} \\ &= \pi(\beta(t)) \end{aligned}$$

Schritt 2. Nun können wir $(*)$ per Induktion über die Struktur von φ beweisen und dabei $(**)$ verwenden.

Induktionsanfang.

- Wenn $\varphi = (t_1 = t_2)$, dann haben wir:

$$\begin{aligned} \mathfrak{I}_A \models t_1 = t_2 \quad &\text{gdw.} \quad \beta(t_1) = \beta(t_2) && \text{(Def. Semantik FO)} \\ &\text{gdw.} \quad \pi(\beta(t_1)) = \pi(\beta(t_2)) && \text{(weil } \pi \text{ Funktion und injektiv)} \\ &\text{gdw.} \quad \beta'(t_1) = \beta'(t_2) && (**) \\ &\text{gdw.} \quad \mathfrak{I}_B \models t_1 = t_2 && \text{(Def. Semantik FO)} \end{aligned}$$

- Wenn $\varphi = P(t_1, \ldots, t_n)$, dann haben wir:

$$\begin{aligned} \mathfrak{I}_A \models P(t_1, \ldots, t_n) \quad &\text{gdw.} \quad (\beta(t_1), \ldots, \beta(t_n)) \in P^{\mathfrak{A}} && \text{(Def. Semantik FO)} \\ &\text{gdw.} \quad (\pi(\beta(t_1)), \ldots, \pi(\beta(t_n))) \in P^{\mathfrak{B}} && \text{(weil } \pi \text{ Isomorphismus)} \\ &\text{gdw.} \quad (\beta'(t_1), \ldots, \beta'(t_n)) \in P^{\mathfrak{B}} && (**) \\ &\text{gdw.} \quad \mathfrak{I}_B \models P(t_1, \ldots, t_n) && \text{(Def. Semantik FO)} \end{aligned}$$

Induktionsschritt.

- Die Fälle $\varphi = \neg\psi$, $\varphi = \psi_1 \wedge \psi_2$ und $\varphi = \psi_1 \vee \psi_2$ sind eine gute Gelegenheit zum Üben – probiert es aus!

- Im Fall $\varphi = \exists x\, \psi$ argumentieren wir wie folgt.

$$\begin{array}{lll} \mathfrak{I}_A \models \exists x\, \psi & \text{gdw. } \mathfrak{A}, \beta[x/a] \models \psi \text{ für ein } a \in A & \text{(Def. Semantik FO)} \\ & \text{gdw. } \mathfrak{B}, \beta'[x/\pi(a)] \models \psi \text{ für ein } a \in A & \text{(Induktionsvoraussetzung)} \\ & \text{gdw. } \mathfrak{B}, \beta'[x/b] \models \psi \text{ für ein } b \in B & \text{(weil } \pi \text{ surjektiv)} \\ & \text{gdw. } \mathfrak{I}_B \models \exists x\, \psi & \text{(Def. Semantik FO)} \end{array}$$

  Um im zweiten Schritt die Induktionsvoraussetzung anwenden zu können, müssen wir natürlich prüfen, ob die Voraussetzung von $(*)$ von $(\mathfrak{A}, \beta[x/a])$ und $(\mathfrak{B}, \beta'[x/\pi(a)])$ erfüllt sind. Dies ist der Fall, denn wir arbeiten unter der Voraussetzung

$$\beta'(y) = \pi(\beta(y)) \quad \text{für alle } y \in \mathsf{Frei}(\exists x\, \psi),$$

  woraus folgt, dass

$$\beta'[x/\pi(a)](y) = \pi(\beta'[x/a](y)) \quad \text{für alle } y \in \mathsf{Frei}(\exists x\, \psi),$$

  d. h. $y = x$ ist möglich.
- Im Fall $\varphi = \forall x\, \psi$ argumentieren wir analog. ❑

T2.10 Beispiel für den Auswertungsalgorithmus

Betrachte die Eingabe $(\mathfrak{A}, \beta, \varphi)$ bestehend aus

- der Struktur $\mathfrak{A}$: P (a) $\xleftarrow{R}$ (b)
- der leeren Zuweisung β und
- dem Satz $\varphi = \forall x \underbrace{\exists y \Big(\overbrace{P(x) \vee R(x,y)}^{\vartheta(x,y)} \Big)}_{\psi(x)}$.

Ein Lauf des Algorithmus auf dieser Eingabe wird durch den Baum in Abbildung 1 wiedergegeben. Dabei steht jeder Knoten für einen Aufruf von `ausw` mit den angegebenen Parametern. Die Kinder jedes Knoten stehen für die Unteraufrufe in diesem Aufruf. Dabei steht „. . . “ für die Zuweisung, die bereits im Elternknoten verwendet wurde. Die Operatoren $\forall, \exists, \vee$ geben die Art der jeweiligen Verzweigung an (äußerer Operator der Formel in diesem Aufruf). Die Zahlen 0 und 1 geben den Rückgabewert des jeweiligen Aufrufs an.

Der Aufruf $\texttt{ausw}(\mathfrak{A}, \beta, \varphi)$ liefert den Wert 1 zurück; also gilt $\mathfrak{A}, \beta \models \varphi$.

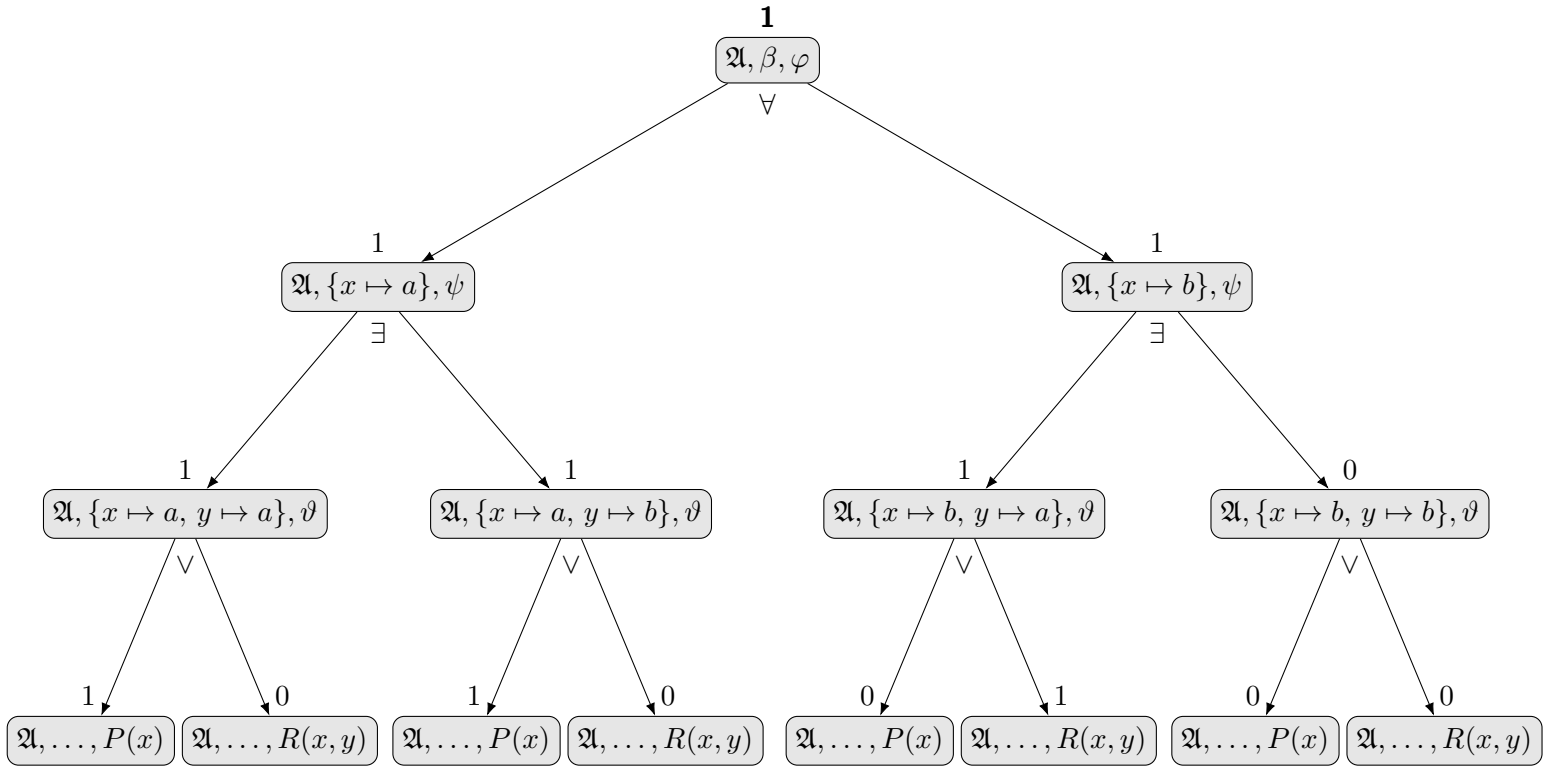


Abbildung 1.: Beispiellauf des Auswertungsalgorithmus

T2.11 Korrektheit & Platzbedarf des Auswertungsalgorithmus

Lemma 2.13. Der Algorithmus `ausw`

1. ist korrekt: $\texttt{ausw}(\mathfrak{A}, \beta, \varphi) = 1$ gdw. $\mathfrak{A}, \beta \models \varphi$;
2. benötigt nur polynomiell viel Platz.

Beweis.

1. Dies lässt sich durch einfache Induktion über die Struktur von φ zeigen: Die Fälle des Algorithmus spiegeln direkt die Semantik der Operatoren wider.
2. Jeder einzelne Fall (z. B. Prüfen, ob $\beta(t) = \beta(t')$ oder Iteration über alle $a \in A$) kann mit polynomiell viel Platz ausgeführt werden. Die Rekursionstiefe ist durch $\mathsf{st}(\varphi)$ (Schachtelungstiefe) begrenzt, also wird auch der Rekursionsstapel nur polynomiell groß. ❑

T2.12 Weitere Beispiele für Datenbankanfragen

Freie Variablen sind unterstrichen.

- „Gib alle Paare von Schauspieler_innen und Regisseur_innen, so dass erstere_r in einem Film von letzterer/m mitgespielt hat“.

$$\psi(x, y) = \exists z\, \exists z' \Big(\mathsf{Schauspieler\_in}(\underline{x}, z) \wedge \mathsf{Film}(z, z', \underline{y}) \Big)$$

$$\mathsf{ans}(\mathfrak{A}, \psi) = \{(\text{Connery}, \text{Hitchcock}),\ (\text{Connery}, \text{Hamilton}),\ (\text{Hedren}, \text{Hitchcock})\}$$

- „Gib alle Paare von verschiedenen Filmen, die im selben Jahr gedreht wurden“.

$$\vartheta(x, y) = \exists z_1\, \exists z_2\, \exists z_3 \Big(\mathsf{Film}(\underline{x}, z_1, z_2) \wedge \mathsf{Film}(\underline{y}, z_1, z_3) \wedge x \neq y \Big)$$

$$\mathsf{ans}(\mathfrak{A}, \vartheta) = \{(\text{Marnie}, \text{Goldfinger}),\ (\text{Goldfinger}, \text{Marnie})\}$$

(Man beachte das zweifache Vorkommen von z_1, was einem „equijoin“ in SQL entspricht.)

T2.13 Beispiele für domänenabhängige Formeln

Betrachte ein einstelliges Relationssymbol P und die Strukturen $\mathfrak{A}$ mit $A = \{a\}$ und $P^{\mathfrak{A}} = \{a\}$ sowie $\mathfrak{B}$ mit $B = \{a, a'\}$ und $P^{\mathfrak{B}} = \{a\}$. Die folgenden Formeln sind domänen*abhängig*:

- $\varphi(x) = \neg P(\underline{x})$,
  denn $\mathsf{ans}(\mathfrak{A}, \varphi) = \emptyset$, aber $\mathsf{ans}(\mathfrak{B}, \varphi) = \{a'\}$
- $\psi(x) = P(\underline{x}) \vee \exists y.P(y)$,
  denn $\mathsf{ans}(\mathfrak{A}, \psi) = \{a\}$, aber $\mathsf{ans}(\mathfrak{B}, \psi) = \{a, a'\}$

T2.14 Beispiele für Gültigkeit, Erfüllbarkeit, Konsequenz

- Die folgenden Sätze sind gültig:

$$\forall x\, \exists y \Big(y = f(x) \Big)$$
$$\forall x \Big(f(x) = x \to f(f(x)) = x \Big)$$

- Der folgende Satz ist erfüllbar, aber nur in Modellen mit *unendlich großem* Universum:

$$\forall x\, \exists y\, P(x,y) \quad \wedge \quad \forall x\, \forall y\, \forall z \Big(P(x,y) \wedge P(y,z) \to P(x,z) \Big) \quad \wedge \quad \neg \exists x\, P(x,x)$$

  Die drei Teile des Satzes sagen, dass

  (1) jedes Element einen „P-Nachfolger“ hat;

  (2) die Relation P transitiv ist;

  (3) kein Element sich selbst als „P-Nachfolger“ haben darf.

  Damit dieser Satz erfüllbar ist, muss es mindestens ein Element a_1 im Universum A geben (Universen dürfen nicht leer sein). Wegen (1) muss es ein Element a_2 mit $P^{\mathfrak{A}}(a_1, a_2)$ geben. Wegen (3) muss $a_2 \neq a_1$ sein. Wegen (1) muss es ein Element a_3 mit $P^{\mathfrak{A}}(a_2, a_3)$ geben. Wegen (3) muss $a_3 \neq a_2$ sein. Wegen (2) und (3) muss $a_3 \neq a_1$ sein. So kann man die Argumentation induktiv fortsetzen und erhält eine (abzählbar) unendliche Folge $a_1, a_2, a_3, \ldots$ von paarweise verschiedenen Elementen aus A.

- Folgendes ist ein Beispiel für Konsequenz:

$$\exists x\, \forall y\, (x = y) \;\models\; f(c) = c$$

  Anschaulich gesprochen: wenn es nur ein Element im Universum gibt (linke Seite), dann muss der Funktionswert jedes Elements das Element selbst sein, für eine beliebige einstellige Funktion f und ein beliebiges Element (durch die Konstante c repräsentiert).

T2.15 Beweis der Äquivalenzen für das PNF-Theorem

Lemma. Falls x nicht frei in φ vorkommt, gilt:

(1) $\varphi \vee \exists x\, \psi \;\equiv\; \exists x\, (\varphi \vee \psi)$
(2) $\varphi \wedge \exists x\, \psi \;\equiv\; \exists x\, (\varphi \wedge \psi)$
(3) $\varphi \vee \forall x\, \psi \;\equiv\; \forall x\, (\varphi \vee \psi)$
(4) $\varphi \wedge \forall x\, \psi \;\equiv\; \forall x\, (\varphi \wedge \psi)$

Beweis. Wir beweisen exemplarisch (1).
Sei $(\mathfrak{A}, \beta)$ eine beliebige Interpretation. Dann gilt:

$\mathfrak{A}, \beta \models \varphi \vee \exists x\, \psi$
gdw.
$\mathfrak{A}, \beta \models \varphi$ oder es gibt $a \in A$ mit $\mathfrak{A}, \beta[x/a] \models \psi$
gdw.
es gibt $a \in A$, so dass $\mathfrak{A}, \beta[x/a] \models \varphi$ oder $\mathfrak{A}, \beta[x/a] \models \psi$
gdw.
$\mathfrak{A}, \beta \models \exists x\, (\varphi \vee \psi)$

Der erste und dritte Schritt gilt wegen der Semantik der Operatoren $\vee, \exists$. Der zweite Schritt gilt wegen $x \notin \mathsf{Frei}(\varphi)$, denn deshalb liefert das Koinzidenzlemma:

$$\mathfrak{A}, \beta \models \varphi \quad \text{gdw.} \quad \mathfrak{A}, \beta[x/a] \models \varphi$$ ❑

T2.16 Beweis des PNF-Theorems

Theorem 2.21. Jede FO-Formel φ kann in Linearzeit in eine äquivalente Formel in PNF gewandelt werden.

Beweis. Wir beweisen das Theorem per Induktion über die Struktur von φ, nach [Grä19].

Induktionsanfang. Wenn φ atomar ist (d. h. $\varphi = (t_1 = t_2)$ oder $\varphi = P(t_1, \ldots, t_n)$), dann enthält die Formel keinen Quantor, ist also trivialerweise in PNF.

Induktionsschritt. Wir unterscheiden drei Fälle:

(1) $\varphi = \neg\psi$.

Nach Induktionsvoraussetzung gibt es eine PNF-Formel

$$\psi' \;=\; \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_n x_n\, \vartheta$$

mit $\psi' \equiv \psi$. Die Dualität von $\exists$ und $\forall$[3] liefert

$$\varphi \;\equiv\; \underbrace{\overline{\mathsf{Q}_1} x_1 \cdots \overline{\mathsf{Q}_n} x_n\, \neg\vartheta}_{\text{in PNF}},$$

wobei $\overline{\exists} = \forall$ und $\overline{\forall} = \exists$.

[3] d. h. für alle Formeln ξ gilt $\neg\forall x\, \xi \equiv \exists x \neg\xi$ und $\neg\exists x\, \xi \equiv \forall x \neg\xi$

(2) $\varphi = \psi_1 \circ \psi_2$ mit $\circ \in \{\wedge, \vee\}$.

Nach Induktionsvoraussetzung gibt es PNF-Formeln ψ_1', ψ_2' mit $\psi_1' \equiv \psi_1$ und $\psi_2' \equiv \psi_2$. Durch Variablenumbenennung erreichen wir, dass ψ_1' und ψ_2' folgende Form haben:

$$\begin{aligned} \psi_1' &= \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_n y_n \, \vartheta_1 \\ \psi_2' &= \mathsf{Q}_1' y_1 \cdots \mathsf{Q}_m' y_m \, \vartheta_2 \end{aligned}$$

wobei die Variablen $x_1, \ldots, x_n, y_1, \ldots, y_m$ paarweise verschieden sowie verschieden von allen freien Variablen in ψ_1' und ψ_2' sind.

Offenbar ist

$$\varphi' \;=\; \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_n x_n \, \mathsf{Q}_1' y_1 \cdots \mathsf{Q}_m' y_m \, (\vartheta_1 \circ \vartheta_2)$$

in PNF. Da $y_1, \ldots, y_m$ nicht in ψ_1' vorkommen und $x_1, \ldots, x_n$ nicht in ψ_2', liefern die Äquivalenzen des obigen Lemmas, wenn man sie $(n+m)$-mal auf $\psi_1' \circ \psi_2'$ anwendet: $\varphi' \equiv \varphi$.

(3) $\varphi = \mathsf{Q} x \, \psi$ mit $\mathsf{Q} \in \{\exists, \forall\}$.

Nach Induktionsvoraussetzung gibt es eine PNF-Formel $\psi' = \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_n x_n \, \vartheta$ mit $\psi' \equiv \psi$. Durch Umbenennen kann erreicht werden, dass $x \notin \{x_1, \ldots, x_n\}$. Dann ist $\mathsf{Q} x \, \psi'$ äquivalent zu φ und in PNF (insbesondere haben wir durch das Umbenennen sichergestellt, dass $\mathsf{Q} x \, \psi'$ bereinigt ist). ❏

T2.17 Beispiel zur Umwandlung in PNF

Sei

$$\varphi \;=\; \neg\forall x \Big(\underbrace{R(x,x)}_{\text{PNF}} \;\wedge\; \underbrace{\forall \underline{x} \, \exists y \, R(\underline{x}, y)}_{\text{PNF}} \Big).$$

Um φ in PNF zu wandeln, gehen wir gemäß der strukturellen Induktion im vorangehenden Beweis von innen nach außen vor. Die größten Teilformeln, die bereits in PNF sind, sind markiert. Als nächstes muss die Konjunktion gemäß Fall (2) umgeformt werden. Dazu ist es zunächst notwendig, das *gebundene* Vorkommen von x in der rechten Teilformel (durch Unterstreichen markiert) umzubenennen:

$$\varphi \;\equiv\; \neg\forall x \Big(R(x,x) \wedge \forall \underline{z} \, \exists y \, R(\underline{z}, y) \Big).$$

Die Konstruktion in Fall (2) liefert dann:

$$\varphi \;\equiv\; \neg \; \underbrace{\forall x \, \forall z \, \exists y \Big(R(x,x) \wedge R(z,y) \Big)}_{\text{PNF}}$$

Gemäß Fall (1) müssen wir jetzt nur noch Quantoren „umdrehen“ und die Negation nach innen ziehen:

$$\varphi \;\equiv\; \exists x \, \exists z \, \forall y \, \neg \Big(R(x,x) \wedge R(z,y) \Big)$$

T2.18 Beispiel für das Postsche Korrespondenzproblem (PKP)

Sei $F = \underbrace{(0,1)}_{\text{Index 1}}, \underbrace{(1,10)}_{\text{Index 2}}, \underbrace{(01,1)}_{\text{Index 3}}$. Dann ist die Indexfolge $2,3$ eine Lösung, denn

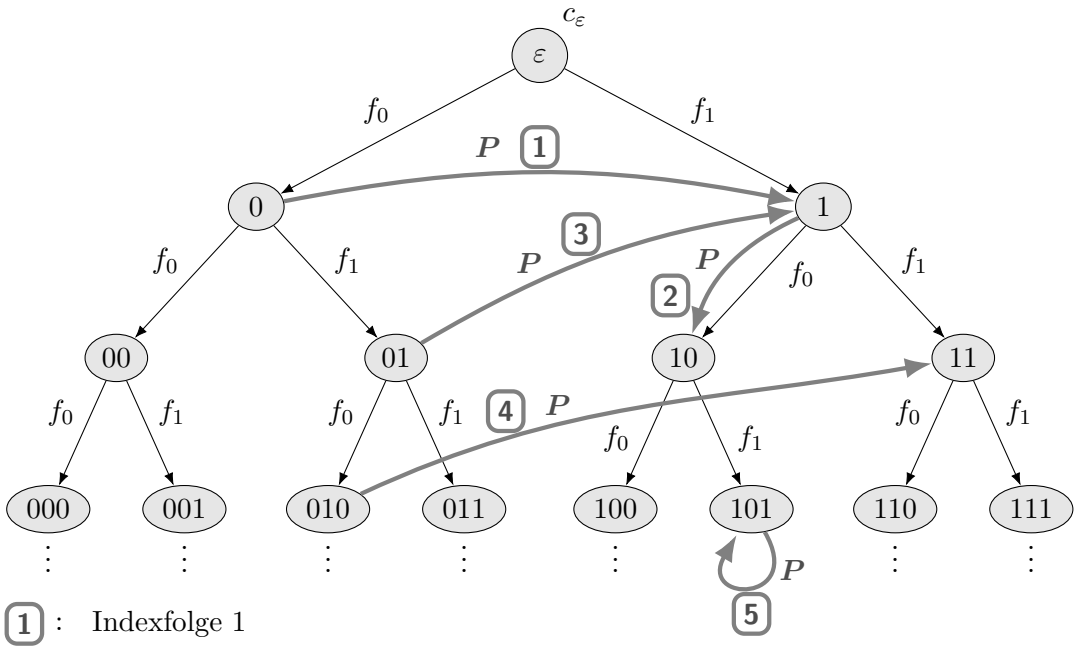
- die linke Konkatenation liefert $1 \cdot 01 = 101$ und
- die rechte Konkatenation liefert $10 \cdot 1 = 101$.

Schematisch kann man das auch so darstellen:

	2	3	
Linke Konkatenation	1	0	1
Rechte Konkatenation	1	0	1

T2.19 Beispiel für die Kodierung des PKP

Für das PKP F aus dem vorangehenden Beispiel erhält man ein unendliches Modell, das wie folgt aussieht („$\cdots$“ deuten an, dass sowohl das Universum A als auch die Interpretationen der Funktionen f_1, f_1 und des Relationssymbols P unendlich sind):



1 : Indexfolge 1

2 : Indexfolge 2

3 : Indexfolge 3

4 : 3 $\cdot (0,1)$, also Indexfolge $3,1$

5 : 2 $\cdot (01,1)$, also Indexfolge $2,3$

P-Kante Nr. 5 bezeugt, dass F eine Lösung hat.

T2.20 Beweis der Korrektheit der PKP-Reduktion

Wir verwenden die auf Folie 59 eingeführte Notation $t_w(x)$:

- Für ein Wort $w = w_1 \cdots w_n \in \{0,1\}^*$ steht $t_w(x)$ für $f_{w_n}(f_{w_{n-1}}(\cdots f_{w_1}(x)))$.
- Wir schreiben außerdem $t_w^{\mathfrak{A}}(x)$ für $f_{w_n}^{\mathfrak{A}}(f_{w_{n-1}}^{\mathfrak{A}}(\cdots f_{w_1}^{\mathfrak{A}}(x)))$.

Lemma 2.24. F hat eine Lösung gdw. φ_F gültig ist.

Beweis. Sei $F = (u_1, v_1), \ldots, (u_k, v_k)$. Wir beweisen beide Richtungen von „genau dann, wenn“ getrennt.

„⇐“ Sei φ_F gültig. Dann ist jede Struktur ein Modell von φ_F. Unter diesen Modellen gibt es auch Strukturen, die völlig anders beschaffen sind, als wir es im vorigen Bild gezeichnet haben, d. h. das Universum ist kein Baum und/oder die Funktions- und Relationssymbole werden anders interpretiert, als wir es beabsichtigen. Wir betrachten jedoch ein spezielles Modell, das sich so verhält, wie wir es wollen, und das deshalb auch *kanonisches Modell* genannt wird. Es ist die Struktur

$$\mathfrak{A} = \{A, c_\varepsilon, f_0, f_1, P) \qquad \text{mit}$$

$$\begin{aligned} A &= \{0,1\}^* \\ c_\varepsilon^{\mathfrak{A}} &= \varepsilon \\ f_0^{\mathfrak{A}}(w) &= w0 \ \text{ für alle } w \in A \\ f_1^{\mathfrak{A}}(w) &= w1 \ \text{ für alle } w \in A \\ P^{\mathfrak{A}} &= \{(u,v) \mid \text{es gibt } i_1, \ldots, i_\ell \text{ mit } u = u_{i_1} \cdots u_{i_\ell} \text{ und } v = v_{i_1} \cdots v_{i_\ell}\} \end{aligned}$$

Für dieses Modell gilt:

(1) $\mathfrak{A} \models \varphi$: nach Definition von $P^{\mathfrak{A}}$ und wegen $t_w(c_\varepsilon)^{\mathfrak{A}} = w$ für alle $w \in \{0,1\}^*$

(2) $\mathfrak{A} \models \psi$: Wenn $P^{\mathfrak{A}}(u,v)$, dann folgt mit Definition von $P^{\mathfrak{A}}$ auch $P^{\mathfrak{A}}(uu_i, vv_i)$ für alle $i \le k$. Wegen $t_w^{\mathfrak{A}}(w') = w'w$ für alle $w, w \in \{0,1\}^*$ gilt so $P^{\mathfrak{A}}(t_{u_i}^{\mathfrak{A}}(u), t_{v_i}^{\mathfrak{A}}(v))$.

Weil φ_F gültig ist, folgt aus (1) und (2) $\mathfrak{A} \models \exists x\, P(x,x)$. Nach Definition von $P^{\mathfrak{A}}$ hat also F eine Lösung.

„⇒“ Sei $i_1, \ldots, i_\ell$ eine Lösung für F und $\mathfrak{A} = \{A, c_\varepsilon, f_0, f_1, P)$ eine beliebige Struktur. Zu zeigen ist $\mathfrak{A} \models \varphi_F$. Wenn $\mathfrak{A} \not\models \varphi \wedge \psi$, dann gilt $\mathfrak{A} \models \varphi_F$. Nehmen wir nun also $\mathfrak{A} \models \varphi \wedge \psi$ an und zeigen $\mathfrak{A} \models \exists x\, P(x,x)$.

Obwohl $\mathfrak{A}$ nicht notwendigerweise kanonisch ist, können wir darin trotzdem die Lösung von F wiederfinden. Dazu definieren wir eine Abbildung $h : \{0,1\}^* \to A$, die jedem 0-1-Wort das zugehörige Element im Universum von $\mathfrak{A}$ zuordnet:

$$\begin{aligned} h(\varepsilon) &= c_\varepsilon^{\mathfrak{A}} \\ h(w0) &= f_0^{\mathfrak{A}}(h(w)) \ \text{ für alle } w \in \{0,1\}^* \\ h(w1) &= f_1^{\mathfrak{A}}(h(w)) \ \text{ für alle } w \in \{0,1\}^* \end{aligned}$$

Man sieht nun leicht, dass $h(w) = t_w^{\mathfrak{A}}(c_\varepsilon)$ für alle $w \in \{0,1\}^*$ gilt. Wegen $\mathfrak{A} \models \varphi$ gilt also

$$\big(h(u_{i_1}),\ h(v_{i_1})\big) \in P^{\mathfrak{A}}.$$

Wegen $\mathfrak{A} \models \psi$ können wir induktiv schließen, dass

$$\big(h(u_{i_1} \cdots u_{i_r}),\ h(v_{i_1} \cdots v_{i_r})\big) \in P^{\mathfrak{A}} \quad \text{für alle } r \leq \ell. \qquad (*)$$

Sei nun $u_{i_1} \cdots u_{i_\ell} = v_{i_1} \cdots v_{i_\ell} = w$ (da $i_1, \ldots, i_\ell$ eine Lösung für F ist). Dann gilt wegen $(*)$ also $(h(w),\, h(w)) \in P^{\mathfrak{A}}$ und damit $\mathfrak{A} \models \exists x\, P(x,x)$, was zu zeigen war. ❑

T2.21 Beispiel für endliche Erfüllbarkeit

Folgender FO-Satz ist erfüllbar, aber nicht endlich erfüllbar:

$$\forall x\, \neg R(x,c) \quad \wedge \quad \forall x\, \exists y\, R(x,y) \quad \wedge \quad \forall x\, \forall x'\, \forall y\, \big(R(x,y) \wedge R(x',y) \to x = x'\big)$$

Dieser Satz ist eine leichte Variation des 2. Beispiels in T2.14. Seine Teile sagen, dass

(1) das Element $c^{\mathfrak{A}}$ keinen „R-Vorgänger“ hat;

(2) jedes Element einen „R-Nachfolger“ hat;

(3) kein Element zwei „R-Vorgänger“ haben darf.

Damit dieser Satz erfüllbar ist, muss es mindestens ein Element $a_0 = c^{\mathfrak{A}}$ im Universum A geben. Wegen (2) muss es ein Element a_1 mit $R^{\mathfrak{A}}(a_0, a_1)$ geben. Wegen (1) muss $a_1 \neq a_0$ sein. Wegen (2) muss es ein Element a_2 mit $R^{\mathfrak{A}}(a_1, a_2)$ geben. Wegen (1) muss $a_2 \neq a_0$ sein; wegen (3) muss $a_2 \neq a_1$ sein. So kann man die Argumentation induktiv fortsetzen und erhält eine (abzählbar) unendliche Folge $a_0, a_1, a_2, \ldots$ von paarweise verschiedenen Elementen aus A.

T2.22 Beispiele für Theorien

1. Die Menge $\mathsf{Taut}(\tau)$ aller Tautologien in einer fixen Signatur τ ist
   - eine FO-Theorie:

     Wegen der Def. von Tautologien gilt: $\mathfrak{A} \models \mathsf{Taut}(\tau)$ für *jede* Struktur $\mathfrak{A}$ $(*)$

     Also ist
     - $\mathsf{Taut}(\tau)$ erfüllbar wegen $(*)$;
     - $\mathsf{Taut}(\tau)$ abgeschlossen unter Konsequenz: wenn $\mathsf{Taut}(\tau) \models \varphi$, dann ist φ Tautologie (wegen $(*)$ und der Definition von $\models$) und damit $\varphi \in \mathsf{Taut}(\tau)$.
   - nicht vollständig:

     Es gibt Sätze φ, die weder Tautologie sind noch unerfüllbar – also gilt weder $\varphi \in \mathsf{Taut}(\tau)$, noch $\neg\varphi \in \mathsf{Taut}(\tau)$. Finde selbst einen solchen Satz.

- enthalten in allen anderen Theorien:

  Jede Theorie enthält alle Tautologien, denn diese sind Konsequenzen *aller* Formelmengen (siehe Def. Tautologie bzw. Konsequenz).

2. Sei $\mathfrak{A}$ eine τ-Struktur. Dann ist

$$\mathsf{Th}(\mathfrak{A}) = \{\varphi \text{ ist } \tau\text{-Satz} \mid \mathfrak{A} \models \varphi\} \qquad (*)$$

   - eine FO-Theorie:
     - $\mathsf{Th}(\mathfrak{A})$ ist erfüllbar, denn wegen $(*)$ gilt $\mathfrak{A} \models \mathsf{Th}(\mathfrak{A})$. $(**)$
     - $\mathsf{Th}(\mathfrak{A})$ ist abgeschlossen unter Konsequenz, denn wenn $\mathsf{Th}(\mathfrak{A}) \models \varphi$, dann wegen $(**)$ auch $\mathfrak{A} \models \varphi$; also gilt wegen $(*)$: $\varphi \in \mathsf{Th}(\mathfrak{A})$.
   - vollständig:

     Für jede τ-Struktur $\mathfrak{A}$ und jeden τ-Satz φ gilt $\mathfrak{A} \models \varphi$ oder $\mathfrak{A} \models \neg\varphi$ (was leicht per strukturelle Induktion gezeigt werden kann).

3. Sei Ω eine erfüllbare Menge von FO-Sätzen. Dann ist

$$\mathsf{Abschluss}(\Omega) = \{\varphi \text{ ist } \tau\text{-Satz} \mid \Omega \models \varphi\}$$

   - eine FO-Theorie:
     - $\mathsf{Abschluss}(\Omega)$ ist erfüllbar, da Ω erfüllbar ist und alle Konsequenzen aus Ω in den Modellen von Ω ebenfalls wahr sind.
     - $\mathsf{Abschluss}(\Omega)$ ist aufgrund seiner Definition und der Transitivität der Konsequenz abgeschlossen unter Konsequenz: Wenn $\mathsf{Abschluss}(\Omega) \models \varphi$, dann bereits $\Omega \models \varphi$, also $\varphi \in \mathsf{Abschluss}(\Omega)$.
   - im Allgemeinen nicht vollständig:

     Für $\Omega = \emptyset$ beispielsweise ist $\mathsf{Abschluss}(\Omega) = \mathsf{Taut}(\tau)$, was wegen Punkt 1 nicht vollständig ist. Ist andererseits Ω selbst bereits eine vollständige Theorie, dann ist $\mathsf{Abschluss}(\Omega) = \Omega$ und damit vollständig.

4. Sei $\mathcal{K}$ eine Klasse von τ-Strukturen. Dann ist

$$\mathsf{Th}(\mathcal{K}) = \bigcap_{\mathfrak{A} \in \mathcal{K}} \mathsf{Th}(\mathfrak{A})$$

   - eine FO-Theorie:
     - $\mathsf{Th}(\mathcal{K})$ ist erfüllbar: wegen $(**)$ aus Punkt 2 gilt $\mathfrak{A} \models \mathsf{Th}(\mathcal{K})$ für alle $\mathfrak{A} \in \mathcal{K}$.
     - $\mathsf{Th}(\mathcal{K})$ abgeschlossen unter Konsequenz, weil es der Schnitt von unter Konsequenz abgeschlossenen Mengen ist: Wenn $\mathsf{Th}(\mathcal{K}) \models \varphi$, dann $\mathfrak{A} \models \varphi$ für alle $\mathfrak{A} \in \mathcal{K}$. Also $\varphi \in \mathsf{Th}(\mathfrak{A})$ wegen Punkt 2, und damit $\varphi \in \mathsf{Th}(\mathcal{K})$.
   - im Allgemeinen nicht vollständig:

     Wenn beispielweise $\mathcal{K}$ die Klasse *aller* τ-Strukturen ist, dann ist $\mathsf{Th}(\mathcal{K}) = \mathsf{Taut}(\tau)$, was wegen Punkt 1 nicht vollständig ist. Ist andererseits $\mathcal{K} = \{\mathfrak{A}\}$ eine einelementige Klasse, dann ist $\mathsf{Th}(\mathcal{K}) = \mathsf{Th}(\mathfrak{A})$ und wegen Punkt 2 vollständig.

T2.23 Beweis des Lemmas über Vollständigkeit von Theorien

Lemma 2.30. Sei Γ eine FO-Theorie. Dann sind die folgenden Aussagen äquivalent:

(1) Γ ist vollständig

(2) $\Gamma = \mathsf{Th}(\mathfrak{A})$ für eine Struktur $\mathfrak{A}$

(3) alle Modelle $\mathfrak{A}', \mathfrak{A}''$ von Γ sind elementar äquivalent

Beweis.

„(1) ⇒ (2)“.

Sei Γ vollständig. Da Γ als Theorie erfüllbar ist, gibt es ein Modell $\mathfrak{A}$ mit $\mathfrak{A} \models \Gamma$; also ist $\Gamma \subseteq \mathsf{Th}(\mathfrak{A})$. Es bleibt zu zeigen, dass die umgekehrte Inklusion $\mathsf{Th}(\mathfrak{A}) \subseteq \Gamma$ gilt. Sei also $\varphi \in \mathsf{Th}(\mathfrak{A})$. Da Γ vollständig ist, gilt $\varphi \in \Gamma$ oder $\neg\varphi \in \Gamma$. Letzteres ist aber unmöglich, weil $\mathfrak{A} \models \Gamma$. Also $\varphi \in \Gamma$.

„(2) ⇒ (3)“.

Sei $\Gamma = \mathsf{Th}(\mathfrak{A})$ und seien $\mathfrak{A}', \mathfrak{A}''$ Modelle von Γ. Sei $\mathfrak{A}' \models \varphi$. Zu zeigen ist $\mathfrak{A}'' \models \varphi$. Wegen $\mathfrak{A} \models \varphi$ oder $\mathfrak{A} \models \neg\varphi$ ist auch $\varphi \in \Gamma$ oder $\neg\varphi \in \Gamma$. Da $\mathfrak{A}' \models \Gamma$ und $\mathfrak{A}' \models \varphi$, ist $\neg\varphi \in \Gamma$ ausgeschlossen. Also ist $\varphi \in \Gamma$ und wegen $\mathfrak{A}'' \models \Gamma$ auch $\mathfrak{A}'' \models \varphi$.

„(3) ⇒ (1)“.

Seien alle Modelle von Γ elementar äquivalent und sei φ ein Satz. Zu zeigen ist: $\varphi \in \Gamma$ oder $\neg\varphi \in \Gamma$. Angenommen, keines von beiden ist der Fall. Dann gilt wegen der Abgeschlossenheit von Γ unter Konsequenz mittels Kontraposition: $\Gamma \not\models \varphi$ und $\Gamma \not\models \neg\varphi$. Also gibt es Modelle $\mathfrak{A}', \mathfrak{A}''$ von Γ mit $\mathfrak{A}' \models \neg\varphi$ und $\mathfrak{A}'' \models \varphi$. Dann sind aber $\mathfrak{A}'$ und $\mathfrak{A}''$ nicht elementar äquivalent; ein Widerspruch zur Annahme. ❑

Teil III.

Mehr zu Prädikatenlogik 1. Stufe

T3.1 Erklärungen zu den Schlussregeln des Sequenzenkalküls

Am besten liest man die Regeln „von oben nach unten“:

$$(\wedge \Rightarrow)\ \frac{\Gamma, \varphi, \psi \Rightarrow \Delta}{\Gamma, \varphi \wedge \psi \Rightarrow \Delta}$$

Wenn die obere Sequenz $\Gamma, \varphi, \psi \Rightarrow \Delta$ gültig ist, dann auch die untere Sequenz $\Gamma, \varphi \wedge \psi \Rightarrow \Delta$, denn ihre Antezedensen $\Gamma \cup \{\varphi, \psi\}$ und $\Gamma \cup \{\varphi \wedge \psi\}$ haben dieselben Modelle.

$$(\Rightarrow \wedge)\ \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}$$

Wenn die oberen beiden Sequenzen gültig sind, dann gilt:

$$\bigwedge \Gamma \models \Big(\bigvee (\Delta \cup \{\varphi\}) \wedge \bigvee (\Delta \cup \{\psi\}) \Big)$$

Nach dem Distributivgesetz[4] gilt dann auch:

$$\bigwedge \Gamma \models \bigvee (\Delta \cup \{\varphi \wedge \psi\})$$

$$(\vee \Rightarrow)\ \frac{\Gamma, \varphi \Rightarrow \Delta \quad \Gamma, \psi \Rightarrow \Delta}{\Gamma, \varphi \vee \psi \Rightarrow \Delta}$$

Diese Regel ist dual zu $(\Rightarrow \wedge)$ in dem Sinne, dass hier die Antezedensen so verändert werden wie dort die Sukzedensen, wobei “$\wedge$” durch “$\vee$” ersetzt wird.

Man kann hier auch wieder analog wie oben argumentieren, unter Verwendung des Distributivgesetzes (probiert es aus).

$$(\Rightarrow \vee)\ \frac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}$$

Diese Regel ist dual zu $(\wedge \Rightarrow)$ in dem Sinne, dass hier die Sukzedensen so verändert werden wie dort die Antezedensen, wobei “$\wedge$” durch “$\vee$” ersetzt wird.

Man kann hier auch wieder analog wie ganz oben argumentieren: die Sukzedensen $\Delta \cup \{\varphi, \psi\}$ und $\Delta \cup \{\varphi \vee \psi\}$ (als Disjunktionen aufgefasst!) haben dieselben Modelle.

$$(\neg \Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma, \neg\varphi \Rightarrow \Delta}$$

Wenn die obere Sequenz gültig ist, dann gilt:

$$\bigwedge \Gamma \models \bigvee (\Delta \cup \{\varphi\}) \qquad (*)$$

[4] Das dürfen wir anwenden, weil wir *endliche* Formelmengen betrachten und deshalb alle Konjunktionen bzw. Disjunktionen (endliche) FO-Formeln sind.

Um zu zeigen, dass daraus

$$\bigwedge(\Gamma \cup \{\neg\varphi\}) \models \bigvee \Delta \qquad (**)$$

folgt, betrachten wir ein beliebiges Modell $\mathfrak{A} \models \bigwedge(\Gamma \cup \{\neg\varphi\})$. Insbesondere haben wir also $\mathfrak{A} \models \bigwedge \Gamma$ und $\mathfrak{A} \models \neg\varphi$ (Semantik der Konjunktion). Wegen $\mathfrak{A} \models \bigwedge \Gamma$ und $(*)$ gilt $\mathfrak{A} \models \bigvee(\Delta \cup \{\varphi\})$. Zusammen mit $\mathfrak{A} \models \neg\varphi$ folgt $\mathfrak{A} \models \bigvee \Delta$. Also ist $(**)$ gültig.

$$(\Rightarrow \neg)\ \frac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}$$

Wenn die obere Sequenz gültig ist, dann gilt:

$$\bigwedge(\Gamma \cup \{\varphi\}) \models \bigvee \Delta \qquad (*)$$

Um zu zeigen, dass daraus

$$\bigwedge \Gamma \models \bigvee(\Delta \cup \{\neg\varphi\}) \qquad (**)$$

folgt, betrachten wir ein beliebiges Modell $\mathfrak{A} \models \bigwedge \Gamma$. Wenn $\mathfrak{A} \models \varphi$, dann folgt wegen $(*)$, dass $\mathfrak{A} \models \bigvee \Delta$, also auch $\mathfrak{A} \models \bigvee(\Delta \cup \{\neg\varphi\})$. Wenn $\mathfrak{A} \models \neg\varphi$, dann gilt sowieso $\mathfrak{A} \models \bigvee(\Delta \cup \{\neg\varphi\})$. Also ist $(**)$ gültig.

$$(\exists \Rightarrow)\ \frac{\Gamma, \varphi[c] \Rightarrow \Delta}{\Gamma, \exists x\, \varphi(x) \Rightarrow \Delta}$$ wenn die Konstante c nirgends in $\Gamma, \Delta, \varphi(x)$ vorkommt.

Intuitiv besagt diese Regel, dass man eine Konstante, die sonst nirgends vorkommt, auch durch ein „anonymes Objekt“ ersetzen darf.

Genauer: Wenn die obere Sequenz gültig ist, dann gilt:

$$\bigwedge(\Gamma \cup \{\varphi[c]\}) \models \bigvee \Delta \qquad (*)$$

Um zu zeigen, dass daraus

$$\bigwedge(\Gamma \cup \{\exists x\, \varphi(x)\}) \models \bigvee \Delta \qquad (**)$$

folgt, betrachten wir ein beliebiges Modell $\mathfrak{A} \models \bigwedge(\Gamma \cup \{\exists x\, \varphi(x)\})$. Insbesondere gilt $\mathfrak{A} \models \exists x\, \varphi(x)$, also gibt es ein $a \in A$ mit $\mathfrak{A} \models \varphi[a]$. Sei $\mathfrak{A}'$ die Struktur, die man aus $\mathfrak{A}$ erhält, indem man zusätzlich $c^{\mathfrak{A}'} = a$ setzt. Dann gilt:

- $\mathfrak{A}' \models \varphi[c]$ (weil c nicht in $\varphi(x)$ vorkommt, kann sich der Wahrheitswert von φ durch die Transformation auch nicht ändern) und
- und $\mathfrak{A}' \models \Gamma$ (weil c nicht in Γ vorkommt).

Also $\mathfrak{A}' \models \bigwedge(\Gamma \cup \{\varphi[c]\})$. Wegen $(*)$ erhalten wir $\mathfrak{A}' \models \bigvee \Delta$. Weil c nicht in Δ vorkommt, folgt $\mathfrak{A} \models \bigvee \Delta$. Also ist $(**)$ gültig.

Diese Argumentation benutzt die Seitenbedingung „wenn die Konstante c nirgends in $\Gamma, \Delta, \varphi(x)$ vorkommt“. Ohne diese Bedingung wäre die Regel nicht korrekt, was man am besten an konkreten Beispielen sieht:

$$\frac{P(c), Q(c) \;\Rightarrow\; P(c) \wedge Q(c)}{P(c), \exists x\, Q(x) \;\Rightarrow\; P(c) \wedge Q(c)} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist nicht gültig}\end{array}$$

$$\frac{P(c) \wedge Q(c) \;\Rightarrow\; P(c)}{\exists x\,(P(x) \wedge Q(x)) \;\Rightarrow\; P(c)} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist nicht gültig}\end{array}$$

($\Rightarrow \exists$) $\dfrac{\Gamma \Rightarrow \Delta, \varphi[t]}{\Gamma \Rightarrow \Delta, \exists x\, \varphi(x)}$ wobei t ein beliebiger Term ist

Intuitiv besagt diese Regel: wenn $\varphi[t]$ für ein konkretes Element t *impliziert wird*, dann auch für ein beliebiges, existentiell quantifiziertes Objekt. Probiert die formale Argumentation gern selbst aus!

Die Seitenbedingung wird nicht gebraucht; insbesondere sind die obigen Beispiele hier keine Gegenbeispiele:

$$\frac{P(c), Q(c) \;\Rightarrow\; P(c) \wedge Q(c)}{P(c), Q(c) \;\Rightarrow\; \exists x\,(P(x) \wedge Q(x))} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist auch gültig}\end{array}$$

$$\frac{P(c) \wedge Q(c) \;\Rightarrow\; P(c)}{P(c) \wedge Q(c) \;\Rightarrow\; \exists x\, P(x)} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist auch gültig}\end{array}$$

($\forall \Rightarrow$) $\dfrac{\Gamma, \varphi[t] \Rightarrow \Delta}{\Gamma, \forall x\, \varphi(x) \Rightarrow \Delta}$ wobei t ein beliebiger Term ist

Dual zu ($\Rightarrow \exists$).

($\Rightarrow \forall$) $\dfrac{\Gamma \Rightarrow \Delta, \varphi[c]}{\Gamma \Rightarrow \Delta, \exists x\, \varphi(x)}$ wenn die Konstante c nirgends in $\Gamma, \Delta, \varphi(x)$ vorkommt.

Dual zu ($\exists \Rightarrow$).

Anmerkung zu den Seitenbedingungen von $(\exists \Rightarrow)$ und $(\Rightarrow \forall)$

Diese Anmerkung ist nicht wesentlich fürs Verständnis des Sequenzenkalküls **und kann getrost übersprungen werden.** Ich hatte sie aufgenommen, als im WiSe 2016/17 die Frage aufkam, ob die Seitenbedingungen „c nicht in $\Gamma, \Delta, \varphi(x)$“ der Regeln $(\exists \Rightarrow)$ und $(\Rightarrow \forall)$ abgeschwächt werden können.

Im obigen Beweis der Korrektheit von $(\exists \Rightarrow)$ wird verwendet, dass c *weder* in Γ, *noch* in Δ, *noch* in $\varphi(x)$ vorkommt. Im vorangehenden Beispiel für die Inkorrektheit der Regel ohne die Seitenbedingung kommt c *sowohl* in Γ *als auch* in Δ *als auch* in $\varphi(x)$ vor. Es gibt aber auch Beispiele, in denen die Regel inkorrekt ist, wenn c *nur in* Γ oder *nur in* Δ oder *nur in* $\varphi(x)$ vorkommt:

- nur in Δ:

  Betrachte $\Gamma = \emptyset$, $\Delta = P(c)$, $\varphi(x) = P(x)$; also $\varphi[c] = P(c)$.

  $$\dfrac{P(c) \;\Rightarrow\; P(c)}{\exists x\, P(c) \;\Rightarrow\; P(c)} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist nicht gültig}\end{array}$$

  Für die ungültige Sequenz in der unteren Zeile betrachte man $\mathfrak{A}$ mit $A = \{a, b\}$, $c^{\mathfrak{A}} = a$ und $P^{\mathfrak{A}} = \{b\}$.

- nur in Γ:

  Betrachte $\Gamma = \neg P(c)$, $\Delta = \emptyset$, $\varphi(x) = P(x)$; also $\varphi[c] = P(c)$.

  $$\dfrac{\neg P(c), P(c) \;\Rightarrow\; \emptyset}{\neg P(c), \exists x\, P(x) \;\Rightarrow\; \emptyset} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist nicht gültig}\end{array}$$

  Für die gültige Sequenz in der oberen Zeile beobachte man, dass das Antezedens unerfüllbar ist; für die ungültige Sequenz in der unteren Zeile betrachte man dasselbe Modell $\mathfrak{A}$ wie oben.

- nur in $\varphi(x)$:

  Betrachte $\Gamma = \Delta = \emptyset$, $\varphi(x) = P(x) \wedge \neg P(c)$; also $\varphi[c] = P(c) \wedge \neg P(c)$.

  $$\dfrac{P(c) \wedge \neg P(c) \;\Rightarrow\; \emptyset}{\exists x\, (P(x) \wedge \neg P(c)) \;\Rightarrow\; \emptyset} \qquad \begin{array}{l}\text{ist gültig}\\ \text{ist nicht gültig}\end{array}$$

  Die Argumentation für (Un)Gültigkeit ist dieselbe wie im letzten Fall.

T3.2 Beispiel für eine ableitbare Sequenz

- $P(c), Q(c) \;\Rightarrow\; P(c), R(c)$ ist ein Axiom, weil $P(c)$ auf beiden Seiten auftritt.
- $P(c), Q(c) \;\Rightarrow\; Q(c), R(c)$ ist ein Axiom, weil $Q(c)$ auf beiden Seiten auftritt.
- Setzt man nun $\Gamma = \{P(c), Q(c)\}$ und $\Delta = \{R(c)\}$, so erhält man mit der Regel $(\Rightarrow \wedge)$ die Sequenz:
  $P(c), Q(c) \;\Rightarrow\; P(c) \wedge Q(c), R(c)$

T3.3 Beispiele für SK-Beweise

Man beginnt mit der zu beweisenden Sequenz und wendet Regeln *von unten nach oben* an, bis man Axiome erhält. Man lese die folgenden Beweise also *von unten nach oben!*

1. Für beliebige Formeln φ, ψ ist folgendes ein SK-Beweis:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi, \psi}{\varphi \Rightarrow \varphi \vee \psi}(\Rightarrow \vee)}{\neg(\varphi \vee \psi), \varphi \Rightarrow \emptyset}(\neg \Rightarrow)}{\neg(\varphi \vee \psi) \Rightarrow \neg\varphi}(\Rightarrow \neg) \qquad \dfrac{\dfrac{\dfrac{\psi \Rightarrow \varphi, \psi}{\psi \Rightarrow \varphi \vee \psi}(\Rightarrow \vee)}{\neg(\varphi \vee \psi), \psi \Rightarrow \emptyset}(\neg \Rightarrow)}{\neg(\varphi \vee \psi) \Rightarrow \neg\psi}(\Rightarrow \neg)}{\neg(\varphi \vee \psi) \Rightarrow \neg\varphi \wedge \neg\psi}(\Rightarrow \wedge)$$

2. Ein SK-Beweis, der nur Quantoren-Regeln verwendet:

$$\dfrac{\dfrac{\dfrac{\dfrac{R(c,d) \Rightarrow R(c,d)}{R(c,d) \Rightarrow \exists x\, R(x,d)}(\Rightarrow \exists)}{\forall y\, R(c,y) \Rightarrow \exists x\, R(x,d)}(\forall \Rightarrow)}{\forall y\, R(c,y) \Rightarrow \forall y \exists x\, R(x,y)}(\Rightarrow \forall)}{\exists x \forall y\, R(x,y) \Rightarrow \forall y \exists x\, R(x,y)}(\exists \Rightarrow)$$

   Bei den unteren beiden Schritten muss die Seitenbedingung eingehalten werden; bei den oberen beiden Schritten nicht – deshalb dürfen wir hier die bereits benutzten Terme (Konstanten) c und d „einführen“. Es kommt also in diesem Beispiel auf die Reihenfolge der Regelanwendung an. Wenn wir die zwei letzten (obersten) Regeln $(\forall \Rightarrow)$ und $(\Rightarrow \exists)$ zuerst (zuunterst) anwenden würden, hätten wir bereits c und d eingeführt und dürften mit den anderen beiden Regeln wegen ihrer Seitenbedingung kein weiteres c bzw. d einführen.

3. Ein SK-Beweis, der Formeln im Antezedens „behält“ (siehe Folie 11):

$$\dfrac{\dfrac{\dfrac{P(c), P(d) \Rightarrow P(c) \qquad P(c), P(d) \Rightarrow P(d)}{P(c), P(d) \Rightarrow P(c) \wedge P(d)}(\Rightarrow \wedge)}{\forall x\, P(x), P(c) \Rightarrow P(c) \wedge P(d)}(\forall \Rightarrow)}{\forall x\, P(x) \Rightarrow P(c) \wedge P(d)}(\forall \Rightarrow)$$

   Im untersten Schritt wendet man also die Regel $(\forall \Rightarrow)$ auf $\forall x\, P(x)$ an, ohne dieses zu löschen. Das ist zugelassen, denn „$\Gamma, \varphi \Rightarrow \Delta, \psi$“ schließt auch den Fall $\varphi \in \Gamma$ ein. In diesem Beispiel kann man zwar auf das Behalten von $(\forall \Rightarrow)$ verzichten, wenn man die letzte (oberste) Regel $(\Rightarrow \wedge)$ zuerst (zuunterst) anwendet; für das Ableiten anderer Sequenzen ist das Behalten aber essentiell. Dies ist im übrigen auch der Grund, warum die Länge von SK-Beweisen *nicht* exponentiell in der Länge der zu beweisenden Sequenz beschränkt ist.

T3.4 Beweis der Korrektheit des SK

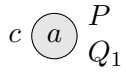
Nach den Bemerkungen auf Folie 12 genügt es zu zeigen, dass jede einzelne SK-Regel korrekt ist, d. h. wenn die obere(n) Sequenz(en) gültig ist/sind, dann auch die untere. Dies haben wir jedoch bereits in T3.1 für die einzelnen Regeln gezeigt.

T3.5 Beispiele für die Vervollständigung von Γ

Der Beweis der Vollständigkeit des SK wird mittels Kontraposition geführt. Wir nehmen also an, $\Gamma \Rightarrow \Delta$ sei *nicht ableitbar*. Das Ziel ist zu zeigen, dass es ein Modell $\mathfrak{A} \models \Gamma \cup \neg\Delta$ gibt, wobei $\neg\Delta = \{\neg\varphi \mid \varphi \in \Delta\}$. Dieses Modell $\mathfrak{A}$ möchten wir gern aus Γ „ablesen“.

(1) Wenn $\Gamma = \{Q_1(c), \neg Q_2(c), \exists x\, P(x), P(c)\}$ und $\Delta = \{Q_2(c), \neg P(c)\}$ wie auf Folie 14, dann ist klar, wie man $\mathfrak{A}$ aus Γ ablesen kann und dass $\mathfrak{A} \models \neg\Delta$:

$$\begin{aligned} A &= \{a\} \\ c^{\mathfrak{A}} &= a \\ P^{\mathfrak{A}} &= \{a\} \\ Q_1^{\mathfrak{A}} &= \{a\} \\ Q_2^{\mathfrak{A}} &= \emptyset \end{aligned}$$



(2) Sei nun $\Gamma = \{Q_1(c) \vee Q_2(c), \exists x\, P(x)\}$ und $\Delta = \{\neg\exists x\, Q_2(x), P(c), \dots\}$. Dann ist $\mathfrak{A}$ durch Γ nicht eindeutig bestimmt, denn

   (a) es gibt zwei Möglichkeiten die Disjunktion $Q_1(c) \vee Q_2(c)$ zu erfüllen;

   (b) es muss ein konkretes Element benannt werden, das $\exists x\, P(x)$ „bezeugt“;

   (c) damit $\mathfrak{A} \models \neg\Delta$, muss ein Element benannt werden, das $\exists x\, Q_2(x)$ „bezeugt“.

   Bevor man also $\mathfrak{A}$ aus Γ wie gewünscht ablesen kann, muss man Γ schrittweise wie folgt erweitern:

   (a) Man beobachte, dass eine der zwei folgenden Sequenzen nicht ableitbar sein kann:

$$\Gamma \cup \{Q_1(c)\} \;\Rightarrow\; \Delta \qquad (i)$$
$$\Gamma \cup \{Q_2(c)\} \;\Rightarrow\; \Delta \qquad (ii)$$

   Wären nämlich beide ableitbar, dann auch $\Gamma \Rightarrow \Delta$ (was im Widerspruch zur Annahme steht):

$$\frac{\Gamma \cup \{Q_1(c)\} \Rightarrow \Delta \qquad \Gamma \cup \{Q_2(c)\} \Rightarrow \Delta}{\Gamma \cup \{Q_1(c) \vee Q_2(c)\} \Rightarrow \Delta}\ (\vee\Rightarrow)$$

   Das Antezedens der unteren Sequenz ist dabei gleich Γ, weil $Q_1(c) \vee Q_2(c)$ bereits in Γ enthalten ist. Je nachdem, welche der Sequenzen (i) oder (ii) nicht

ableitbar ist, wird nun Γ entsprechend erweitert. Wir nehmen o. B. d. A. an, dass (i) nicht ableitbar ist, also erweitern wir Γ zu

$$\Gamma := \{Q_1(c) \vee Q_2(c), \exists x\, P(x), Q_1(c)\}.$$

(b) Wenn wir die Formel $\exists x\, P(x)$ aus Γ betrachten, müssen wir ein Element „festlegen“, welches in $\mathfrak{A}$ dafür sorgt, dass $\mathfrak{A} \models \exists x\, P(x)$. Dazu führen wir eine neue Konstante c_{neu} ein, die weder in Γ, noch in Δ auftritt, und beobachten, dass $\Gamma \cup \{P(c_{\mathsf{neu}})\} \Rightarrow \Delta$ nicht ableitbar sein kann:

$$\frac{\Gamma \cup \{P(c_{\mathsf{neu}})\} \Rightarrow \Delta}{\Gamma \cup \{\exists x\, P(x)\} \Rightarrow \Delta} \quad (\exists \Rightarrow)$$

Die untere Sequenz besagt aber wieder, dass $\Gamma \Rightarrow \Delta$ ableitbar wäre, was im Widerspruch zur Annahme steht. Wir erweitern also Γ zu

$$\Gamma := \{Q_1(c) \vee Q_2(c), \exists x\, P(x), Q_1(c), P(c_{\mathsf{neu}})\}.$$

(c) Nun ist zwar $\mathfrak{A}$ im Sinne von Beispiel (1) durch Γ eindeutig bestimmt, aber $\mathfrak{A}$ macht noch nicht $\exists x\, Q_2(x) \in \neg\Delta$ wahr. Man muss also Γ noch mehr erweitern: Wegen $\neg\exists x\, Q_2(x) \in \Delta$ ist $\Gamma \cup \{\exists x\, Q_2(x)\} \Rightarrow \Delta$ nicht ableitbar – denn wenn es das wäre, dann mittels $(\Rightarrow \neg)$ auch $\Gamma \Rightarrow \Delta$. Also wird Γ wieder entsprechend erweitert, und es sind wegen der neu hinzugekommenen Formel $\exists x\, Q_2(x)$ mehr Erweiterungen notwendig ...

T3.6 Beweis des Kompaktheitssatzes

Theorem 3.10. Für alle Mengen von Sätzen $\Gamma \subseteq \mathsf{FO}$ und Sätze $\varphi \in \mathsf{FO}$ gilt:

(1) Γ ist erfüllbar gdw. jede endliche Teilmenge von Γ erfüllbar ist.

(2) $\Gamma \models \varphi$ gdw. endliches $\Gamma_{\mathsf{f}} \subseteq \Gamma$ existiert mit $\Gamma_{\mathsf{f}} \models \varphi$.

Beweis. Da (1) mit den üblichen semantischen Beziehungen aus (2) folgt (siehe Fragebogen), beschränken wir uns darauf, (2) zu zeigen.

„⇐“. Folgt unmittelbar, denn jedes Modell von Γ ist ein Modell jeder Teilmenge von Γ, und nach Definition der Konsequenz ist dann auch jede Konsequenz von Γ eine Konsequenz jeder beliebigen Teilmenge von Γ.

„⇒“. Gelte $\Gamma \models \varphi$. Dann ist die Sequenz $\emptyset \Rightarrow \{\varphi\}$ aus der Formelmenge Γ folgerbar, d. h. $\Gamma \models \emptyset \Rightarrow \{\varphi\}$ (s. Folie 25). Wegen der Vollständigkeit der Γ-Erweiterung des SK (Theorem 3.11) ist die Sequenz $\emptyset \Rightarrow \Delta$ in der Γ-Erweiterung ableitbar. Jeder SK-Beweis ist jedoch endlich, und dies gilt auch für die Γ-Erweiterung. Betrachte also einen endlichen SK-Beweis B für $\emptyset \Rightarrow \Delta$ in der Γ-Erweiterung. Dieser Beweis kann nur endlich oft die Γ-Regel anwenden und dabei nur endlich viele Elemente $\varphi \in \Gamma$ verwenden. Sei Γ_{f} die Menge aller verwendeten $\varphi \in \Gamma$. Dann ist B auch ein SK-Beweis in der Γ_{f}-Erweiterung. Wegen der Korrektheit der Γ_{f}-Erweiterung des SK folgt nun $\Gamma_{\mathsf{f}} \models \emptyset \Rightarrow \{\varphi\}$, also $\Gamma_{\mathsf{f}} \models \varphi$. ❑

T3.7 Beweis des Satzes über unbeschränkte endliche Modelle

Theorem 3.12. Wenn ein FO-Satz φ beliebig große endliche Modelle besitzt (d. h. für jedes $n \geq 0$ gibt es Modell $\mathfrak{A}$ mit $|A| \geq n$), dann hat φ auch ein unendliches Modell.

Beweis. Sei φ ein FO-Satz, der beliebig große Modelle besitzt. Setze

$$\begin{aligned} \Gamma &= \{\varphi\} \cup \{\psi_n \mid n > 0\}, \quad \text{wobei} \\ \psi_n &= \exists x_1 \cdots \exists x_n \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j \quad \text{für alle } n > 0. \end{aligned}$$

Die Sätze ψ_n besagen also: „Das Modell hat die Größe $\geq n$“; genauer: jede Formel ψ_n ist genau in denjenigen Strukturen erfüllt, deren Universum mindestens n Elemente hat.

Um zu zeigen, dass φ ein unendliches Modell hat, genügt es demnach zu zeigen, dass Γ erfüllbar ist (denn wegen der ψ_n müssen die Modelle von Γ dann mehr Elemente haben als jede natürliche Zahl n). Wegen Kompaktheit (Thm. 3.10) genügt es, die Erfüllbarkeit jeder endlichen Teilmenge $\Gamma_\mathsf{f} \subseteq \Gamma$ zu zeigen. Betrachte ein solches Γ_f. Dann kommen darin auch nur endlich viele der ψ_n vor. Sei n die größte Zahl mit $\psi_n \in \Gamma_\mathsf{f}$. Nach Annahme gibt es ein Modell $\mathfrak{A}$ von φ mit $|A| \geq n$. Offensichtlich ist $\mathfrak{A}$ auch ein Modell von Γ_f. ❏

(fakultativ) Beweis Satz von Löwenheim-Skolem, aufsteigend

Theorem 3.13. Wenn ein FO-Satz φ ein unendliches Modell besitzt, dann gibt es für jede Menge U ein Modell $\mathfrak{A}$ von φ mit $|A| \geq |U|$.

Anmerkung. Die Menge U wird in der Formulierung des Satzes nur benötigt, um auszudrücken, dass es „beliebig große unendliche Modelle gibt“. Genauer heißt das: für jede Kardinalität κ gibt es ein Modell, das mindestens κ viele Elemente hat. Da nach dem Satz von Cantor (Diagonalisierung!) die Potenzmenge jeder Menge M mächtiger ist als M selbst, gibt es unendlich viele Kardinalitäten (unendlicher) Mengen. Diese werden alle durch die beliebige Menge U repräsentiert.

Beweis. Habe φ ein unendliches Modell $\mathfrak{A}$ und sei U eine Menge. Sei $\{c_u \mid u \in U\}$ eine Menge von paarweise verschiedenen Konstanten, die nicht in φ vorkommen. Setze

$$\Gamma = \{\varphi\} \cup \{c_u \neq c_v \mid u, v \in U,\, u \neq v\}. \qquad (*)$$

Es genügt zu zeigen, dass Γ erfüllbar ist (denn dann muss wegen der $c_u \neq c_v$ ein Modell von Γ mindestens $|U|$ Elemente haben). Wegen des Kompaktheitssatzes ist es ausreichend, die Erfüllbarkeit jeder endlichen Teilmenge $\Gamma_\mathsf{f} \subseteq \Gamma$ zu zeigen. Betrachte ein solches Γ_f. Sei

$$C = \{c_u \mid u \in U \text{ und } c_u \text{ kommt in } \Gamma_\mathsf{f} \text{ vor}\}.$$

Da C endlich ist, aber das obige Modell $\mathfrak{A}$ von φ ein unendliches Universum A hat, gibt es eine injektive Abbildung $\delta : C \to A$. Sei nun $\widehat{\mathfrak{A}}$ die Struktur, die man aus $\mathfrak{A}$ erhält, indem man zusätzlich setzt:

$$c_u^{\widehat{\mathfrak{A}}} = \delta(c_u) \quad \text{für alle } c_u \in C$$

Offensichtlich gilt $\widehat{\mathfrak{A}} \models \Gamma_\mathsf{f}$. ❏

(fakultativ) Beweis Satz von Löwenheim-Skolem, absteigend

Theorem 3.14. Wenn ein FO-Satz φ ein Modell besitzt, dann hat φ auch ein endliches oder abzählbar unendliches Modell.

Beweis. Hier können wir den (in der Vorlesung nur skizzierten) Vollständigkeitsbeweis des SK verwenden:

Sei φ erfüllbar. Dann ist die Sequenz $\{\varphi\} \Rightarrow \emptyset$ *nicht* gültig. Wegen der Korrektheit des (nicht erweiterten) SK ist diese Sequenz auch nicht ableitbar. Im Vollständigkeitsbeweis des SK wird für jede nicht ableitbare Sequenz $\Gamma \Rightarrow \Delta$ ein endliches oder abzählbar unendliches Modell für $\Gamma \cup \neg\Delta$, also für

$$\bigwedge \Gamma \wedge \bigwedge_{\varphi \in \Delta} \neg\varphi$$

konstruiert, in diesem Fall also für φ. ❏

T3.8 Nicht-Ausdrückbarkeit Zusammenhang via Kompaktheit

Theorem 3.16. Zusammenhang von ungerichteten Graphen ist *nicht* FO-ausdrückbar.

Beweis. Sei $\tau = \{E\}$. Angenommen, es gebe einen Satz $\varphi \in \mathsf{FO}(\tau)$, der Zusammenhang ausdrückt. Seien c_1, c_2 Konstantensymbole. Für $n \geq 0$ definieren wir Formeln ψ_n, die ausdrücken sollen, dass es *keinen* Pfad der Länge n zwischen c_1 und c_2 gibt, d. h.:

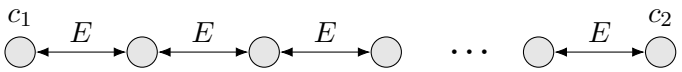
$$\psi_n = \neg\Big(\exists x_0 \ldots \exists x_n \Big(c_1 = x_0 \ \wedge \ c_2 = x_n \ \wedge \bigwedge_{0 \leq i < n} E(x_i, x_{i+1})\Big)\Big)$$

Wir setzen

$$\Gamma = \{\varphi\} \ \cup \ \{\forall x \forall y \, (E(x,y) \to E(y,x))\} \ \cup \ \{\psi_n \mid n \geq 0\}.$$

Behauptung. Γ ist erfüllbar.

Beweis der Behauptung. Wegen des Kompaktheitssatzes ist es ausreichend, die Erfüllbarkeit jeder *endlichen* Teilmenge $\Gamma_\mathsf{f} \subseteq \Gamma$ zu zeigen. Für ein beliebiges solches Γ_f sei m maximal mit $\psi_m \in \Gamma_\mathsf{f}$. Dann ist die folgende Struktur mit einem Pfad der Länge $m+1$ zwischen c_1 und c_2 ein Modell von Γ_f:



Da nun Γ ein Modell $\mathfrak{A}$ hat, muss insbesondere $\mathfrak{A} \models \varphi$ gelten; somit ist der Graph $(A, E^\mathfrak{A})$ zusammenhängend. Da außerdem $\mathfrak{A} \models \psi_n$ für alle $n \geq 0$, gibt es jedoch keinen Pfad von $c_1^\mathfrak{A} \in A$ zu $c_2^\mathfrak{A} \in A$; ein Widerspruch. ❏

T3.9 Keine Kompaktheit auf endlichen Strukturen

Der Kompaktheitssatz für endliche Strukturen würde lauten (Variante für Erfüllbarkeit):

> Eine Menge Γ von FO-Sätzen ist *endlich* erfüllbar gdw. jede endliche Teilmenge von Γ *endlich* erfüllbar ist.

Dabei bedeutet „endlich erfüllbar“, dass die jeweilige Menge ein *endliches* Modell $\mathfrak{A}$ hat.

Diese Aussage lässt sich jedoch wie folgt widerlegen. Betrachte dazu:

$$\psi_n = \exists x_1, \ \ldots \exists x_n \bigwedge_{i \neq j} x_i \neq x_j$$

$$\Gamma = \{\psi_n \mid n \geq 0\}$$

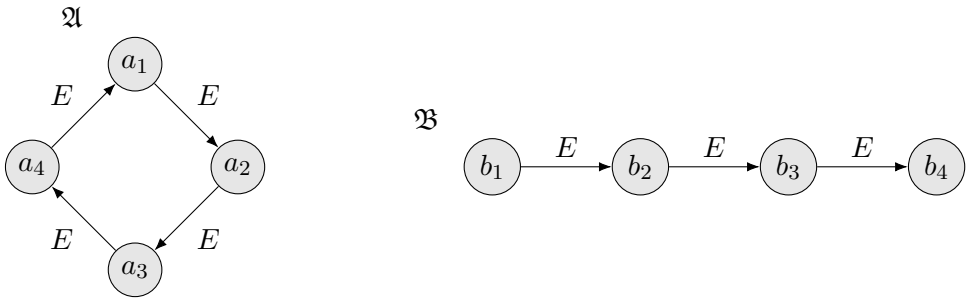
Jede endliche Teilmenge $\Gamma_\mathsf{f} \subseteq \Gamma$ ist endlich erfüllbar: wähle dazu eine beliebige Struktur $\mathfrak{A}$ mit

$$|A| = \mathsf{max}\{n \mid \psi_n \in \Gamma_\mathsf{f}\}.$$

Offensichtlich ist aber Γ nicht endlich erfüllbar.

T3.10 Beispiel eines Ehrenfeucht-Fraïssé-Spiels

Betrachte die Signatur $\tau = \{E\}$ für ein binäres Relationssymbol E, also die Signatur von (gerichteten) Graphen. Das Spielbrett bestehe aus folgenden Strukturen.



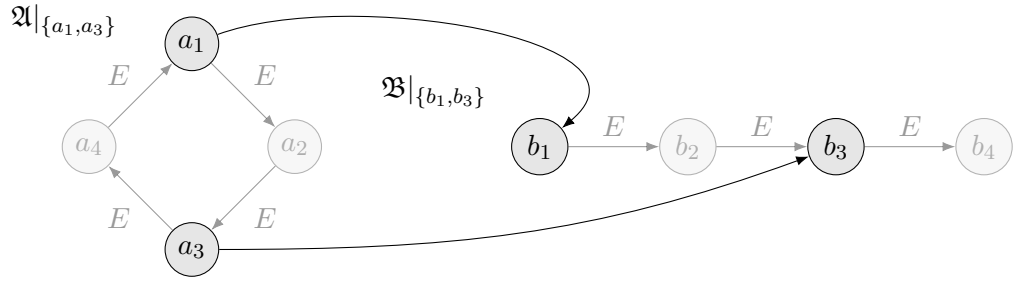
Ein möglicher Spielverlauf ist:

Runde	*Spoiler*	*Duplicator*
1	a_1	b_2
2	b_1	a_4
3	a_3	b_3

T3.11 Beispiele für partielle Isomorphismen

Betrachte die Strukturen $\mathfrak{A}, \mathfrak{B}$ aus dem vorangehenden Beispiel. Dann ist

- $\delta_1 : \begin{cases} a_1 \mapsto b_1 \\ a_3 \mapsto b_3 \end{cases}$ ein partieller Isomorphismus von $\mathfrak{A}|_{\{a_1,a_3\}}$ nach $\mathfrak{B}|_{\{b_1,b_3\}}$:



- $\delta_2 : \begin{cases} a_1 \mapsto b_2 \\ a_2 \mapsto b_3 \end{cases}$ ein partieller Isomorphismus
- $\delta_3 : \begin{cases} a_1 \mapsto b_2 \\ a_3 \mapsto b_3 \end{cases}$ *kein* partieller Isomorphismus
- $\delta_4 : \begin{cases} a_1 \mapsto b_1 \\ a_3 \mapsto b_1 \end{cases}$ *kein* partieller Isomorphismus

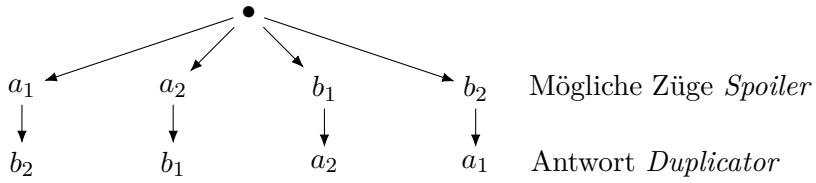
T3.12 Beispiele für Gewinnstrategien in EF-Spielen

Wir verwenden weiterhin die Signatur $\tau = \{E\}$ für ein binäres Relationssymbol E, also die Signatur von (gerichteten) Graphen.

Beispiel 1. Betrachte die folgenden Strukturen.

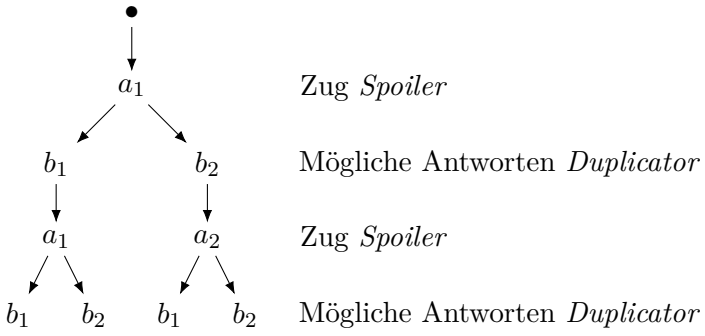
Dann können wir beobachten:

- *Duplicator* gewinnt $\mathcal{G}_0(\mathfrak{A}, \mathfrak{B})$. (Die „leere“ Abbildung ist auch ein partieller Isomorphismus, denn sie kann die Eigenschaften eines Isomorphismus nicht verletzen.)
- *Duplicator* hat eine Gewinnstrategie für $\mathcal{G}_1(\mathfrak{A}, \mathfrak{B})$, die sich wie folgt als Spielbaum darstellen lässt:



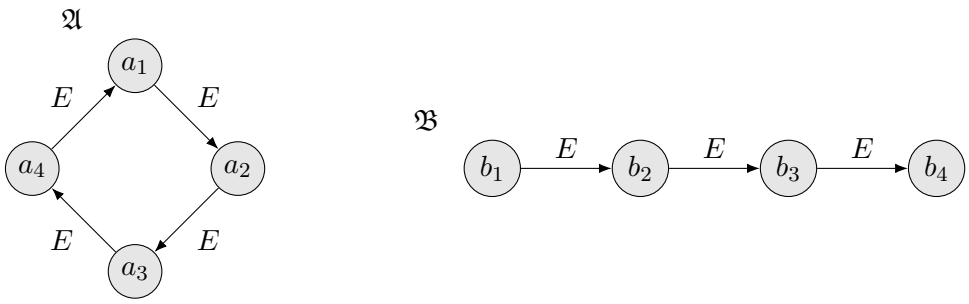
Dabei definiert jeder Pfad einen partiellen Isomorphismus, z. B. $\{a_1 \mapsto b_2\}$.

- *Spoiler* hat eine Gewinnstrategie für $\mathcal{G}_2(\mathfrak{A}, \mathfrak{B})$, die sich wie folgt als Spielbaum darstellen lässt:



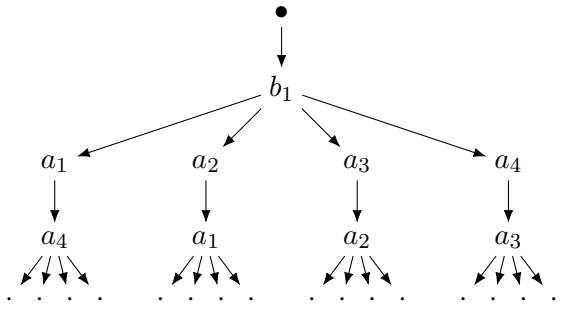
Dabei definiert *kein* Pfad einen partiellen Isomorphismus, z. B. $\{a_1 \mapsto b_2,\ a_2 \mapsto b_2\}$.

Beispiel 2. Betrachte die Strukturen aus den vergangenen beiden Beispielen:



Dann können wir beobachten:

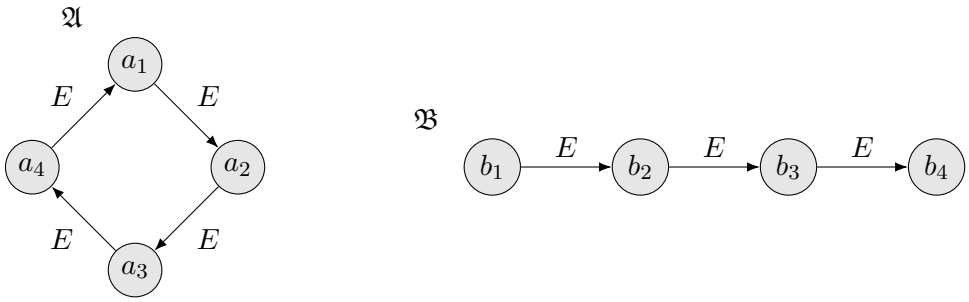
- *Duplicator* hat eine Gewinnstrategie für $\mathcal{G}_0(\mathfrak{A}, \mathfrak{B})$ und $\mathcal{G}_1(\mathfrak{A}, \mathfrak{B})$ (baue sie selbst).
- *Spoiler* hat eine Gewinnstrategie für $\mathcal{G}_2(\mathfrak{A}, \mathfrak{B})$, die sich wie folgt als Spielbaum darstellen lässt:



Wieder definiert *kein* Pfad einen partiellen Isomorphismus.

T3.13 Beispiele für das EF-Theorem

Betrachte wieder die Strukturen aus dem letzten Beispiel:



- Da *Duplicator* eine Gewinnstrategie für $\mathcal{G}_1(\mathfrak{A}, \mathfrak{B})$ hat, gilt wegen des EF-Theorems *für alle* FO-Sätze φ mit $\mathsf{qr}(\varphi) = 1$:

  $$\mathfrak{A} \models \varphi \quad \text{gdw.} \quad \mathfrak{B} \models \varphi,$$

  also z. B. für die Sätze

  $$\exists x\, E(x, x), \quad \forall x\, (x = x), \quad \ldots$$

- Da *Spoiler* eine Gewinnstrategie für $\mathcal{G}_2(\mathfrak{A}, \mathfrak{B})$ hat, *gibt es* wegen des EF-Theorems einen FO-Satz φ mit $\mathsf{qr}(\varphi) = 2$, durch den sich $\mathfrak{A}$ und $\mathfrak{B}$ *unterscheiden* lassen, z. B.:

  $$\varphi = \exists x\, \forall y\, \neg E(y, x)$$

T3.14 Beweis des Methodologie-Theorems

Theorem 3.20. Sei P eine Eigenschaft. Wenn es für jedes $k \geq 0$ Strukturen $\mathfrak{A}_k$, $\mathfrak{B}_k$ gibt, so dass

1. $\mathfrak{A}_k \in P$ und $\mathfrak{B}_k \notin P$ und
2. *Duplicator* hat eine Gewinnstrategie für $\mathcal{G}_k(\mathfrak{A}_k, \mathfrak{B}_k)$,

dann ist P nicht FO-ausdrückbar.

Beweis. Wir beweisen das Kontrapositiv. Sei also P FO-ausdrückbar mittels eines Satzes φ. Wir müssen zeigen: es *gibt ein* $k \geq 0$, so dass für *alle* Strukturen $\mathfrak{A}, \mathfrak{B}$ gilt: Punkt 1 und 2 aus dem Methodologietheorem sind nicht beide erfüllt.

Wähle dafür $k = \mathsf{qr}(\varphi)$. Seien $\mathfrak{A}, \mathfrak{B}$ beliebige Strukturen. Wenn Punkt 1 für $\mathfrak{A}$ und $\mathfrak{B}$ *nicht* erfüllt ist, dann folgt die Behauptung. Wenn jedoch Punkt 1 erfüllt ist, dann gilt $\mathfrak{A} \in P$ und $\mathfrak{B} \notin P$. Das Ehrenfeucht-Fraïssé-Theorem (Thm. 3.19) liefert dann, dass *Duplicator keine* Gewinnstrategie für $\mathcal{G}_k(\mathfrak{A}, \mathfrak{B})$ hat – also ist Punkt 2 nicht erfüllt. ❑

T3.15 Beweis der Nicht-Ausdrückbarkeit von EVEN und ODD

Theorem 3.21. EVEN und ODD sind nicht FO-ausdrückbar – weder in der Klasse aller Strukturen noch in der Klasse der endlichen Strukturen.

Beweis. Wir zeigen die Behauptung für EVEN mittels des Methodologietheorems (Thm 3.20). Für ODD ist die Argumentation analog.

Sei $k \geq 0$ beliebig. Wähle Strukturen $\mathfrak{A}_k$ und $\mathfrak{B}_k$ wie folgt:

- $\mathfrak{A}_k$ hat $2k$ Elemente und $R^{\mathfrak{A}_k} = \emptyset$ für alle Relationssymbole R.
- $\mathfrak{B}_k$ hat $2k + 1$ Elemente und $R^{\mathfrak{B}_k} = \emptyset$ für alle Relationssymbole R.

Siehe Abbildung 2.

Abbildung 2.: Strukturen $\mathfrak{A}_k$ und $\mathfrak{B}_k$ für $k = 3$

Offenbar ist Punkt 1 des Methodologietheorems (Thm 3.20) erfüllt, denn $\mathfrak{A}_k \in$ EVEN und $\mathfrak{B}_k \notin$ EVEN.

Für Punkt 2 ist zu zeigen, dass *Duplicator* eine Gewinnstrategie für $\mathcal{G}_k(\mathfrak{A}_k, \mathfrak{B}_k)$ hat. Diese ist wie folgt:

- Nach dem ersten Zug von *Spoiler* in einer der beiden Strukturen wählt *Duplicator* ein beliebiges Element aus der jeweils anderen Struktur.

- In jeder weiteren Runde verfährt *Duplicator* so: Wählt *Spoiler* ein in einer früheren Runde gespieltes Element, dann antwortet *Duplicator* mit dem in derselben Runde gespielten Element in der anderen Struktur. Wählt jedoch *Spoiler* ein noch nicht gespieltes Element einer Struktur, dann antwortet *Duplicator* wieder mit einem beliebigen Element der anderen Struktur. Da $|A| \geq k$ und $|B| \geq k$, ist das stets möglich.

Offensichtlich entsteht auf diese Weise ein partieller Isomorphismus. ❑

T3.16 Skizze der Gewinnstrategie für Zusammenhang

Sei $k \geq 0$. Wir wählen ungerichtete Graphen $\mathfrak{A}_k, \mathfrak{B}_k$ wie folgt:

- $\mathfrak{A}_k$ ist ein Kreis K_1 der Länge 2^k (also zusammenhängend).
- $\mathfrak{B}_k$ besteht aus zwei disjunkten Kreisen K_2, K_3 der Länge 2^k (also nicht zusammenhängend).

Siehe Abbildung 3.

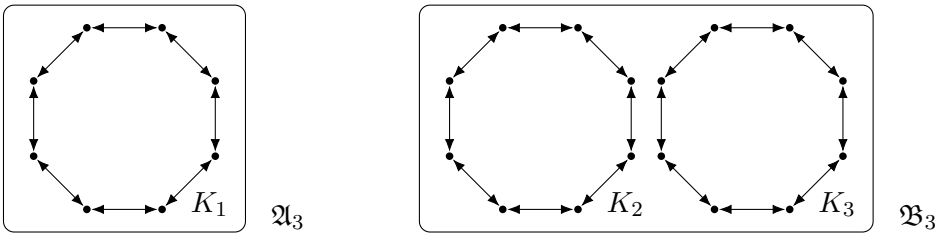


Abbildung 3.: Strukturen $\mathfrak{A}_k$ und $\mathfrak{B}_k$ für $k = 3$. Alle Kanten sind mit E beschriftet.

Behauptung: *Duplicator* hat Gewinnstrategie für $\mathcal{G}_k(\mathfrak{A}_k, \mathfrak{B}_k)$.

Beste Spielweise von *Spoiler* und *Duplicator*:

- In den ersten beiden Runden wählt *Spoiler* ein Element in K_2 und eins in K_3.
- *Duplicator* muss in diesen Runden dann jeweils ein Element in K_1 wählen. Diese sind verbunden, im Gegensatz zu den von Spoiler gewählten Elementen; darum kann *Spoiler* im Prinzip gewinnen. *Duplicator* kann jedoch den Sieg von *Spoiler* so weit hinauszuzögern, dass dieser nicht in den ersten k Runden eintritt. Dazu wählt sie zwei *gegenüberliegende* Elemente in K_1, denn die beiden Pfade zwischen zwei solchen Elementen haben die Länge 2^{k-1}, wohingegen es zwischen zwei nicht gegenüberliegenden Elementen immer einen kürzeren Pfad gibt.
- Die beste Strategie für *Spoiler* besteht nun in „binärer Suche“: Wähle ein Element genau in der Mitte zwischen den von *Duplicator* gewählten, was die Strecke genau halbiert. Auch in den folgenden Runden muss *Spoiler* die Strecke zwischen zwei am wenigsten voneinander entfernten gewählten Elementen halbieren.
- Auf jeden dieser Züge von *Spoiler* antwortet *Duplicator* mit einem Element in K_2 oder K_3, wobei sie dieselben Abstände einhält wie *Spoiler*. Dies kann sie k Runden

lang durchhalten.

Dies ist noch kein Beweis für die obige Behauptung, denn die beschriebene Strategie von *Duplicator* basiert auf der Annahme, dass *Spoiler* optimal spielt. Sie muss aber auch funktionieren, wenn *Spoiler* nicht optimal spielt. Deshalb ist es wesentlich komplizierter, die Gewinnstrategie für *Duplicator* zu beschreiben; siehe den folgenden Beweis von Lemma 3.22.

Man beachte auch, dass man hier nicht einfach einen Spielbaum zeichnen kann, denn dessen Verzweigungsgrad ist durch die Anzahl der Elemente in den Strukturen bestimmt, welche wiederum von k abhängt, und dessen Wert ist beliebig.

(fakultativ) Gewinnstrategie für Zusammenhang

Lemma 3.22. *Duplicator* kann $\mathcal{G}_k(\mathfrak{A}_k, \mathfrak{B}_k)$ so spielen, dass nach i Runden ein Spielstand $\{(a_1, b_1), \ldots, (a_i, b_i)\}$ erreicht ist, so dass für $1 \leq j < \ell \leq i$:

$$d(a_j, a_\ell) \;=\; d(b_j, b_\ell) \quad \text{oder} \quad d(a_j, a_\ell),\; d(b_j, b_\ell) \;>\; 2^{k-i} \qquad (*)$$

Anmerkung. Intuitiv gesprochen sagt die Bedingung $(*)$, dass zwei Elemente in einer Struktur (z. B. a_i, a_ℓ) denselben Abstand haben wie die zugehörigen Elemente in der anderen Struktur (b_i, b_ℓ) *oder* dass beide Abstände einen gewissen Schwellwert überschreiten (2^{k-i}). Dieser Schwellwert wird kleiner, je mehr Runden bereits gespielt wurden: mit wachsendem i fällt der Wert 2^{k-i}. Dies verdeutlicht, dass die „Unterscheidungskraft“ von *Spoiler* sinkt, je weiter das Spiel fortgeschritten ist.

Aus der Bedingung $(*)$ des Lemmas lässt sich bereits eine konkrete Strategie für *Duplicator* ablesen. Wir müssen dann nur noch zeigen, dass diese eine Gewinnstrategie ist. Die Gewinnstrategie lässt sich so formulieren:

Wenn Spoiler a_ℓ wählt, dann gibt es zwei Möglichkeiten:

- Alle schon gewählten Elemente a_j sind „weit genug weg“ (2. Teil „oder“). Dann kann *Duplicator* b_ℓ so wählen, dass alle schon gewählten Elemente b_j ebenfalls „weit weg“ sind;
- Mindestens ein gewähltes Element ist „nah“ bei a_ℓ (1. Teil „oder“). Dann kann *Duplicator* b_ℓ so wählen, dass alle „nahen“ Elemente denselben Abstand zu a_ℓ haben wie ihre Bilder zu b_ℓ.

Was „nah“ bzw. „weit“ ist, verändert sich mit der Spieldauer (Schwellwert 2^{k-1}). Die konkrete Wahl von b_ℓ ist dann ungefähr wie bereits beschrieben.

Beweis. Für ein Element u aus $\mathfrak{A}_k$ oder $\mathfrak{B}_k$ und eine Zahl $\ell \geq 0$ heiße die Menge $N_\ell(u) = \{v \in V \mid d(u, v) \leq \ell\}$ die *ℓ-Nachbarschaft* von u.

Wir beweisen $(*)$ per Induktion über i.

Induktionsanfang. Für $i = 0$ ist $(*)$ trivialerweise erfüllt.

Induktionsschritt.

Wir nehmen an, dass Spoiler im $(i+1)$-ten Zug ein Element $a = a_{i+1} \in A$ wählt. Die Wahl eines $b = b_{i+1} \in B$ kann symmetrisch behandelt werden.

Unterscheide zwei Fälle:

1. Es gibt $a_h \in \{a_1, \ldots, a_i\}$ mit $d(a_h, a) \leq 2^{k-(i+1)}$. ($(*)$-Schwelle für $i+1$)

   Betrachte die Nachbarschaften $N_{2^{k-i}}(a_h)$ und $N_{2^{k-i}}(b_h)$. IV liefert für alle $a_j, a_\ell \in \{a_1, \ldots, a_i\}$:

   (I) $a_j \in N_{2^{k-i}}(a_h)$ gdw. $b_j \in N_{2^{k-i}}(b_h)$

   (II) Wenn $a_j, a_\ell \in N_{2^{k-i}}(a_h)$, dann $d(a_j, a_\ell) = d(b_j, b_\ell)$.

   Also gibt es Bijektion von $N_{2^{k-i}}(a_h)$ auf $N_{2^{k-i}}(b_h)$, die jedes $a_j \in N_{2^{k-i}}(a_h)$, $j \in \{1, \ldots, i\}$, auf das entsprechende b_j abbildet. Es gilt $a \in N_{2^{k-i}}(a_h)$. Sei b das Bild von a unter der Bijektion. Dann gilt für alle $a_j \in \{a_1, \ldots, a_i\}$:

   (III) Wenn $a_j \in N_{2^{k-i}}(a_h)$, dann $d(a_j, a) = d(b_j, b)$.

   Duplikator wählt dieses b als b_{i+1}. Zu zeigen: für alle $a_j \in \{a_1, \ldots, a_i\}$ gilt:

   $$d(a_j, a) = d(b_j, b) \quad \text{oder} \quad d(a_j, a), d(b_j, b) > 2^{k-(i+1)}$$

   Unterscheide 2 Fälle:

   a) $a_j \in N_{2^{k-i}}(a_h)$. Folgt direkt aus (III).

   b) $a_j \notin N_{2^{k-i}}(a_h)$.

      Offensichtlich gilt $d(a_j, a_h) \leq d(a_j, a) + d(a, a_h)$. Also auch

      $$\begin{aligned} d(a_j, a) &\geq d(a_j, a_h) - d(a, a_h) \\ &> 2^{k-i} - 2^{k-(i+1)} \qquad (\text{denn } d(a_j, a_h) > 2^{k-i} \\ &\qquad\qquad\qquad\qquad\quad \text{und } d(a, a_h) \leq 2^{k-(i+1)}) \\ &= 2^{k-(i+1)} \end{aligned}$$

      Nach (I) gilt $b_j \notin N_{2^{k-i}}(b_h)$. Mit (III) auch $d(b, b_h) \leq 2^{k-(i+1)}$. Wir können also ganz analog zeigen, dass $d(b_j, b) > 2^{k-(i+1)}$.

2. Es gibt kein $a_h \in \{a_1, \ldots, a_i\}$ mit $d(a_h, a) \leq 2^{k-(i+1)}$.

   Wir zeigen: es gibt ein $b \in B$ so dass $d(b_j, b) > 2^{k-(i+1)}$ für alle $j \in \{1, \ldots, i\}$. Wählt Duplikator dieses b als b_{i+1}, so ist $(*)$ offensichtlich erfüllt.

   Seien $b_{r_1}, \ldots, b_{r_i}$ die Elemente von $\{b_1, \ldots, b_i\}$, die auf dem ersten Kreis in B liegen, geordnet in der Reihenfolge auf dem Kreis. Angenommen, es gibt kein b wie beschrieben. Dann gilt

   $$d(b_{r_\ell}, b_{r_{\ell+1}}) \leq 2^{k-i} \text{ für } 1 \leq \ell \leq i, \text{ wobei } b_{r_{i+1}} = b_{r_1}.$$

   Also hat der Kreis höchstens

   $$i \cdot 2^{k-i} = 2^{k-i+\log(i)} < 2^k$$

   Knoten. Widerspruch. ❏

T3.17 Korrektheit der Gewinnstrategie

Korollar 3.23. *Duplicator* hat eine Gewinnstrategie für $\mathcal{G}_k(\mathfrak{A}_k, \mathfrak{B}_k)$.

Beweis. Sei also $k \geq 0$ und seien $\mathcal{G}_k(\mathfrak{A}_k, \mathfrak{B}_k)$ die in T3.16 beschriebenen Strukturen. Wenn Duplicator spielt wie im vorangehenden Beweis von Lemma 3.22 beschrieben, dann ergibt das einen abschließenden Spielstand $\delta = \{(a_1, b_1), \ldots, (a_k, b_k)\}$, für den wegen Lemma 3.22 für $1 \leq j < \ell \leq k$ gilt:

$$d(a_j, a_\ell) \;=\; d(b_j, b_\ell) \quad \text{oder} \quad d(a_j, a_\ell),\; d(b_j, b_\ell) \;>\; 1 \qquad (*)$$

Wir müssen noch zeigen, dass δ ein partieller Isomorphismus ist.

- δ ist eine Funktion:

  Wenn $a_i = a_j$ für $i \neq j$, dann ist $d(a_i, a_j) = 0$, also mit $(*)$ auch $d(b_i, b_j) = 0$, also $b_i = b_j$.

- δ ist eine Bijektion:

  Nach Definition ist δ surjektiv. Für die Injektivität betrachte a_i, a_j mit $a_i \neq a_j$. Dann ist $d(a_i, a_j) \neq 0$, also muss wegen $(*)$ auch $d(b_i, b_j) > 0$ sein, also $b_i \neq b_j$.

- δ ist ein Isomorphismus:

$$\begin{array}{lll} (a_i, a_j) \in E^{\mathfrak{A}} & \text{gdw.} & d(a_i, a_j) = 1 \\ & \text{gdw.} & d(b_i, b_j) = 1 \quad \text{wegen } (*) \\ & \text{gdw.} & (b_i, b_j) \in E^{\mathfrak{A}} \end{array}$$

❑

Teil IV.

Prädikatenlogik 2. Stufe

T4.1 Korrektheit der Formel für Erreichbarkeit

Sei $\varphi(x,y) = \forall X \Big(X(x) \wedge \underbrace{\forall z \forall z' \left(X(z) \wedge E(z,z') \rightarrow X(z') \right)}_{\psi(X)} \rightarrow X(y) \Big)$ und $\tau = \{E\}$, also die Signatur gerichteter Graphen. Intuitiv besagt

- $\psi(X)$, dass die Knotenmenge X unter E-Nachfolgern abgeschlossen ist;
- $\varphi(x,y)$, dass jede Knotenmenge, die x enthält und unter E-Nachfolgern abgeschlossen ist, auch y enthält.

Behauptung: Für alle τ-Strukturen $\mathfrak{A}$ und Elemente $a, b \in A$ gilt

$$\mathfrak{A} \models \varphi[a,b] \quad \text{gdw.} \quad \text{es einen Pfad in } \mathfrak{A} \text{ von } a \text{ zu } b \text{ gibt.}$$

Beweis.

„⇒“ Gelte $\mathfrak{A} \models \varphi[a,b]$. Definiere folgende Menge $R \subseteq A$:

$$R = \{\hat{a} \mid \text{es gibt einen Pfad in } \mathfrak{A} \text{ von } a \text{ zu } \hat{a}\}$$

Da die Menge R abgeschlossen unter E-Nachfolgern ist, gilt $\mathfrak{A} \models \psi[R]$. Außerdem gilt $a \in R$. Wegen $\mathfrak{A} \models \varphi[a,b]$ (Voraussetzung) gilt dann $b \in R$. Nach Definition von R gibt es demnach einen Pfad in $\mathfrak{A}$ von a zu b.

„⇐“ Gebe es einen Pfad in $\mathfrak{A}$ von a zu b. Sei $R \subseteq A$ beliebig mit (i) $a \in R$ und (ii) $\mathfrak{A} \models \psi[R]$. Wegen (ii) ist R abgeschlossen unter Nachfolgern. Da es einen Pfad von a zu b gibt, muss auch $b \in R$ sein. Also gilt $\mathfrak{A} \models \varphi[a,b]$. ❏

T4.2 Platzkomplexität des Auswertungsalgorithmus

Lemma. 4.5 Der Algorithmus `ausw` benötigt

1. polynomiell viel Platz, wenn die Eingabe eine MSO-Formel ist;
2. exponentiell viel Platz im Allgemeinen.

Beweis. Die Eingabe sei eine Struktur $\mathfrak{A}$ der Größe n und eine Formel φ der Größe k.

1. *(MSO)*
   In jedem Schritt ist eine Teilmenge von A zu speichern, z. B. über einen Bitstring der Länge $|A| \leq n$. Zudem ist die Rekursionstiefe nach wie vor durch die Verschachtelungstiefe der Formel, also letztlich durch k beschränkt. Dies liefert einen gesamten Platzbedarf von $\mathcal{O}(n \cdot k)$.
2. *(SO)*
   In jedem Schritt ist eine Menge von ℓ-Tupeln von Elementen aus A zu speichern. Es gibt maximal n^ℓ solche Tupel; zudem ist $\ell < k$ (die Stelligkeit von in der Formel vorkommenden Relationsvariablen kann natürlich nicht größer sein als die Länge der Formel). Bei Rekursionstiefe k ist der Platzbedarf also $\mathcal{O}(n^k \cdot k)$. ❏

T4.3 Zeitkomplexität des Auswertungsalgorithmus

Lemma 4.6. Bei Eingabe einer Struktur $\mathfrak{A}$ der Größe n und einer Formel φ der Größe k benötigt der Algorithmus `ausw`

1. Zeit $\mathcal{O}(n^k)$, wenn φ eine FO-Formel ist;
2. Zeit $2^{\mathcal{O}(nk)}$, wenn φ eine MSO-Formel ist;
3. Zeit $2^{\mathcal{O}(n^{2k})}$ im Allgemeinen.

Beweis. Der Zeitbedarf entspricht im Wesentlichen der Größe des Rekursionsbaums, also der Anzahl dessen Knoten. Wir müssen also jeweils die maximale Tiefe und den maximalen Verzweigungsgrad bestimmen. In allen drei Fällen ist die Tiefe nach wie vor durch k beschränkt (wie oben).

1. *(FO)*

   Hier ist der Verzweigungsgrad durch n beschränkt, denn im Fall $\exists x\,\psi$ bzw. $\forall x\,\psi$ muss für jedes Element $a \in A$ ein Unteraufruf ausgeführt werden, und $|A| \leq n$. Damit hat der Baum maximal n^k Knoten.

2. *(MSO)*

   Hier ist der Verzweigungsgrad durch 2^n beschränkt, denn im Fall $\exists X\,\psi$ bzw. $\forall X\,\psi$ muss für jede *Teilmenge* $B \subseteq A$ ein Unteraufruf ausgeführt werden, und es gibt 2^n solche Teilmengen, da $|A| \leq n$. Damit hat der Baum maximal $(2^n)^k = 2^{nk}$ Knoten.

3. *(SO)*

   Hier ist der Verzweigungsgrad durch 2^{n^ℓ} beschränkt, wenn ℓ die maximale Stelligkeit einer in φ vorkommenden Relationsvariable ist, denn im Fall $\exists X\,\psi$ bzw. $\forall X\,\psi$ muss nun für jede Teilmenge $B \subseteq A^\ell$ ein Unteraufruf ausgeführt werden. Da ℓ auch durch die Formelgröße k beschränkt ist, hat der Baum maximal $(2^{n^k})^k = 2^{n^k k} \leq 2^{n^{2k}}$ Knoten. ❏

T4.4 SO-Tautologien sind nicht rekursiv aufzählbar

Theorem 4.8. Die Menge der Tautologien in SO ist nicht rekursiv aufzählbar.

Beweis. Wir nehmen an, die Tautologien in SO seien rekursiv aufzählbar. Wir wollen zeigen, dass dann auch die *erfüllbaren* Formeln in $\mathsf{FO}(\tau)$ rekursiv aufzählbar wären, wobei τ die Signatur ist, die aus einem einzigen zweistelligen Relationssymbol R besteht. Da dies ein Widerspruch zu Korollar 3.8 ist, muss unsere Annahme falsch sein, und die Behauptung gilt.

Sei $\varphi \in \mathsf{FO}(\{R\})$. Es gilt:

- φ hat ein Modell der Größe $n \in \mathbb{N}$ gdw.

$$\varphi_n \to \exists R\,\varphi \quad \text{gültig,} \qquad \text{wobei}$$
$$\varphi_n = \exists x_1 \cdots \exists x_n \Big(\bigwedge_{1 \leq i < j \leq n} x_i \neq x_j \;\wedge\; \forall y \bigvee_{1 \leq i \leq n} y = x_i \Big).$$

- φ hat ein unendliches Modell gdw.

$$\varphi_\infty \to \exists R\, \varphi \quad \text{gültig,}$$

  wobei φ_∞ die SO-Formel ist, die ausdrückt, dass ein Modell unendlich groß ist (siehe Beispiel auf Folie 9). Nach den Sätzen von Löwenheim-Skolem für FO braucht man hier nicht zwischen unendlichen Modellen verschiedener Größe zu unterscheiden.

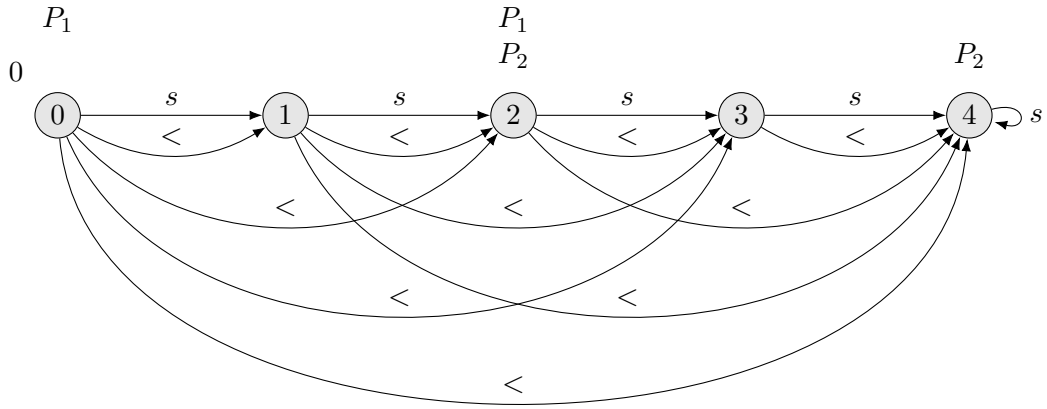
Unter der Annahme, dass die SO-Tautologien rekursiv aufzählbar seien, kann man nun die erfüllbaren FO-Formeln wie folgt aufzählen:

- Zähle alle gültigen SO-Formeln auf.
- Für Formel der Form $\varphi_n \to \exists R\, \varphi$ oder $\varphi_\infty \to \exists R\, \varphi$ mit $\varphi \in \mathsf{FO}(\{R\})$ gib φ aus.

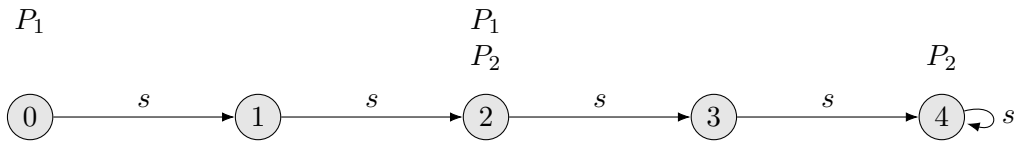
Damit ist der gewünschte Widerspruch hergestellt und die Annahme widerlegt. ❑

T4.5 Beispiel einer S1S-Struktur und -Formel

Betrachte folgende S1S-Struktur $\mathfrak{A}$:



Von nun an werden wir nur noch die Nachfolgerfunktion s einzeichnen, nicht mehr die Konstante 0 oder die Relation $<$. Die obige Struktur $\mathfrak{A}$ stellen wir also abgekürzt so dar:



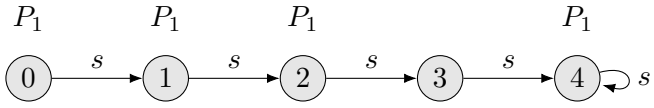
Es gilt $\mathfrak{A} \models \forall x \left(P_1(x) \vee P_2(x)\right)$ (d. h.: jedes Element ist entweder mit P_1 markiert oder sein Nachfolger mit P_2).

T4.6 Beispiel für Strukturen vs. Wörter, Alphabetgröße $n = 1$

Sei $n = 1$, also $\Sigma_1 = \{0, 1\}$. Dann gibt es nur ein einziges P_i, nämlich P_1, und

- wenn ein Element i mit P_1 markiert ist (also $i \in P_1^{\mathfrak{A}}$), dann entspricht das dem Buchstaben 1 an Position i im Wort;
- wenn ein Element i *nicht* mit P_1 markiert ist (also $i \notin P_1^{\mathfrak{A}}$), dann entspricht das dem Buchstaben 0 an Position i im Wort.

Die Struktur



repräsentiert also das Wort 11101.

T4.7 Anmerkung zu „krummen“ Alphabetgrößen

Die Alphabete $\Sigma_n = 0, 1^n$ haben natürlich 2^n Elemente. Man kann aber trotzdem auch Alphabete betrachten mit einer Anzahl von Buchstaben, die keine Zweierpotenz ist: Wenn man z. B. an $\Sigma = \{a, b, c\}$ interessiert ist, dann trifft man zunächst die Zuordnung

$$\binom{0}{0} \mapsto a, \qquad \binom{0}{1} \mapsto b, \qquad \binom{1}{0} \mapsto c.$$

Dann muss man noch ausschließen, dass der nicht benötigte Buchstabe $\binom{1}{1}$ nirgends vorkommt. Dazu fügt man jeder S1S-Formel das folgende Konjunkt hinzu:

$$\neg\exists x \Big(P_0(x) \wedge P_1(x) \Big)$$

Jede so erweiterte Formel hat nur Modelle, in denen $\binom{1}{1}$ nicht als Markierung vorkommt.

T4.8 Beispiele MSO-definierter Sprachen

Im Folgenden wählen wir der Einfachheit halber $n = 1$, also $\Sigma_1 = \{0, 1\}$. Wir verwenden $\mathsf{last}(x)$ als Abkürzung für $s(x) = x$, also eine Formel mit einer freien Variablen, die besagt, dass das entsprechende Element (Position im Wort) das Wortende ist.

Beispiel 1

$$\begin{aligned} \varphi_1 \;=\; P_1(0) \;\wedge\; \forall x \Big(& \Big((P_1(x) \wedge \neg\mathsf{last}(x)) \rightarrow \neg P_1\big(s(x)\big) \Big) \;\wedge \\ & \Big((\neg P_1(x) \wedge \neg\mathsf{last}(x)) \rightarrow P_1\big(s(x)\big) \Big) \;\wedge \\ & \mathsf{last}(x) \rightarrow \neg P_1(x) \Big) \end{aligned}$$

$$L(\varphi_1) \;=\; (10)^*$$

Beachte: die Objektvariablen stehen für Elemente des Universums, also quantifiziert $\forall x$ über die *Positionen des Wortes*.

Beispiel 2

$$\begin{aligned}\varphi_2 \;=\; \forall X \Big(& X(0) \;\wedge \\ & \forall y \Big(\big((X(y) \wedge \neg \mathsf{last}(y)) \rightarrow \neg X(s(y)) \big) \;\wedge \\ & \quad \big((\neg X(y) \wedge \neg \mathsf{last}(y)) \rightarrow X(s(y)) \big) \Big) \\ & \rightarrow \forall y \Big(\mathsf{last}(y) \rightarrow \neg X(y) \Big) \Big)\end{aligned}$$

$$L(\varphi_2) \;=\; \{w \in \Sigma^* \mid |w| \text{ ist geradzahlig}\}$$

Beispiel 3

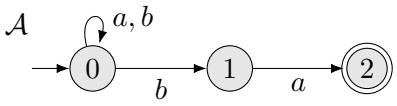
$$\begin{aligned}\varphi_3 \;=\;& \neg\exists x \Big(P_1(x) \wedge \neg P_1\big(s(x)\big) \Big) \\ & \vee \neg\exists x \Big(\neg P_1(x) \wedge P_1\big(s(x)\big) \Big)\end{aligned}$$

$$L(\varphi_3) \;=\; 0^*1^* \cup 1^*0^*$$

T4.9 Beweis Satz von Büchi-Elgot-Trakhtenbrot, „1 ⇒ 2“

Behauptung: Wenn L regulär ist, dann $L = L(\varphi)$ für einen S1S-Satz φ.

Beweis. Wir wollen zeigen, dass man jeden nichtdeterministischen endlichen Automaten (NEA) $\mathcal{A}$ in einen S1S-Satz φ umwandeln kann mit $L(\varphi) = L(\mathcal{A})$. Sei also $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ ein NEA. Eine akzeptierende Berechnung von $\mathcal{A}$ auf einer Eingabe w kann man auffassen als Beschriftung von w mit Zuständen aus Q, die die Übergangsrelation einhält.[5] Betrachten wir also z. B. den NEA



und das Eingabewort $w = aabaaba$, so gibt es folgende Berechnung von $\mathcal{A}$ auf w:

$$\begin{array}{rl} w \;=\; & a\,a\,b\,a\,a\,b\,a \\ & 0\,0\,0\,0\,0\,1\,2 \end{array}$$

[5] Dies ist eine Variante der üblichen Definition von Akzeptanz über Pfade, siehe auch „Theoretische Informatik 1“ [Sch19].)

Diese Beschriftung stellt tatsächlich eine akzeptierende Berechnung von $\mathcal{A}$ auf w dar, denn

- 0 (1. Markierung) ist der Anfangszustand von $\mathcal{A}$;
- $(0, a, 0) \in \Delta$ (1. Markierung, 1. Buchstabe von w, 2. Markierung);
- $(0, a, 0) \in \Delta$ (2. Markierung, 2. Buchstabe von w, 3. Markierung);

  $\vdots$
- $(1, a, 2) \in \Delta$ (vorletzte Markierung, letzter Buchstabe von w, letzte Markierung);
- $2 \in F$ (letzte Markierung ist ein akzeptierender Zustand).

Wir definieren nun einen S1S-Formel $\varphi_{\mathcal{A}}$, die ausdrückt, dass eine beliebige Beschriftung eine akzeptierende Berechnung im Sinne der eben aufgelisteten Eigenschaften darstellt. Dazu nehmen wir o. B. d. A. an, die Zustandsmenge von $\mathcal{A}$ sei $Q = \{q_0, \ldots, q_m\}$ und das Alphabet sei Σ_n. Für jedes Zeichen $a = (b_1, \ldots b_n) \in \Sigma_n$ verwenden wir eine Abkürzung, die besagt, dass sich an einer Position x im Wort das Zeichen a befindet:

$$S_a(x) = \bigwedge_{\substack{1 \le i \le n \\ b_i = 0}} \neg P_i(x) \ \wedge \bigwedge_{\substack{1 \le i \le n \\ b_i = 1}} P_i(x)$$

Außerdem verwenden wie die Abkürzung $\mathsf{last}(x)$ aus dem vorigen Beispiel.
Wir definieren $\varphi_{\mathcal{A}}$ wie folgt.

$$\begin{aligned}
\varphi_{\mathcal{A}} \ = \ & \exists Q_0 \cdots \exists Q_m \Bigg(\\
& Q_0(0) \ \wedge \\
& \forall x \Big(\neg \mathsf{last}(x) \rightarrow \bigvee_{(q_i, a, q_j) \in \Delta} \Big(Q_i(x) \wedge S_a(x) \wedge Q_j(s(x)) \Big) \Big) \ \wedge \\
& \forall x \Big(\mathsf{last}(x) \rightarrow \bigvee_{\substack{(q_i, a, q_j) \in \Delta \\ q_j \in F}} \Big(Q_i(x) \wedge S_a(x) \Big) \Big) \ \wedge \\
& \forall x \bigwedge_{0 \le i < j \le m} \neg \Big(Q_i(x) \wedge Q_j(x) \Big)
\end{aligned}$$

Die Intuition hinter $\varphi_{\mathcal{A}}$ ist folgende. Die einstelligen Relationsvariablen Q_i repräsentieren alle diejenigen Positionen im Wort, die mit q_i beschriftet sind. Dabei läuft i von 0 bis einschließlich m, denn das Wort hat m Buchstaben an Positionen $0, \ldots, m-1$, und die Beschriftung hat eine Position mehr (siehe Beispiel). Die einzelnen Konjunkte in $\varphi_{\mathcal{A}}$ besagen nun, dass

1. die Berechung von $\mathcal{A}$ in einem Startzustand beginnt,
2. die Übergangsrelation eingehalten wird,
3. die Berechnung in einem akzeptierenden Zustand endet
   (dieser kann *keiner* Position im Wort zugeordnet werden),

4. die Beschriftung nicht „entartet“ ist, also jede Position mit höchstens einem Zustand beschriftet ist („mit *mindestens* einem“ folgt aus den vorhergehenden Konjunkten – überzeuge Dich selbst!).

Man zeigt nun leicht:

$$w \in L(\mathcal{A}) \quad \text{gdw.} \quad w \in L(\varphi_{\mathcal{A}}) \text{ für alle } w \in \Sigma^*$$

Dazu muss man aus einer akzeptierenden Berechnung von $\mathcal{A}$ (Beschriftung von w) ein S1S-Modell für $\varphi_{\mathcal{A}}$ konstruieren und umgekehrt. Das Beschreiben dieser Konstruktionen ist eine gute Übung für Euch. :) ❑

T4.10 Beispiel S1S-Normalform

Wir betrachten wieder $\Sigma_1 = \{0, 1\}$. Die folgende S1S-Formel ist in Normalform:

$$\begin{aligned} \varphi \;=\; & \neg\exists X\, \exists Y \Big(X \subseteq P_1 \;\wedge\; \neg(Y \subseteq P_1) \;\wedge\; \mathsf{succ}(X) = Y \Big) \\ & \vee \neg\exists X\, \exists Y \Big(\neg(X \subseteq P_1) \;\wedge\; Y \subseteq P_1 \;\wedge\; \mathsf{succ}(X) = Y \Big) \end{aligned}$$

Man beachte dabei: die Konjunkte $\mathsf{succ}(X) = Y$ beinhalten, dass X und Y Einermengen sind; deshalb bedeutet z. B. $\neg(Y \subseteq P_1)$, dass an der Position, die durch Y repräsentiert wird, *nicht* P_1 steht.

Diese Formel entspricht der aus Beispiel 3 in T4.8: die beiden Disjunkte besagen, dass es kein Infix 10 bzw. 01 gibt. Damit ist wieder $L(\varphi) = 0^*1^* \cup 1^*0^*$.

T4.11 Beweis des Lemmas zur S1S-Normalform

Lemma 4.12. Jeder S1S-Satz kann effektiv in einen äquivalenten Satz in Normalform gewandelt werden.

Beweis. (Skizze) Im Folgenden werden die notwendigen Schritte zur Herstellung der Normalform an konkreten Beispielen verdeutlicht. Es ist vergleichsweise leicht, diese Schritte allgemeingültig zu formulieren und zu zeigen, dass nach jedem Schritt die umgewandelte Teilformel äquivalent zur ursprünglichen ist.

Schritt 1: Eliminieren der Symbole 0 und $<$

Zum Beispiel:

$$\begin{aligned} P_i(0) \;&\equiv\; \exists x \Big(\forall y\, \neg\big(s(y) = x\big) \;\wedge\; P_i(x) \Big) \\ x > 0 \;&\equiv\; \exists y \Big(s(y) = x \Big) \\ t < t' \;&\equiv\; \exists X \Big(X(t') \;\wedge\; \forall z \Big(X(z) \to X\big(s(z)\big) \Big) \;\wedge\; \neg X(t) \Big) \end{aligned}$$

In der ersten und zweiten Zeile wird die Anfangsposition des Wortes ähnlich umschrieben wie in den vorigen Beispielen die Endposition mittels $\mathsf{last}(x)$. In der dritten Zeile stehen t und t' für beliebige Terme (die aus Objektvariablen, dem Konstantensymbol 0 und der Nachfolgerfunktion s gebildet werden können). Die rechte Formel besagt, dass es eine Menge von Positionen gibt (X), die t' enthält, abgeschlossen unter Nachfolgern ist (also „induktiv“ alle Positionen hinter t' enthält) und nicht t enthält. Das ist natürlich genau dann der Fall, wenn Position t vor t' liegt.

Schritt 2: Schachtelung von s eliminieren

Zum Beispiel:

$$x = s(s(s(y))) \;\equiv\; \exists z_1\, \exists z_2 \Big(z_1 = s(y) \;\wedge\; z_2 = s(z_1) \;\wedge\; x = s(z_2) \Big)$$

Mit anderen geschachtelten Vorkommen von s, z. B. in $P_i(s(s(s(y))))$, wird ganz analog umgegangen.

Schritt 3: Umschreiben atomarer Formeln, so dass nur noch Atome der folgenden Formen vorkommen

$$x = y, \quad s(x) = y, \quad P_i(x), \quad X(x)$$

Zum Beispiel:

$$P_i(s(x)) \;\equiv\; \exists y \Big(y = s(x) \;\wedge\; P_i(y) \Big)$$

Schritt 4: Eliminieren von Objektvariablen $(x, \dots)$ in Quantoren und Atomen.

Dazu definieren wir eine Formel $\mathsf{Einer}(X)$ mit einer freien einstelligen Relationsvariablen X, die ausdrückt, dass die entsprechende Menge eine Einermenge ist – dass erreicht man, indem man fordert, dass diese Menge genau eine echte Teilmenge hat:

$$\mathsf{Einer}(X) \;:=\; \exists Y \Big(Y \subseteq X \;\wedge\; \neg(Y = X) \;\wedge\; \forall Z \, (Z \subseteq X \;\to\; Z = X \vee Z = Y) \Big), \quad \text{wobei}$$
$$X = Y \;:=\; X \subseteq Y \wedge Y \subseteq X$$

Beispiele für Schritt 4 sind nun:

$$\exists x\, P_i(x) \;\equiv\; \exists X \Big(\mathsf{Einer}(X) \;\wedge\; X \subseteq P_i \Big)$$
$$\forall x \exists y \Big(s(x) = y \;\wedge\; Z(y) \Big) \;\equiv\; \forall X \Big(\mathsf{Einer}(X) \;\to\; \exists Y \Big(\mathsf{Einer}(Y) \;\wedge\; \mathsf{succ}(X) = Y \;\wedge\; Y \subseteq Z \Big) \Big)$$

(❑)

T4.12 Beweis Satz von Büchi-Elgot-Trakhtenbrot, „2 ⇒ 1“

Behauptung: Für jeden S1S-Satz φ ist $L(\varphi)$ regulär.

Beweis. Sei φ ein S1S-Satz. Wegen des vorangehenden Lemmas können wir o. B. d. A. annehmen, dass φ in Normalform vorliegt. Wir wollen nun induktiv über den Aufbau von φ einen NEA $\mathcal{A}_\varphi$ konstruieren, der $L(\varphi)$ erkennt, also mit $L(\mathcal{A}_\varphi) = L(\varphi)$. Dazu müssen wir auch Teilformeln von φ betrachten, die in der Regel freie Relationsvariablen haben. Diese werden genauso behandelt wie die P_i: beispielsweise hat die Formel $\exists Y\,(X \subseteq Y \wedge P_1 \subseteq Y)$ eine freie Relationsvariable (X) und ein Relationssymbol (P_1) und definiert damit eine Sprache über $\Sigma_2 = \{0,1\}^2$, wenn man sich auf eine Reihenfolge der Relationssymbole und Variablen festlegt (z. B. „das erste Bit entspricht P_1 und das zweite X“).

Wir konstruieren nun $\mathcal{A}_\varphi$ induktiv wie folgt.

Induktionsanfang.

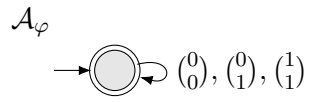
Laut Definition der Normalform gibt es zwei mögliche Formen von atomaren Formeln:

- $\varphi = X \subseteq Y$

  Dann ist das Alphabet $\Sigma_2 = \{0,1\}^2$ und es gilt

  $$L(\varphi) = \left\{ w = \begin{pmatrix} b_1 \\ b'_1 \end{pmatrix} \cdots \begin{pmatrix} b_n \\ b'_n \end{pmatrix} \;\middle|\; \text{wenn } b_i = 1, \text{ dann } b'_i = 1 \right\},$$

  also ist $\mathcal{A}_\varphi$ wie folgt:

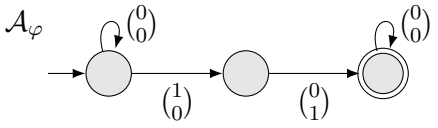
  

- $\varphi = \mathsf{succ}(X) = Y$

  Dann ist das Alphabet wieder $\Sigma_2 = \{0,1\}^2$ und es gilt

  $$L(\varphi) = \left\{ w = \begin{pmatrix} b_1 \\ b'_1 \end{pmatrix} \cdots \begin{pmatrix} b_n \\ b'_n \end{pmatrix} \;\middle|\; \begin{array}{l} \text{es gibt } k \text{ mit } \; b_k = 1,\ b'_{k+1} = 1, \\ \qquad\qquad\quad b_j = 0 \text{ für alle } j \neq k, \\ \qquad\qquad\quad b'_j = 0 \text{ für alle } j \neq k+1 \end{array} \right\},$$

  also ist $\mathcal{A}_\varphi$ wie folgt:

Induktionsschritt.

Wir nehmen die reduzierte Form an, d. h. wir brauchen nur die Operatoren $\neg, \wedge, \exists X$ zu betrachten und können auf $\vee, \forall X$ verzichten (siehe Teil 2, Grundlagen FOL).

- $\varphi = \neg\psi$

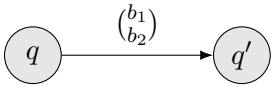
  Nach Induktionsvoraussetzung gibt es einen NEA $\mathcal{A}_\psi$ mit $L(\mathcal{A}_\psi) = L(\psi)$. Da $L(\varphi) = \overline{L(\psi)}$ (Semantik der Negation), suchen wir einen NEA $\mathcal{A}_\varphi$ mit $L(\mathcal{A}_\varphi) = \overline{L(\mathcal{A}_\psi)}$. Diesen können wir konstruieren, indem wir $\mathcal{A}_\psi$ mittels Potenzmengenkonstruktion in einen DEA umwandeln und dann dessen akzeptierende und nicht-akzeptierende Zustände vertauschen (Theoretische Informatik 1, [Sch19]).

- $\varphi = \psi_1 \wedge \psi_2$

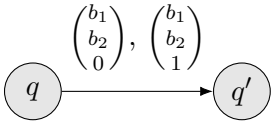
  Nach Induktionsvoraussetzung gibt es einen NEAs $\mathcal{A}_{\psi_i}$ mit $L(\mathcal{A}_{\psi_i}) = L(\psi_i)$ für $i = 1, 2$. Die Semantik der Konjunktion legt nahe, dass $(*)$ $L(\varphi) = L(\psi_1) \cap L(\psi_2)$ ist und wir deshalb einen NEA $\mathcal{A}_\varphi$ suchen sollten mit $L(\mathcal{A}_\varphi) = L(\mathcal{A}_{\psi_1}) \cap L(\mathcal{A}_{\psi_2})$. Dazu könnten wir den Produktautomaten konstruieren (Theoretische Informatik 1, [Sch19]).

  Allerdings ist $(*)$ nicht ganz korrekt, denn die freien Variablen in ψ_1 und ψ_2 müssen nicht übereinstimmen. Hat also z. B. ψ_1 die freien Variablen X_1, X_2 und ψ_2 die freien Variablen X_2, X_3, so hat φ die freien Variablen X_1, X_2, X_3. Damit ist $L(\varphi)$ über einem anderen Alphabet ($\{0,1\}^3$) definiert als die $L(\psi_i)$ (Alphabet $\{0,1\}^2$).

  Um die Produktkonstruktion anwenden zu können, müssen wir also zunächst die NEAs $\mathcal{A}_{\psi_i}$ so erweitern, dass ihre Alphabete gleich sind und alle freien Variablen in φ berücksichtigen. Im obigen Beispiel mit 2 bzw. 3 freien Variablen muss man also jeden Übergang

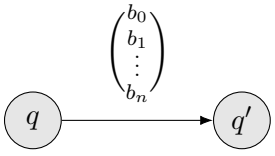
  in $\mathcal{A}_{\psi_1}$ ersetzen durch die Übergänge:

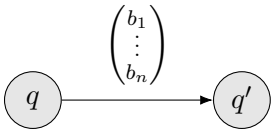
  Der gesuchte NEA $\mathcal{A}_\varphi$ ergibt sich dann mittels Produktkonstruktion aus den modifizierten NEAs $\mathcal{A}_{\psi_i}$, und man kann leicht zeigen, dass $L(\mathcal{A}_\varphi) = L(\varphi)$.

- $\varphi = \exists Y\, \psi(Y, X_1, \ldots, X_n)$ (hier sind die freien Variablen relevant)

  Nach Induktionsvoraussetzung gibt es einen NEA $\mathcal{A}_\psi$ über dem Alphabet Σ_{n+1} mit $L(\mathcal{A}_\psi) = L(\psi)$. Daraus erhält man den gewünschten Automaten $\mathcal{A}_\varphi$ über dem Alphabet Σ_n durch folgende Modifikation: Ersetze jeden Übergang

  in $\mathcal{A}_\psi$ durch:



Das Weglassen der Komponente b_0 ist korrekt, denn die Interpretation von Y ist für φ irrelevant. ❏

Man beachte, dass die Konstruktion im letzten Fall ($\exists$) unter Umständen aus einem DEA einen NEA macht, denn es kann vor dem Löschen von b_0 zwei Übergänge zu zwei verschiedenen Zuständen gegeben haben mit zwei Zeichen, die sich nur durch den Wert von b_0 unterscheiden, z. B.:

Dadurch muss man im schlimmsten Fall für jede Negation erneut die Potenzmengenkonstruktion anwenden, welche jedes Mal den Automaten exponentiell vergrößern kann. Im schlimmsten Fall wird der Automat also *nicht-elementar* größer: für eine Formel φ der Länge n kann die Anzahl der Zustände im Automaten $\mathcal{A}_\varphi$ so groß werden:

$$\left.2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}}\right\} \text{Höhe } n$$

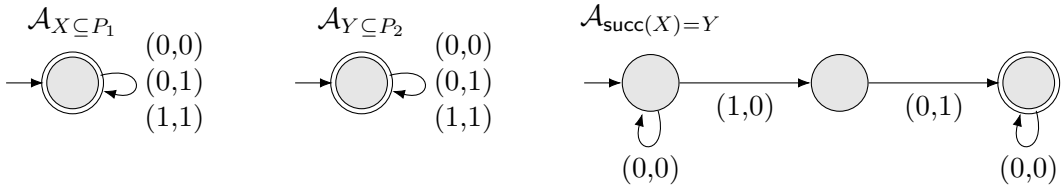
T4.13 Beispiel für die Konstruktion im vorangehenden Beweis

Sei $\varphi = \exists X\, \exists Y \Big(\underbrace{X \subseteq P_1 \wedge Y \subseteq P_2 \wedge \mathsf{succ}(X) = Y}_{\psi} \Big)$. Die Sprache $L(\varphi) \subseteq \Sigma_2$ ist:

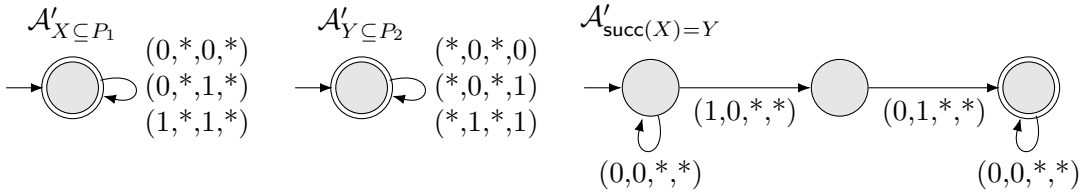
$$\Sigma_2^* \cdot \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \cup \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) \cdot \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \cup \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) \cdot \Sigma_2^*$$

Zur Erinnerung: jedes Relationssymbol P_i und jede *freie* Relationsvariable X, Y wird durch ein „Bit“ repräsentiert. Die Reihenfolge legen wir dabei fest als X, Y, P_1, P_2.

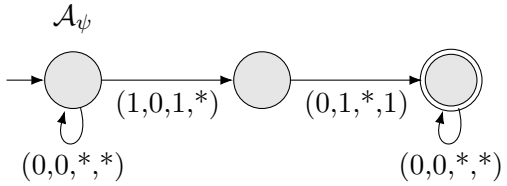
Wir beginnen mit den Automaten für die atomaren Teilformeln $X \subseteq P_1$, $Y \subseteq P_2$, $\mathsf{succ}(X) = Y$, welche jeweils zwei freie Variablen bzw. Relationssymbole haben; also ist das Alphabet ebenfalls Σ_2. Wir schreiben ab jetzt die Bitstrings horizontal, also z. B. $(0,1)$ statt $\binom{0}{1}$.



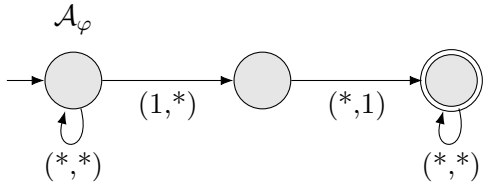
Um den Automaten für die Konjunktion ψ zu bilden, müssen wir zuerst für die bisherigen Automaten das Alphabet auf Σ_4 erweitern, denn in ψ kommen alle 4 Relationsvariablen und -symbole X, Y, P_1, P_2 vor. Da die hinzugekommenen Bits beliebige Werte annehmen können, schreiben wir an diesen Stellen der Einfachheit halber „$*$“. Beispielsweise steht „$(0, *, 0, *)$“ für „$(0, 0, 0, 0)$, $(0, 0, 0, 1)$, $(0, 1, 0, 0)$, $(0, 1, 0, 1)$“. Man beachte, dass diese Positionen von der jeweils hinzugenommenen Variable abhängen (also Y, P_2 und damit Position 2,4 bei $\mathcal{A}'_{X \subseteq P_1}$, X, P_1 und damit Position 1,3 bei $\mathcal{A}'_{Y \subseteq P_2}$ usw.).



Nun können wir den Produktautomaten der drei obigen Automaten bilden (die Produktkonstruktion lässt sich bequem auf mehr als 2 Automaten übertragen):



Im letzten Schritt „verarbeiten“ wir die beiden Existenzquantoren gleichzeitig und bilden den Automaten für φ durch Projektion auf die zwei in φ vorkommenden Relationssymbole P_1, P_2 (also auf Positionen 3 und 4 in den Bitstrings):



Offensichtlich erkennt dieser NEA die Sprache $L(\varphi)$.

T4.14 Beispiele sternfreier Sprachen

$$\Sigma^* = \overline{\emptyset} \qquad 1^*0^* = \overline{\Sigma^*01\Sigma^*}$$

$$\{\varepsilon\} = \overline{\bigcup_{a\in\Sigma} \{a\}\cdot\Sigma^*} \qquad (10)^* = \Big(\overline{\Sigma^*00\Sigma^*} \cap \overline{\Sigma^*11\Sigma^*} \cap 1\Sigma^*0\Big) \cup \{\varepsilon\}$$

Dabei bedeuten

$\overline{\Sigma^*01\Sigma^*}$	Infix 01 ist verboten
$\overline{\Sigma^*00\Sigma^*}$	Infix 00 ist verboten, d. h. auf jede 0 folgt eine 1
$\overline{\Sigma^*11\Sigma^*}$	Infix 11 ist verboten, d. h. auf jede 1 folgt eine 0
$1\Sigma^*0$	das Wort beginnt mit 1 und endet mit 0

T4.15 Beispiele für LTL-Formeln im Verifikations-Szenario

- „Der Wartezustand wird nur durch Ausführen des kritischen Bereichs beendet“:

$$\bigwedge_{i\in\{1,2\}} \Box\Big(W_i \;\to\; \bigcirc(W_i \vee A_i)\Big)$$

- „ Wenn Prozess i im kritischen Abschnitt ist und Prozess $\bar{i} = 3 - i$ darauf wartet, dann betritt Prozess $\bar{i}$ den kritischen Bereich, sobald Prozess i ihn verlässt“:

$$\bigwedge_{i\in\{1,2\}} \Box\Big((A_i \wedge W_{\bar{i}}) \;\to\; A_i \,\mathcal{U}\, A_{\bar{i}}\Big)$$

T4.16 Beispiele zur LTL-Semantik; Äquivalenzen

Zeichnet das Folgende zur Veranschaulichung selbst in lineare Strukturen ein:

- Wenn $\mathfrak{A}, 2 \models p$ und $\mathfrak{A}, n \not\models p$ für alle $n \neq 2$, dann $\mathfrak{A}, 1 \models \bigcirc p$ und $\mathfrak{A}, n \not\models \bigcirc p$ für alle $n \neq 1$.
- Wenn $\mathfrak{A}, 4 \models p$ und $\mathfrak{A}, n \not\models p$ für alle $n \neq 4$, dann $\mathfrak{A}, n \models \Diamond p$ für alle $n \leq 4$ und $\mathfrak{A}, n \not\models \Diamond p$ für alle $n > 4$.
- Wenn $\mathfrak{A}, n \models p$ für alle $n \geq 3$ und $\mathfrak{A}, n \not\models p$ für alle $n < 3$, dann $\mathfrak{A}, n \models \Box p$ für alle $n \geq 3$ und $\mathfrak{A}, n \not\models \Diamond p$ für alle $n < 3$.
- Wenn $\mathfrak{A}, n \models p$ für alle $1 \leq n \leq 3$ und $\mathfrak{A}, 4 \models q$, dann $\mathfrak{A}, n \models p \,\mathcal{U}\, q$ für alle $1 \leq n \leq 4$.

Es gelten folgende Äquivalenzen:

$$\Diamond\varphi \equiv \mathtt{true}\,\mathcal{U}\,\varphi \qquad \text{mit } \mathtt{true} = p \vee \neg p$$
$$\Box\varphi \equiv \neg\Diamond\neg\varphi$$

Es genügen also eigentlich $\bigcirc$ und $\mathcal{U}$ als einzige temporale Operatoren.

T4.17 Beispiel für initiale Äquivalenz

Die LTL-Formel

$$\varphi = \Box(p \to p\,\mathcal{U}\,q)$$

besagt, dass für jeden ($\Box$) Zeitpunkt n gilt: wenn $\mathfrak{A}, n \models p$, dann gibt es einen Zeitpunkt $m \geq n$ mit $\mathfrak{A}, m \models q$ und für alle Zeitpunkte k „dazwischen“ ($\mathcal{U}$) gilt $\mathfrak{A}, k \models p$. Sie ist initial äquivalent zur F1S-Formel (mit einer freien Variablen, wie üblich unterstrichen)

$$\psi(\underline{x}) = \forall y \Big(\big(y \geq \underline{x} \wedge P(y)\big) \to \exists z \Big(z \geq y \wedge Q(z) \wedge \forall u \,\big(y \leq u < z \to P(u)\big)\Big)\Big),$$

wobei „$y \geq x$“ für „$x < y \vee x = y$“ steht; analog für „$z \geq y$“ und „$y \leq u < z$“.

T4.18 Beweis des Lemmas LTL → F1S

Lemma 4.19. Zu jeder LTL-Formel φ existiert eine initial äquivalente F1S-Formel $\psi(x)$.

Beweis. Wir geben eine Übersetzung an, die jeder LTL-Formel φ eine S1S-Formel $\widehat{\varphi}(x)$ mit einer freien Variable zuordnet. Dabei gehen wir induktiv über die Struktur von φ vor.

$$\begin{aligned}
\widehat{p_i}(x) &= P_i(x) \\
\widehat{\neg\varphi}(x) &= \neg\widehat{\varphi}(x) \\
\widehat{\varphi \wedge \psi}(x) &= \widehat{\varphi}(x) \wedge \widehat{\psi}(x) \\
\widehat{\varphi \vee \psi}(x) &= \widehat{\varphi}(x) \vee \widehat{\psi}(x) \\
\widehat{\bigcirc\varphi}(x) &= \exists y\,(y = s(x) \wedge \widehat{\varphi}(y)) \\
\widehat{\Diamond\varphi}(x) &= \exists y\,(x \leq y \wedge \widehat{\varphi}(y)) \\
\widehat{\Box\varphi}(x) &= \forall y\,(x \leq y \to \widehat{\varphi}(y)) \\
\widehat{\varphi\,\mathcal{U}\,\psi}(x) &= \exists y \Big(x \leq y \wedge \widehat{\psi}(y) \wedge \forall z\,(x \leq z < y \to \widehat{\varphi}(z))\Big)
\end{aligned}$$

In den letzten 4 Fällen ist dabei y jeweils eine neue Variable, und die Atome mit „$\leq$“ sind dabei wieder Abkürzungen wie im vorangehenden Beispiel.

Man kann nun leicht per Induktion zeigen, dass für alle LTL-Formeln φ gilt: φ ist initial äquivalent zu $\widehat{\varphi}(x)$. ❏

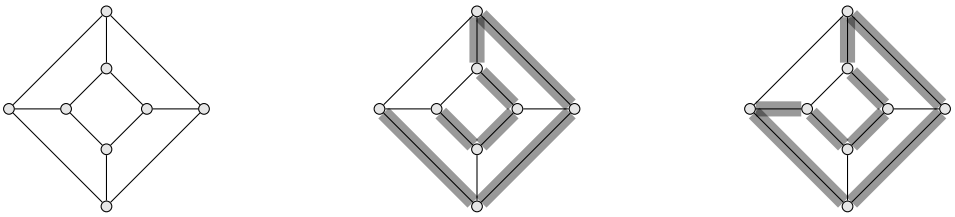
T4.19 Zur ESO-Definierbarkeit von „Hamiltonkreis“

Die folgende Formel definiert die Klasse HK aller ungerichteten Graphen, die einen Hamiltonkreis haben.

$$\begin{aligned}\varphi_{\mathsf{HK}} = \exists L\, \exists S \Big(& L \text{ ist strikte lineare Ordnung } \wedge \\ & \text{alle Elemente des Universums kommen in } L \text{ vor } \wedge \\ & S \text{ ist die } \textit{direkte} \text{ Nachfolgerrelation von } L \text{ und} \\ & \quad \text{verbindet zusätzlich das größte } L\text{-Element mit dem kleinsten } \wedge \\ & \forall x\, \forall y\, \big(S(x,y) \to E(x,y)\big)\Big)\end{aligned}$$

Dabei sind L und S binäre Relationsvariablen; alle Konjunkte sind offensichtlich durch FO-Formeln definierbar, die L und S wie Relationsvariablen behandeln.

Um dies an einem Beispiel zu illustrieren, betrachte den ungerichteten Graphen G im linken Teil des unteren Bildes (den man äquivalent als Struktur $\mathfrak{A}$ mit symmetrischer binärer Relation E auffassen kann). Dieser Graph hat einen Hamiltonkreis: im mittleren Teil des Bildes ist die strikte lineare Ordnung L eingezeichnet, allerdings der Übersichtlichkeit halber ohne transitive Hülle. Im rechten Teil ist die Relation S eingezeichnet, die gleichzeitig den Hamiltonkreis repräsentiert.



Wir wollen nun beweisen, dass HK durch φ_{HK} definiert wird, also dass gilt:

$$\mathfrak{A} = (A, E^{\mathfrak{A}}) \in \mathsf{HK} \quad \text{gdw.} \quad \mathfrak{A} \models \varphi_{\mathsf{HK}}$$

Beweis.

„⇒“ Angenommen, der ungerichtete Graph $\mathfrak{A} = (A, E^{\mathfrak{A}})$ enthält einen Hamiltonkreis

$$H = \{\{a_0, a_1\},\ \ldots,\ \{a_{n-1}, a_n\},\ \{a_n, a_0\}\}.$$

Erweitere nun $\mathfrak{A}$ durch die binären Relationen

$$\begin{aligned} S^{\mathfrak{A}} &:= \{(a_0, a_1),\ \ldots,\ (a_{n-1}, a_n),\ (a_n, a_0)\} \quad \text{(wie } H\text{, aber gerichtet)} \\ L^{\mathfrak{A}} &:= \text{transitive Hülle von } S^{\mathfrak{A}} \setminus \{(a_n, a_0)\}. \end{aligned}$$

Man prüft nun leicht, dass alle FO-Konjunkte von φ_{HK} in $\mathfrak{A}$ erfüllt sind. Deshalb wird durch $S^{\mathfrak{A}}$ und $L^{\mathfrak{A}}$ eine Zuweisung β bestimmt, unter der die gesamte Konjunktion, als SO-Formel aufgefasst, erfüllt ist. Folglich gilt $\mathfrak{A} \models \varphi_{\mathsf{HK}}$.

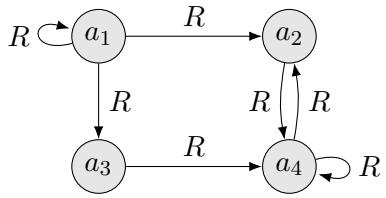
„⇐“ Angenommen $\mathfrak{A} \models \varphi_{\mathsf{HK}}$. Seien $S^{\mathfrak{A}}$ und $L^{\mathfrak{A}}$ Relationen, so dass alle FO-Formeln in φ_{HK} erfüllt sind. Dann ist $S^{\mathfrak{A}}$ ein Hamiltonkreis in $\mathfrak{A} = (A, E^{\mathfrak{A}})$:

- $S^{\mathfrak{A}}$ ist ein Kreis, da $S^{\mathfrak{A}}$ die direkte Nachfolgerrelation der strikten linearen Ordnung $L^{\mathfrak{A}}$ ist und zusätzlich deren größtes Element mit dem kleinsten verbindet.
- $S^{\mathfrak{A}}$ enthält nur Kanten aus E.
- $S^{\mathfrak{A}}$ enthält *alle* Knoten aus A, weil $L^{\mathfrak{A}}$ alle Knoten enthält.
- $S^{\mathfrak{A}}$ enthält jeden Knoten höchstens einmal, weil anderenfalls $L^{\mathfrak{A}}$ keine *strikte* lineare Ordnung wäre.

❑

T4.20 Beispiel für die Kodierung einer Struktur als Wort

Sei $\mathfrak{A} = (A, R^{\mathfrak{A}})$ eine Struktur mit einem zweistelligen Relationssymbol R wie folgt.



Dann wird die Relation $R^{\mathfrak{A}}$ durch folgendes Wort kodiert:

$$w_R = 1\,1\,1\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,1$$

Dabei stehen die 16 Bits in dieser Reihenfolge für die Paare

$$(a_1, a_1) \in R^{\mathfrak{A}},\ \ (a_1, a_2) \in R^{\mathfrak{A}},\ \ (a_1, a_3) \in R^{\mathfrak{A}},\ \ \ldots,\ \ (a_4, a_3) \notin R^{\mathfrak{A}},\ \ (a_4, a_4) \in R^{\mathfrak{A}}.$$

Die gesamte Struktur wird dann so kodiert:

$$\underbrace{0\,0\,0\,0\,1}_{|A|=4}\ \underbrace{0\,0\,1}_{R\text{ binär}}\ w_R$$

T4.21 Datenkomplexität von ESO

Lemma 4.27. Jedes ESO-definierbare Problem K ist in NP.

Beweis. Sei K definierbar durch einen ESO-Satz φ_K. Das Entscheidungsproblem K ist dann äquivalent zu folgendem Problem:

Gegeben: Struktur $\mathfrak{A}$
Frage : Gilt $\mathfrak{A} \models \varphi_K$?

Dies ist nichts anderes als das Auswertungsproblem, wobei φ_K fest ist, also unabhängig von der Eingabe, Es genügt also zu zeigen, dass ESO-Auswertung bezüglich Datenkomplexität in NP ist (deshalb der Zusatz „mit anderen Worten“ auf der Folie).

Sei $\varphi = \exists R_1 \cdots \exists R_n \, \varphi$ der fixierte auszuwertende ESO-Satz mit $\varphi \in \mathsf{FO}$. Wir verwenden den folgenden Algorithmus, der eine Struktur $\mathfrak{A}$ als Eingabe verwendet:

1. Bestimme nichtdeterministisch Relationen $R_1^{\mathfrak{A}}, \ldots, R_n^{\mathfrak{A}}$ passender Stelligkeit.
2. Überprüfe mit dem Auswertungsalgorithmus `ausw` aus Teil 2 (Prädikatenlogik 1. Stufe), ob die um diese Relationen erweiterte Struktur $\mathfrak{A}$ den FO-Satz φ wahr macht. Wenn ja, gib „erfüllt“ aus, sonst „nicht erfüllt“.

Der zweite Schritt ist deterministisch; da φ fixiert ist, kann er in Polynomialzeit ausgeführt werden (siehe Datenkomplexität von FO). ❏

T4.22 Repräsentation der Schrittzähler und Bandpositionen

Wenn die Eingabestruktur $\mathfrak{A}$ durch ein Wort $w_{\mathfrak{A}}$ repräsentiert wird, kann Polyzeit-Turingmaschine M maximal $|w_{\mathfrak{A}}|^k$ viele Berechnungsschritte durchführen. Für jeden Schritt muss die aktuelle Konfiguration festgehalten werden, die neben dem aktuellen Zustand auch aus dem Schrittzähler (Werte $0, \ldots, |w_{\mathfrak{A}}|^k - 1$) und dem Inhalt des Bandes an den maximal $|w_{\mathfrak{A}}|^k$ Bandpositionen besteht. Da es in $\mathfrak{A}$ nur $|A|$ viele Elemente gibt, müssen wir Schrittzähler und und Bandpositionen durch *Tupel* $\overline{a}$ von Elementen von $\mathfrak{A}$ ausdrücken. Dafür wählen wir k' so, dass $|A|^{k'} \leq |w_{\mathfrak{A}}|^k$, also dass es mindestens so viele Tupel gibt wie benötigt.

Um in der Struktur festzuhalten, welches Tupel welcher Zahl aus $\{0, \ldots, |w_{\mathfrak{A}}|^k - 1\}$ entspricht, verwenden wir zusätzlich eine strikte lineare Ordnung über $A^{k'}$. Diese repräsentieren wir mittels einer $2k'$-stelligen Relationsvariable L wie folgt.

$$\varphi_{\mathsf{lin}} := \exists L \Big(\begin{array}{l} L \text{ ist irreflexiv, antisymmetrisch, transitiv} \\ \quad \wedge \quad \forall \overline{x} \forall \overline{y} \left(\overline{x} \neq \overline{y} \rightarrow \big(L(\overline{x}, \overline{y}) \vee L(\overline{y}, \overline{x}) \big) \right) \end{array} \Big),$$

wobei $\overline{x}, \overline{y}$ jeweils k'-Tupel von Variablen sind und die fehlenden FO-Formeln wieder leicht einzusetzen sind. Um nun mittels L die Zahlen $0, \ldots, |w_{\mathfrak{A}}|^k - 1$ zu repräsentieren, benötigt man noch die Nachfolgerrelation auf L, die man wie folgt definieren kann:

$$\varphi_{\mathsf{NF}} := L(\overline{x}, \overline{y}) \wedge \neg \exists \overline{z} \, \big(L(\overline{x}, \overline{z}) \wedge L(\overline{z}, \overline{y}) \big)$$

Nun können wir mittels eines Tupels $\overline{x}$ Schrittzähler und Bandposition repräsentieren.

T4.23 Bestandteile der Formel ψ im Beweis Satz von Fagin

- $\boldsymbol{\psi_{\mathsf{ok}}}$

  Dieses Konjunkt soll sicherstellen, dass Bandsymbole und Zustände eindeutig sind, d. h., an jeder Bandposition (repräsentiert durch ein k'-Tupel $\overline{p}$) steht zu jedem Zeitpunkt der Berechnung (Schrittzähler, repräsentiert durch ein k'-Tupel $\overline{t}$) höchstens ein Zeichen a_i des Arbeitsalphabets (repräsentiert durch eine k'-stellige Relationsvariable T_{a_i}), und wenn dort der Schreib-Lese-Kopf steht, dann in höchstens einem Zustand (repräsentiert durch eine k'-stellige Relationsvariable H_{q_i}); außerdem kann der Schreib-Lese-Kopf dann an keiner weiteren Position $\overline{p}'$ stehen. Dies kann wie folgt ausgedrückt werden:

$$\begin{aligned}\psi_{\mathsf{ok}} \;=\; & \forall\overline{p}\;\forall\overline{t}\;\left(\bigvee_{1\le i\le \ell}\left(T_{a_i}(\overline{p},\overline{t})\wedge\bigwedge_{\substack{1\le j\le \ell\\ j\neq i}}\neg T_{a_j}(\overline{p},\overline{t})\right)\right)\;\wedge\\ & \forall\overline{t}\;\exists\overline{p}\;\left(\bigvee_{1\le i\le m}\left(H_{q_i}(\overline{p},\overline{t})\wedge\bigwedge_{\substack{1\le j\le m\\ j\neq i}}\neg H_{q_j}(\overline{p},\overline{t})\right)\right.\\ & \left.\wedge\;\;\forall\overline{p}'\;\left(\overline{p}\neq\overline{p}'\rightarrow\bigwedge_{1\le i\le m}\neg H_{q_i}(\overline{p}',\overline{t})\right)\right)\end{aligned}$$

- $\boldsymbol{\psi_{\mathsf{move}}}$

  Dieses Konjunkt drückt aus, dass zwei aufeinander folgende Konfigurationen entweder der Übergangsrelation Δ der Maschine entsprechen oder in einem akzeptierenden oder verwerfenden Zustand verbleiben (falls vor Erreichen der n^k Schritte ein solcher Zustand erreicht wird, wird er aus technischen Gründen in den folgenden Zeitpunkten einfach beibehalten):

$$\psi_{\mathsf{move}} \;=\; \forall\overline{t}\;\forall\overline{t}'\left(\varphi_{\mathsf{nf}}(\overline{t},\overline{t}')\;\rightarrow\bigvee_{(q,a,q',b,M)\in\Delta}\alpha_{q,a,q',b,M}(\overline{t},\overline{t}')\;\vee\;\psi_{\mathsf{term}}(\overline{t},\overline{t}')\right),$$

  wobei

$$\alpha_{q,a,q',b,\boldsymbol{R}}(\overline{t},\overline{t}') \;=\; \exists\overline{p}\;\exists\overline{p}'\left(H_q(\overline{p},\overline{t})\wedge T_a(\overline{p},\overline{t})\wedge H_{q'}(\overline{p}',\overline{t}')\wedge T_b(\overline{p}',\overline{t}')\wedge\boldsymbol{\varphi_{\mathsf{nf}}(\overline{t},\overline{t}')}\right),$$

$$\alpha_{q,a,q',b,\boldsymbol{L}}(\overline{t},\overline{t}') \;=\; \exists\overline{p}\;\exists\overline{p}'\left(H_q(\overline{p},\overline{t})\wedge T_a(\overline{p},\overline{t})\wedge H_{q'}(\overline{p}',\overline{t}')\wedge T_b(\overline{p}',\overline{t}')\wedge\boldsymbol{\varphi_{\mathsf{nf}}(\overline{t}',\overline{t})}\right)$$

  und

$$\psi_{\mathsf{term}}(\overline{t},\overline{t}') \;=\; \exists\overline{p}\bigvee_{q\in Q_A\cup Q_R}\left(H_q(\overline{p},\overline{t})\wedge H_q(\overline{p},\overline{t}')\right).$$

- ψ_{acc}

  Für dieses Konjunkt genügt wegen der obigen Beobachtung über akzeptierende (und verwerfende) Zustände:

$$\psi_{\mathsf{acc}} \;=\; \exists \overline{p}\; \exists \overline{t} \bigvee_{q \in Q_A} \Big(H_q(\overline{p}, \overline{t}) \Big)$$

- ψ_{const}

  Dieses Konjunkt lässt sich leicht analog konstruieren; probiert es selbst aus.

- ψ_{ini}

  Dies ist das technisch aufwändigste Konjunkt, denn dazu müssen Addition und Multiplikation von natürlichen Zahlen ESO-definiert werden. Weitere Details kann man z. B. in [EF99, Imm99] nachlesen.

Anhang

Griechische Buchstaben

Kleinbuchstaben

α ..... alpha
β ..... beta
γ ..... gamma
δ ..... delta
ϵ ..... epsilon
ζ ..... zeta
η ..... eta
ϑ, θ ..... theta
ι ..... iota
κ ..... kappa
λ ..... lambda
μ ..... my
ν ..... ny
ξ xi
o ..... omikron
π ..... pi
ρ ..... rho
σ, ς ..... sigma
τ ..... tau
υ ..... ypsilon
φ, ϕ ..... phi
χ ..... chi
ψ ..... psi
ω ..... omega

Großbuchstaben

Γ ..... Gamma
Δ ..... Delta
Θ ..... Theta
Λ ..... Lambda
Π ..... Pi
Ξ Xi
Σ ..... Sigma
Υ ..... Ypsilon
Φ ..... Phi
Ψ ..... Psi
Ω ..... Omega

Die übrigen griechischen Großbuchstaben werden genauso geschrieben wie die entsprechenden lateinischen.

Literaturverzeichnis

[AB09] Arora, S. und B. Barak: *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.

[EF99] Ebbinghaus, H.-D. und J. Flum: *Finite Model Theory*. Springer, 1999.

[Grä19] Grädel, E.: *Mathematische Logik*. Vorlesungsskript, RWTH Aachen, 2019. `https://tinyurl.com/sose19-logik-graedel`.

[Imm99] Immerman, Neil: *Descriptive Complexity*. Springer, 1999.

[Sch19] Schneider, T.: *Theoretische Informatik 1 + 2*. Vorlesungsskript, 2019. siehe Stud.IP.