Vorlesung
Logik und Komplexität

Wintersemester 2025/26

Prof. Dr. Nicole Schweikardt

Lehrstuhl Logik in der Informatik
Institut für Informatik
Humboldt-Universität zu Berlin

Kapitel 0:

Einleitung

Logiken zur Beschreibung von Berechnungsproblemen

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$$\mathcal{G} = (V, E) \text{ ist 3-färbbar}$$

$$\iff$$

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$$\mathcal{G} = (V, E) \text{ ist 3-färbbar}$$

$$\iff \text{es gibt } R \subseteq V, G \subseteq V, B \subseteq V, \text{ so dass}$$

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$$\mathcal{G} = (V, E) \text{ ist 3-färbbar}$$

$$\iff \text{ es gibt } R \subseteq V, G \subseteq V, B \subseteq V, \text{ so dass } V = R \,\dot{\cup}\, G \,\dot{\cup}\, B \text{ und}$$

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$\mathcal{G} = (V, E)$ ist 3-färbbar

$\iff$ es gibt $R \subseteq V, G \subseteq V, B \subseteq V$, so dass $V = R \,\dot\cup\, G \,\dot\cup\, B$ und für alle $u, v \in V$ gilt: falls $(u, v) \in E$, so gilt

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$\mathcal{G} = (V, E)$ ist 3-färbbar

$\iff$ es gibt $R \subseteq V, G \subseteq V, B \subseteq V$, so dass $V = R \mathbin{\dot\cup} G \mathbin{\dot\cup} B$ und für alle $u, v \in V$ gilt: falls $(u, v) \in E$, so gilt

$$\neg\Big((u \in R \wedge v \in R) \vee (u \in G \wedge v \in G) \vee (u \in B \wedge v \in B) \Big)$$

Beispiel 0.1 (3-Färbbarkeit von Graphen)

Ein Graph $\mathcal{G} = (V, E)$ heißt 3-färbbar, falls seine Knoten so mit 3 Farben gefärbt werden können, dass benachbarte Knoten verschiedene Farben haben.

Somit gilt:

$$\mathcal{G} = (V, E) \text{ ist 3-färbbar}$$

$$\begin{aligned} \Leftrightarrow\ & \text{es gibt } R \subseteq V, G \subseteq V, B \subseteq V, \text{ so dass } V = R \,\dot\cup\, G \,\dot\cup\, B \text{ und} \\ & \text{für alle } u, v \in V \text{ gilt: falls } (u, v) \in E, \text{ so gilt} \\ & \neg\Big((u \in R \wedge v \in R) \vee (u \in G \wedge v \in G) \vee (u \in B \wedge v \in B) \Big) \end{aligned}$$

$$\Leftrightarrow\ \mathcal{G} \models \Phi_{\text{3-COL}}, \text{ wobei}$$

$$
\begin{aligned}
\Phi_{\text{3-COL}} \ :=\ \exists R\, \exists G\, \exists B \Big(\\
& \forall v \Big(\ \big(R(v) \land \neg G(v) \land \neg B(v)\big) \lor \\
& \qquad \big(\neg R(v) \land G(v) \land \neg B(v)\big) \lor \\
& \qquad \big(\neg R(v) \land \neg G(v) \land B(v)\big) \ \Big) \ \land \\
& \forall u\, \forall v \Big(\ E(u,v) \rightarrow \\
& \qquad \neg\Big(\big(R(u) \land R(v)\big) \lor \big(G(u) \land G(v)\big) \lor \big(B(u) \land B(v)\big) \Big) \Big) \Big)
\end{aligned}
$$

$$
\begin{aligned}
\Phi_{\text{3-COL}} \ := \ \exists R\, \exists G\, \exists B \Bigg(\\
& \forall v \Big(\quad (R(v) \land \neg G(v) \land \neg B(v)) \lor \\
& \qquad\quad (\neg R(v) \land G(v) \land \neg B(v)) \lor \\
& \qquad\quad (\neg R(v) \land \neg G(v) \land B(v)) \quad \Big) \ \land \\
& \forall u\, \forall v \Big(\ E(u,v) \rightarrow \\
& \qquad\quad \neg\Big((R(u) \land R(v)) \lor (G(u) \land G(v)) \lor (B(u) \land B(v)) \Big) \Big) \Bigg)
\end{aligned}
$$

$\Phi_{\text{3-COL}}$ ist eine Formel der Logik zweiter Stufe (SO), die eine wichtige Rolle in dieser Vorlesung spielt. Wir werden z.B. Folgendes nachweisen:

Satz von Fagin:

Ein Problem gehört genau dann zur Komplexitätsklasse NP*, wenn es durch einen Satz der existentiellen Logik zweiter Stufe* (ESO) *beschrieben werden kann.*

Satz von Fagin:
Ein Problem gehört genau dann zur Komplexitätsklasse NP, *wenn es durch einen Satz der existentiellen Logik zweiter Stufe* (ESO) *beschrieben werden kann.*

Bemerkung

Unter Verwendung von Ehrenfeucht-Fraïssé-Spielen und logischen Reduktionen werden wir zeigen, dass 3-Färbbarkeit von Graphen nicht durch einen Satz der Logik erster Stufe (FO) beschrieben werden kann.

Beispiel 0.2 (Erreichbarkeit)

Wir betrachten das Berechnungsproblem

> *Erreichbarkeit*
>
> *Eingabe:* ein Graph $\mathcal{G} = (V, E)$ und zwei Knoten $s, t \in V$
>
> *Frage:* Gibt es in $\mathcal{G}$ einen Weg von s nach t?

Beispiel 0.2 (Erreichbarkeit)

Wir betrachten das Berechnungsproblem

Erreichbarkeit
Eingabe: ein Graph $\mathcal{G} = (V, E)$ und zwei Knoten $s, t \in V$
Frage: Gibt es in $\mathcal{G}$ einen Weg von s nach t?

Dann gilt:

$$\text{Antwort „ja“} \iff (\mathcal{G}, s, t) \models \Phi_{\text{REACH}}, \text{ wobei}$$

$$\Phi_{\text{REACH}} := [\text{lfp}_{R,x}\varphi](t) \text{ mit}$$

$$\varphi := \Big(x = s \ \lor \ \exists z \ \big(R(z) \land E(z, x)\big)\Big)$$

Zur Auswertung von Φ_{REACH} in $(\mathcal{G}, s, t)$ wird induktiv eine 1-stellige Relation R definiert:

- Starte mit $R := \emptyset$
- Iteriere so lange, bis sich nichts mehr ändert:
  - $R_{\text{neu}} := \{x \in V \, : \, (\mathcal{G}, s, t, R, x) \models \varphi\}$
  - $R := R_{\text{neu}}$

Zur Auswertung von Φ_{REACH} in $(\mathcal{G}, s, t)$ wird induktiv eine 1-stellige Relation R definiert:

- Starte mit $R := \emptyset$
- Iteriere so lange, bis sich nichts mehr ändert:
  - $R_{\text{neu}} := \{x \in V : (\mathcal{G}, s, t, R, x) \models \varphi\}$
  - $R := R_{\text{neu}}$

Φ_{REACH} ist eine Formel der kleinsten Fixpunktlogik (LFP). Fixpunktlogiken dienen der Charakterisierung der Komplexitätsklassen P und PSPACE.

Logiken zur Beschreibung formaler Sprachen

Wir betrachten Worte über dem Alphabet $\Sigma = \{a, b\}$ und repräsentieren nicht-leere Worte $w \in \Sigma^*$ durch Strukturen über der Signatur $\sigma_\Sigma := \{\leqslant, P_a, P_b\}$, wobei $\leqslant$ 2-stellig ist, und P_a sowie P_b 1-stellig sind.

Wir betrachten Worte über dem Alphabet $\Sigma = \{a, b\}$ und repräsentieren nicht-leere Worte $w \in \Sigma^*$ durch Strukturen über der Signatur $\sigma_\Sigma := \{\leqslant, P_a, P_b\}$, wobei $\leqslant$ 2-stellig ist, und P_a sowie P_b 1-stellig sind.

Beispiel

Das Wort $w = aabab$ wird durch die Wortstruktur $\mathcal{A}_w$ mit Universum $\{1, 2, 3, 4, 5\}$ repräsentiert, wobei $\leqslant^{\mathcal{A}_w}$ die natürliche Ordnung auf dem Universum ist und $P_a^{\mathcal{A}_w} = \{1, 2, 4\}$ bzw. $P_b^{\mathcal{A}_w} = \{3, 5\}$ die Menge der Positionen von w ist, an denen der Buchstabe a bzw. der Buchstabe b steht.

Beispiel 0.3

Die durch den regulären Ausdruck a^+b^* beschriebene Sprache $L_{a^+b^*}$ wird durch die FO[σ_Σ]-Formel

$$\varphi_{a^+b^*} :=$$

beschrieben, d.h., es gilt für jedes nicht-leere Wort $w \in \Sigma^*$:

$$w \in L_{a^+b^*} \iff \mathcal{A}_w \models \varphi_{a^+b^*}.$$

Beispiel 0.3

Die durch den regulären Ausdruck a^+b^* beschriebene Sprache $L_{a^+b^*}$ wird durch die FO$[\sigma_\Sigma]$-Formel

$$\varphi_{a^+b^*} := \exists x \, \forall y \, \big(P_a(y) \leftrightarrow y \leqslant x \big)$$

beschrieben, d.h., es gilt für jedes nicht-leere Wort $w \in \Sigma^*$:

$$w \in L_{a^+b^*} \iff \mathcal{A}_w \models \varphi_{a^+b^*}.$$

Beispiel 0.3

Die durch den regulären Ausdruck a^+b^* beschriebene Sprache $L_{a^+b^*}$ wird durch die $\mathrm{FO}[\sigma_\Sigma]$-Formel

$$\varphi_{a^+b^*} := \exists x\, \forall y \, \big(P_a(y) \leftrightarrow y \leqslant x \big)$$

beschrieben, d.h., es gilt für jedes nicht-leere Wort $w \in \Sigma^*$:

$$w \in L_{a^+b^*} \iff \mathcal{A}_w \models \varphi_{a^+b^*}.$$

Beispiel 0.4

Die Sprache $L := \{w \in \Sigma^* : \#_a(w) \text{ ist ungerade}\}$ wird durch folgende Formel Φ_L der monadischen Logik zweiter Stufe (MSO) beschrieben. Dabei steht der Ausdruck $\#_a(w)$ für die Anzahl der a's in dem Wort w.

$$
\begin{aligned}
\Phi_L := \exists Z \Bigg(\\
& \forall x \Big(\big(P_a(x) \land \neg\exists y \left(P_a(y) \land y \leqslant x \land \neg\, y{=}x \right) \big) \rightarrow Z(x) \Big) \land \\
& \forall x\, \forall y \Big(\Big(\quad P_a(x) \land P_a(y) \land x \leqslant y \land \neg\, x{=}y \land \\
& \qquad\qquad \neg\exists z \left(x \leqslant z \land z \leqslant y \land \neg\, z{=}x \land \neg\, z{=}y \land P_a(z) \right) \Big) \\
& \qquad\qquad \rightarrow \big(Z(x) \leftrightarrow \neg Z(y) \big) \Big) \land \\
& \forall x \Big(\big(P_a(x) \land \neg\exists y \left(P_a(y) \land x \leqslant y \land \neg\, y{=}x \right) \big) \rightarrow Z(x) \Big) \land \\
& \exists x\, P_a(x) \Bigg)
\end{aligned}
$$

Die Zeile 2 der Formel besagt (in $(\mathcal{A}_w, Z)$), dass die Menge Z die erste Position enthält, die den Buchstaben a trägt.

Die Zeilen 3 bis 5 der Formel besagen, dass die Menge Z „jede zweite“ Position enthält, die den Buchstaben a trägt. Genauer: Sind x und y Positionen, an denen ein a steht und zwischen denen kein weiteres a steht, so gehört genau eine der beiden Positionen x und y zu Z.

Die Zeile 6 der Formel besagt, dass die letzte Position, die den Buchstaben a trägt, zur Menge Z gehört.

Die letzte Zeile 7 der Formel gewährleistet, dass das Wort mindestens ein a enthält.

Insgesamt gilt für jedes nicht-leere Wort $w \in \Sigma^*$:

$$\begin{aligned} \mathcal{A}_w \models \Phi_L \quad &\Longleftrightarrow \quad w \text{ enthält eine ungerade Anzahl von a's} \\ &\Longleftrightarrow \quad w \in L. \end{aligned}$$

Insgesamt gilt für jedes nicht-leere Wort $w \in \Sigma^*$:

$$\begin{aligned} \mathcal{A}_w \models \Phi_L &\iff w \text{ enthält eine ungerade Anzahl von a's} \\ &\iff w \in L. \end{aligned}$$

Wir werden Folgendes nachweisen:

(Teil aus dem) Satz von Büchi:
Eine Sprache $L \subseteq \Sigma^+$ ist genau dann regulär, wenn sie von einem MSO$[\sigma_\Sigma]$*-Satz beschrieben wird.*

Logiken zur Beschreibung von Datenbank-Anfragen

Beispiele dazu wurden bereits in der Veranstaltung „Logik in der Informatik“ betrachet.

Voraussetzungen zur Teilnahme an der Veranstaltung

Die Beherrschung des in der Veranstaltung „Logik in der Informatik“ vermittelten Stoffs ist Voraussetzung für die Teilnahme an der Veranstaltung „Logik und Komplexität“.

Zur Vorbereitung wird dringend empfohlen, insbesondere die Kapitel „Logik erster Stufe“ und „Grundlagen des automatischen Schließens“ des Skripts „Logik in der Informatik“ (N. Schweikardt, HU Berlin, [**S-LI**]) durchzuarbeiten.

Aufbau der Vorlesung „Logik und Komplexität“

Organisatorisches

Website zur Vorlesung:

`https://www.informatik.hu-berlin.de/de/forschung/gebiete/loginf/lehre/wise-2025-26/vorlesung-logik-und-komplexitaet`

Dort finden sich auch Details zum Ablauf des Übungsbetriebs und zur Zulassung zur Modulabschlussprüfung.

Weitere Lektüre

- Lehrbuch [**L**]: Kapitel 1.
- Viele weitere Beispiele zur Bedeutung der Logik in der Informatik finden sich in dem Artikel

  „*On the unusual effectiveness of logic in computer science*“ von Halpern, Harper, Immerman, Kolaitis, Vardi und Vianu, Bulletin of Symbolic Logic 7(2):213-236 (2001)

  Eine Vorabversion des Artikels finden Sie hier.

Kapitel 1:

Grundlagen und der Satz von Trakhtenbrot

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“
- leeres Wort: ε
- $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“
- leeres Wort: ε
- $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$
- f.a. $w \in \Sigma^+$, für $n := |w|$ und f.a. $i \in \{0, 1, \ldots, n-1\}$ schreibe w_i für das Symbol an Position i in w, d.h. $w = w_0 w_1 \ldots w_{n-1}$

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“
- leeres Wort: ε
- $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$
- f.a. $w \in \Sigma^+$, für $n := |w|$ und f.a. $i \in \{0, 1, \ldots, n-1\}$ schreibe w_i für das Symbol an Position i in w, d.h. $w = w_0 w_1 \ldots w_{n-1}$
- Potenzmenge einer Menge M: $\mathcal{P}(M) := \mathrm{Pot}(M) := 2^M := \{X : X \subseteq M\}$

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“
- leeres Wort: ε
- $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$
- f.a. $w \in \Sigma^+$, für $n := |w|$ und f.a. $i \in \{0, 1, \ldots, n-1\}$ schreibe w_i für das Symbol an Position i in w, d.h. $w = w_0 w_1 \ldots w_{n-1}$
- Potenzmenge einer Menge M: $\mathcal{P}(M) := \mathrm{Pot}(M) := 2^M := \{X : X \subseteq M\}$
- A, B Mengen, $f : A \to B$, $\overline{a} = (a_1, \ldots, a_k) \in A^k$
  $\rightsquigarrow\ f(\overline{a}) := \big(f(a_1), \ldots, f(a_k)\big) \in B^k$

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“
- leeres Wort: ε
- $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$
- f.a. $w \in \Sigma^+$, für $n := |w|$ und f.a. $i \in \{0, 1, \ldots, n-1\}$ schreibe w_i für das Symbol an Position i in w, d.h. $w = w_0 w_1 \ldots w_{n-1}$
- Potenzmenge einer Menge M: $\mathcal{P}(M) := \mathrm{Pot}(M) := 2^M := \{X : X \subseteq M\}$
- A, B Mengen, $f : A \to B$, $\overline{a} = (a_1, \ldots, a_k) \in A^k$
  $\rightsquigarrow\ f(\overline{a}) := \big(f(a_1), \ldots, f(a_k)\big) \in B^k$
- $R \subseteq A^k \ \rightsquigarrow\ f(R) := \{f(\overline{a}) : \overline{a} \in R\}$

Notation

- s.d. : „so dass“
- f.a. : „für alle“
- ex. : „es existiert / es gibt“
- leeres Wort: ε
- $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$
- f.a. $w \in \Sigma^+$, für $n := |w|$ und f.a. $i \in \{0, 1, \dots, n-1\}$ schreibe w_i für das Symbol an Position i in w, d.h. $w = w_0 w_1 \dots w_{n-1}$
- Potenzmenge einer Menge M: $\mathcal{P}(M) := \text{Pot}(M) := 2^M := \{X : X \subseteq M\}$
- A, B Mengen, $f : A \to B$, $\overline{a} = (a_1, \dots, a_k) \in A^k$
  $\rightsquigarrow f(\overline{a}) := \big(f(a_1), \dots, f(a_k)\big) \in B^k$
- $R \subseteq A^k \rightsquigarrow f(R) := \{f(\overline{a}) : \overline{a} \in R\}$
- natürliche Zahlen:

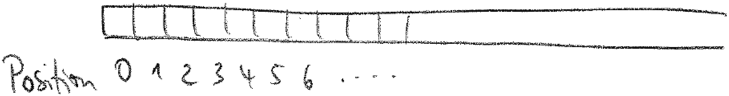
$$
\begin{aligned}
\mathbb{N} &:= \{0, 1, 2, \dots\} \\
\mathbb{N}_{\geqslant 1} &:= \mathbb{N} \setminus \{0\}
\end{aligned}
$$

Syntax und Semantik der Logik erster Stufe (FO)

siehe Skript zur Vorlesung „Logik in der Informatik“ (N. Schweikardt, HU Berlin, [**S-LI**])

Turingmaschinen

Intuitiv stellt eine Turingmaschine (TM) ein Band dar, welches linksseitig begrenzt ist:



Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

- einer endlichen Menge Q von Zuständen

Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

- einer endlichen Menge Q von Zuständen
- einem endlichen Arbeitsalphabet Γ
  mit ausgezeichnetem Blank-Symbol $\Box$

Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

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

Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

- einer endlichen Menge Q von Zuständen
- einem endlichen Arbeitsalphabet Γ
  mit ausgezeichnetem Blank-Symbol $\Box$
- einem Eingabealphabet $\Sigma \subseteq \Gamma \setminus \{\Box\}$
- einem Anfangszustand $q_0 \in Q$

Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

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

Definition 1.1 (Turingmaschine)

Eine nichtdeterministische Turingmaschine (NTM)

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

besteht aus

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

M heißt deterministisch (kurz: M ist eine DTM), falls

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

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

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

Definition 1.2 (Konfiguration einer TM)

(a) Eine Konfiguration einer TM $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ ist

Definition 1.2 (Konfiguration einer TM)

(a) Eine Konfiguration einer TM $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ ist ein Tripel

$$\mathrm{C} = (q, p, u) \in Q \times \mathbb{N} \times \Gamma^* \quad \text{mit } p < |u|.$$

Idee: $\mathrm{C} = (q, p, u)$ gibt an, dass die TM sich im Zustand q befindet, der Kopf an Position p steht und die Inschrift des Arbeitsbandes u gefolgt von Blank-Symbolen ist.

Definition 1.2 (Konfiguration einer TM)

(a) Eine Konfiguration einer TM $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ ist ein Tripel

$$\mathrm{C} = (q, p, u) \in Q \times \mathbb{N} \times \Gamma^* \quad \text{mit } p < |u|.$$

Idee: $\mathrm{C} = (q, p, u)$ gibt an, dass die TM sich im Zustand q befindet, der Kopf an Position p steht und die Inschrift des Arbeitsbandes u gefolgt von Blank-Symbolen ist.

(b) $\mathcal{C}_M := \{(q, p, u) \in Q \times \mathbb{N} \times \Gamma^* : p < |u|\}$ bezeichnet die Menge aller möglichen Konfigurationen von M.

Definition 1.2 (Konfiguration einer TM)

(a) Eine Konfiguration einer TM $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ ist ein Tripel

$$\mathrm{C} = (q, p, u) \in Q \times \mathbb{N} \times \Gamma^* \quad \text{mit } p < |u|.$$

Idee: $\mathrm{C} = (q, p, u)$ gibt an, dass die TM sich im Zustand q befindet, der Kopf an Position p steht und die Inschrift des Arbeitsbandes u gefolgt von Blank-Symbolen ist.

(b) $\mathcal{C}_M := \{(q, p, u) \in Q \times \mathbb{N} \times \Gamma^* : p < |u|\}$ bezeichnet die Menge aller möglichen Konfigurationen von M.

(c) Die Startkonfiguration von M bei Eingabe $w \in \Sigma^*$ ist

$$\mathrm{C}_0(w) := (q_0, 0, w\Box).$$

Definition 1.2 (Konfiguration einer TM)

(a) Eine Konfiguration einer TM $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ ist ein Tripel

$$\mathrm{C} = (q, p, u) \in Q \times \mathbb{N} \times \Gamma^* \quad \text{mit } p < |u|.$$

Idee: $\mathrm{C} = (q, p, u)$ gibt an, dass die TM sich im Zustand q befindet, der Kopf an Position p steht und die Inschrift des Arbeitsbandes u gefolgt von Blank-Symbolen ist.

(b) $\mathcal{C}_M := \{(q, p, u) \in Q \times \mathbb{N} \times \Gamma^* : p < |u|\}$ bezeichnet die Menge aller möglichen Konfigurationen von M.

(c) Die Startkonfiguration von M bei Eingabe $w \in \Sigma^*$ ist

$$\mathrm{C}_0(w) := (q_0, 0, w\Box).$$

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

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

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

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

$$\text{Next}_M(\mathrm{C}) =$$

D.h.: $\text{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

$$\mathrm{Next}_M : \mathcal{C}_M \to \mathcal{P}(\mathcal{C}_M),$$

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

$$\mathrm{Next}_M(\mathrm{C}) = \{ (q', p', u') \in \mathcal{C}_M : \text{es gibt } (q, a, q', b, m) \in \Delta \text{ mit:}$$

D.h.: $\mathrm{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

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

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

$$\begin{aligned}\text{Next}_M(\mathrm{C}) = \{\, (q', p', u') \in \mathcal{C}_M :\ & \text{es gibt } (q, a, q', b, m) \in \Delta \text{ mit:} \\ & p' = p + m,\end{aligned}$$

D.h.: $\text{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

$$\mathrm{Next}_M : \mathcal{C}_M \to \mathcal{P}(\mathcal{C}_M),$$

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

$$\begin{aligned}\mathrm{Next}_M(\mathrm{C}) = \{\, (q', p', u') \in \mathcal{C}_M \,:\, & \text{es gibt } (q, a, q', b, m) \in \Delta \text{ mit:} \\ & p' = p + m, \\ & u_p = a, \\ & u'_p = b,\end{aligned}$$

D.h.: $\mathrm{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

$$\mathsf{Next}_M : \mathcal{C}_M \to \mathcal{P}(\mathcal{C}_M),$$

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

$$\begin{aligned}\mathsf{Next}_M(\mathrm{C}) = \{ (q', p', u') \in \mathcal{C}_M \ : \ &\text{es gibt } (q, a, q', b, m) \in \Delta \text{ mit:} \\ &p' = p + m, \\ &u_p = a, \\ &u'_p = b, \\ &|u'| = |u|, \qquad\qquad\qquad\qquad \text{falls } p' < |u|, \\ &|u'| = |u| + 1 \text{ und } u'_{p'} = \Box, \qquad \text{falls } p' = |u|,\end{aligned}$$

D.h.: $\mathsf{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

$$\mathsf{Next}_M : \mathcal{C}_M \to \mathcal{P}(\mathcal{C}_M),$$

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

$$\begin{aligned}
\mathsf{Next}_M(\mathrm{C}) = \{\, (q', p', u') \in \mathcal{C}_M \ :\ &\text{es gibt } (q, a, q', b, m) \in \Delta \text{ mit:} \\
& p' = p + m, \\
& u_p = a, \\
& u'_p = b, \\
& |u'| = |u|, && \text{falls } p' < |u|, \\
& |u'| = |u| + 1 \text{ und } u'_{p'} = \Box, && \text{falls } p' = |u|, \\
& u'_i = u_i \ \text{ f.a. } i < |u| \text{ mit } i \neq p \,\}
\end{aligned}$$

D.h.: $\mathsf{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.

Definition 1.3 (Lauf einer TM)

Sei $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ eine TM.

(a) Die Übergangsrelation Δ induziert eine Funktion

$$\mathsf{Next}_M : \mathcal{C}_M \to \mathcal{P}(\mathcal{C}_M),$$

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

$$\begin{aligned} \mathsf{Next}_M(\mathrm{C}) = \{\, (q', p', u') \in \mathcal{C}_M \ : \ & \text{es gibt } (q, a, q', b, m) \in \Delta \text{ mit:} \\ & p' = p + m, \\ & u_p = a, \\ & u'_p = b, \\ & |u'| = |u|, && \text{falls } p' < |u|, \\ & |u'| = |u| + 1 \text{ und } u'_{p'} = \Box, && \text{falls } p' = |u|, \\ & u'_i = u_i \ \ \text{f.a. } i < |u| \text{ mit } i \neq p \,\} \end{aligned}$$

D.h.: $\mathsf{Next}_M(\mathrm{C})$ enthält alle Konfigurationen, in die M von C aus in genau einem Schritt gelangen kann.
Beachte: Ist M deterministisch, so ist $|\mathsf{Next}_M(\mathrm{C})| \leqslant 1$ f.a. $\mathrm{C} \in \mathcal{C}_M$.

(b) Ein Lauf von M bei Eingabe w ist ein Tupel $\mathcal{L} = \big(\mathrm{C}_0, \mathrm{C}_1, \ldots\big)$ von Konfigurationen von M, so dass gilt:

(b) Ein Lauf von M bei Eingabe w ist ein Tupel $\mathcal{L} = \big(\mathrm{C}_0, \mathrm{C}_1, \ldots\big)$ von Konfigurationen von M, so dass gilt:

- $\mathrm{C}_0 = \mathrm{C}_0(w)$,

(b) Ein Lauf von M bei Eingabe w ist ein Tupel $\mathcal{L} = \big(\mathrm{C}_0, \mathrm{C}_1, \ldots\big)$ von Konfigurationen von M, so dass gilt:

- $\mathrm{C}_0 = \mathrm{C}_0(w)$,
- $\mathrm{C}_i \in \mathsf{Next}_M(\mathrm{C}_{i-1})$ f.a. $i \geqslant 1$, und

(b) Ein Lauf von M bei Eingabe w ist ein Tupel $\mathcal{L} = \big(\mathrm{C}_0, \mathrm{C}_1, \ldots\big)$ von Konfigurationen von M, so dass gilt:

- $\mathrm{C}_0 = \mathrm{C}_0(w)$,
- $\mathrm{C}_i \in \mathsf{Next}_M(\mathrm{C}_{i-1})$ f.a. $i \geqslant 1$, und
- entweder ist das Tupel $\mathcal{L}$ unendlich lang, oder es endet mit einer Endkonfiguration. Im letzteren Fall heißt der Lauf dann endlich bzw. terminierend.

(b) Ein Lauf von M bei Eingabe w ist ein Tupel $\mathcal{L} = \big(\mathrm{C}_0, \mathrm{C}_1, \dots\big)$ von Konfigurationen von M, so dass gilt:

- $\mathrm{C}_0 = \mathrm{C}_0(w)$,
- $\mathrm{C}_i \in \mathsf{Next}_M(\mathrm{C}_{i-1})$ f.a. $i \geqslant 1$, und
- entweder ist das Tupel $\mathcal{L}$ unendlich lang, oder es endet mit einer Endkonfiguration. Im letzteren Fall heißt der Lauf dann endlich bzw. terminierend.

Eine DTM hat folglich für jede Eingabe maximal einen Lauf.

(b) Ein Lauf von M bei Eingabe w ist ein Tupel $\mathcal{L} = \big(\mathrm{C}_0, \mathrm{C}_1, \ldots\big)$ von Konfigurationen von M, so dass gilt:

- $\mathrm{C}_0 = \mathrm{C}_0(w)$,
- $\mathrm{C}_i \in \mathsf{Next}_M(\mathrm{C}_{i-1})$ f.a. $i \geqslant 1$, und
- entweder ist das Tupel $\mathcal{L}$ unendlich lang, oder es endet mit einer Endkonfiguration. Im letzteren Fall heißt der Lauf dann endlich bzw. terminierend.

Eine DTM hat folglich für jede Eingabe maximal einen Lauf.

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

Definition 1.4 (Sprache einer TM)

(a) Eine TM M akzeptiert eine Eingabe $w \in \Sigma^*$, falls es (mindestens) einen akzeptierenden Lauf von M auf w gibt. M verwirft w, falls alle terminierenden Läufe von M auf w verwerfen.

Definition 1.4 (Sprache einer TM)

(a) Eine TM M akzeptiert eine Eingabe $w \in \Sigma^*$, falls es (mindestens) einen akzeptierenden Lauf von M auf w gibt. M verwirft w, falls alle terminierenden Läufe von M auf w verwerfen.

(b) Die von einer TM M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* : M \text{ akzeptiert } w\}.$$

Definition 1.4 (Sprache einer TM)

(a) Eine TM M akzeptiert eine Eingabe $w \in \Sigma^*$, falls es (mindestens) einen akzeptierenden Lauf von M auf w gibt. M verwirft w, falls alle terminierenden Läufe von M auf w verwerfen.

(b) Die von einer TM M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \,:\, M \text{ akzeptiert } w\}.$$

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

Definition 1.4 (Sprache einer TM)

(a) Eine TM M akzeptiert eine Eingabe $w \in \Sigma^*$, falls es (mindestens) einen akzeptierenden Lauf von M auf w gibt. M verwirft w, falls alle terminierenden Läufe von M auf w verwerfen.

(b) Die von einer TM M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* : M \text{ akzeptiert } w\}.$$

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

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

Aus der Veranstaltung „Einführung in die Theoretische Informatik“ (HU Berlin) sind die beiden folgenden Sätze bekannt (hier ohne Beweise):

Aus der Veranstaltung „Einführung in die Theoretische Informatik" (HU Berlin) sind die beiden folgenden Sätze bekannt (hier ohne Beweise):

Satz 1.5

Jede durch eine NTM entscheidbare (bzw. semi-entscheidbare) Sprache ist auch durch eine DTM entscheidbar (bzw. semi-entscheidbar).

Aus der Veranstaltung „Einführung in die Theoretische Informatik“ (HU Berlin) sind die beiden folgenden Sätze bekannt (hier ohne Beweise):

Satz 1.5

Jede durch eine NTM entscheidbare (bzw. semi-entscheidbare) Sprache ist auch durch eine DTM entscheidbar (bzw. semi-entscheidbar).

Satz 1.6

Das Halteproblem auf leerem Eingabewort

H_ε	
Eingabe:	Eine DTM M
Frage:	Hält M bei Eingabe des leeren Worts ε?

ist nicht entscheidbar.

Aus der Veranstaltung „Einführung in die Theoretische Informatik" (HU Berlin) sind die beiden folgenden Sätze bekannt (hier ohne Beweise):

Satz 1.5

Jede durch eine NTM entscheidbare (bzw. semi-entscheidbare) Sprache ist auch durch eine DTM entscheidbar (bzw. semi-entscheidbar).

Satz 1.6

Das Halteproblem auf leerem Eingabewort

H_ε	
Eingabe:	Eine DTM M
Frage:	Hält M bei Eingabe des leeren Worts ε?

ist nicht entscheidbar. Es ist jedoch semi-entscheidbar.

Der Satz von Trakhtenbrot

Boris A. Trakhtenbrot: russisch-israelischer Mathematiker (1921–2016).
Alternative Schreibweisen: Trachtenbrot, Trahenbrot.

Boris A. Trakhtenbrot: russisch-israelischer Mathematiker (1921–2016).
Alternative Schreibweisen: Trachtenbrot, Trahenbrot.

Definition 1.7

Eine funktionenfreie, endliche Signatur (im Folgenden kurz: Signatur) ist eine endliche Menge

$$\sigma = \{R_1, \dots, R_k, c_1, \dots, c_\ell\} \text{ mit } k, \ell \in \mathbb{N}.$$

Dabei stellen $R_1, \dots, R_k$ Relations- und $c_1, \dots, c_\ell$ Konstantensymbole dar.
Jedes R_i hat eine feste Stelligkeit $\mathrm{ar}(R_i) \in \mathbb{N}_{\geqslant 1}$.

Definition 1.8

Sei σ eine (funktionenfreie, endliche) Signatur.
Das endliche Erfüllbarkeitsproblem für FO[σ] ist das Berechnungsproblem mit:

Eingabe:	Ein FO[σ]-Satz φ
Frage:	Gibt es eine endliche σ-Struktur $\mathcal{A}$ mit $\mathcal{A} \models \varphi$?

Definition 1.8

Sei σ eine (funktionenfreie, endliche) Signatur.
Das endliche Erfüllbarkeitsproblem für FO[σ] ist das Berechnungsproblem mit:

Eingabe:	Ein FO[σ]-Satz φ
Frage:	Gibt es eine endliche σ-Struktur $\mathcal{A}$ mit $\mathcal{A} \models \varphi$?

Bemerkung

Man kann sich leicht überlegen, dass das endliche Erfüllbarkeitsproblem für FO[σ] semi-entscheidbar ist.

Definition 1.8

Sei σ eine (funktionenfreie, endliche) Signatur.
Das endliche Erfüllbarkeitsproblem für $\text{FO}[\sigma]$ ist das Berechnungsproblem mit:

Eingabe:	Ein $\text{FO}[\sigma]$-Satz φ
Frage:	Gibt es eine endliche σ-Struktur $\mathcal{A}$ mit $\mathcal{A} \models \varphi$?

Bemerkung

Man kann sich leicht überlegen, dass das endliche Erfüllbarkeitsproblem für $\text{FO}[\sigma]$ semi-entscheidbar ist.

Theorem 1.9 (Satz von Trakhtenbrot, 1950)

Es gibt eine (endliche, funktionenfreie) Signatur σ, so dass das endliche Erfüllbarkeitsproblem für $\text{FO}[\sigma]$ unentscheidbar ist.

Bemerkung 1.10

Man kann sogar zeigen, dass der Satz von Trakhtenbrot für die Signatur $\sigma := \{E\}$ gilt, wobei E ein 2-stelliges Relationssymbol ist.

Bemerkung 1.10

Man kann sogar zeigen, dass der Satz von Trakhtenbrot für die Signatur $\sigma := \{E\}$ gilt, wobei E ein 2-stelliges Relationssymbol ist.
Daraus folgt dann natürlich auch, dass der Satz von Trakhtenbrot für jede Signatur gilt, die mindestens ein Relationssymbol der Stelligkeit $\geqslant 2$ enthält.

Bemerkung 1.10

Man kann sogar zeigen, dass der Satz von Trakhtenbrot für die Signatur $\sigma := \{E\}$ gilt, wobei E ein 2-stelliges Relationssymbol ist.
Daraus folgt dann natürlich auch, dass der Satz von Trakhtenbrot für jede Signatur gilt, die mindestens ein Relationssymbol der Stelligkeit $\geqslant 2$ enthält.

Andererseits werden wir in einem späteren Kapitel unter Verwendung von Lokalitätsresultaten zeigen, dass

Bemerkung 1.10

Man kann sogar zeigen, dass der Satz von Trakhtenbrot für die Signatur $\sigma := \{E\}$ gilt, wobei E ein 2-stelliges Relationssymbol ist.
Daraus folgt dann natürlich auch, dass der Satz von Trakhtenbrot für jede Signatur gilt, die mindestens ein Relationssymbol der Stelligkeit $\geqslant 2$ enthält.

Andererseits werden wir in einem späteren Kapitel unter Verwendung von Lokalitätsresultaten zeigen, dass für jede funktionenfreie Signatur σ, deren Relationssymbole alle die Stelligkeit 1 haben, das endliche Erfüllbarkeitsproblem für FO[σ] entscheidbar ist.

Bemerkung 1.10

Man kann sogar zeigen, dass der Satz von Trakhtenbrot für die Signatur $\sigma := \{E\}$ gilt, wobei E ein 2-stelliges Relationssymbol ist.
Daraus folgt dann natürlich auch, dass der Satz von Trakhtenbrot für jede Signatur gilt, die mindestens ein Relationssymbol der Stelligkeit $\geqslant 2$ enthält.

Andererseits werden wir in einem späteren Kapitel unter Verwendung von Lokalitätsresultaten zeigen, dass für jede funktionenfreie Signatur σ, deren Relationssymbole alle die Stelligkeit 1 haben, das endliche Erfüllbarkeitsproblem für FO[σ] entscheidbar ist.

Bemerkung 1.11

Alternativ zum Halteproblem H_ε kann man auf Grund des Satzes von Trakhtenbrot auch das endliche Erfüllbarkeitsproblem für FO[σ] als Grundlage für Reduktionen nutzen, mit denen man die Unentscheidbarkeit bestimmter Probleme nachweist. Beispiele dazu werden in den Übungen betrachtet.

Weitere Lektüre

- Lehrbuch [**L**]: Kapitel 2.1, 2.2 und 9.1.

Kapitel 2:

Logik zweiter Stufe und die Sätze von Büchi und Fagin

Abschnitt 2.1:

Syntax und Semantik der Logik zweiter Stufe

Idee: In der Logik zweiter Stufe gibt es für jede Stelligkeit $k \geqslant 1$ abzählbar viele k-stellige Relationsvariablen $\mathsf{Var}_1^k, \mathsf{Var}_2^k, \ldots$, die wir meistens mit Großbuchstaben wie $X, Y, \ldots$ bezeichnen werden.

Idee: In der Logik zweiter Stufe gibt es für jede Stelligkeit $k \geqslant 1$ abzählbar viele k-stellige Relationsvariablen $\mathsf{Var}_1^k, \mathsf{Var}_2^k, \ldots$, die wir meistens mit Großbuchstaben wie $X, Y, \ldots$ bezeichnen werden.

Definition 2.1 (SO: Syntax)

Sei σ eine (endliche, funktionenfreie) Signatur.

Idee: In der Logik zweiter Stufe gibt es für jede Stelligkeit $k \geqslant 1$ abzählbar viele k-stellige Relationsvariablen $\mathsf{Var}_1^k, \mathsf{Var}_2^k, \ldots$, die wir meistens mit Großbuchstaben wie $X, Y, \ldots$ bezeichnen werden.

Definition 2.1 (SO: Syntax)

Sei σ eine (endliche, funktionenfreie) Signatur.

(a) $\mathsf{Var}_2 := \{\mathsf{Var}_i^k \,;\, i, k \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge aller Variablen zweiter Stufe (oder: Relationsvariablen). Die Relationsvariable Var_i^k hat die Stelligkeit $\mathrm{ar}(\mathsf{Var}_i^k) = k$.

Idee: In der Logik zweiter Stufe gibt es für jede Stelligkeit $k \geqslant 1$ abzählbar viele k-stellige Relationsvariablen $\mathsf{Var}_1^k, \mathsf{Var}_2^k, \ldots$, die wir meistens mit Großbuchstaben wie $X, Y, \ldots$ bezeichnen werden.

Definition 2.1 (SO: Syntax)

Sei σ eine (endliche, funktionenfreie) Signatur.

(a) $\mathsf{Var}_2 := \{\mathsf{Var}_i^k \,;\, i, k \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge aller Variablen zweiter Stufe (oder: Relationsvariablen). Die Relationsvariable Var_i^k hat die Stelligkeit $\mathrm{ar}(\mathsf{Var}_i^k) = k$.

$\mathsf{Var}_1 := \mathsf{Var} = \{\mathsf{var}_i \,;\, i \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge aller Variablen erster Stufe (oder: Individuenvariablen).

Idee: In der Logik zweiter Stufe gibt es für jede Stelligkeit $k \geqslant 1$ abzählbar viele k-stellige Relationsvariablen $\mathsf{Var}_1^k, \mathsf{Var}_2^k, \ldots$, die wir meistens mit Großbuchstaben wie $X, Y, \ldots$ bezeichnen werden.

Definition 2.1 (SO: Syntax)

Sei σ eine (endliche, funktionenfreie) Signatur.

(a) $\mathsf{Var}_2 := \{\mathsf{Var}_i^k \,;\, i, k \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge aller Variablen zweiter Stufe (oder: Relationsvariablen). Die Relationsvariable Var_i^k hat die Stelligkeit $\mathrm{ar}(\mathsf{Var}_i^k) = k$.

$\mathsf{Var}_1 := \mathsf{Var} = \{\mathsf{var}_i \,;\, i \in \mathbb{N}_{\geqslant 1}\}$ ist die Menge aller Variablen erster Stufe (oder: Individuenvariablen).

Ein σ-Term ist ein Element der Menge $\mathsf{T}_\sigma := \mathsf{Var}_1 \cup \{c \in \sigma : c \text{ ist ein Konstantensymbol}\}$.

(b) Die Formelmenge SO[σ] ist rekursiv wie folgt definiert:

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\mathrm{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\mathrm{SO}[\sigma]$:

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\mathrm{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\mathrm{SO}[\sigma]$:

- $\neg\psi$
- $(\varphi_1 \vee \varphi_2)$
- $(\varphi_1 \wedge \varphi_2)$
- $(\varphi_1 \rightarrow \varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2)$

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\mathrm{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\mathrm{SO}[\sigma]$:

- $\neg\psi$
- $(\varphi_1 \vee \varphi_2)$
- $(\varphi_1 \wedge \varphi_2)$
- $(\varphi_1 \to \varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2)$

(Q1) Ist ψ eine Formel in $\mathrm{SO}[\sigma]$ und $x \in \mathrm{Var}_1$, so gehören auch folgende Formeln zu $\mathrm{SO}[\sigma]$:

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1,\dots,t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1,\dots,t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1,\dots,t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1,\dots,t_k \in \mathrm{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\mathrm{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\mathrm{SO}[\sigma]$:
- $\neg\psi$
- $(\varphi_1 \vee \varphi_2)$
- $(\varphi_1 \wedge \varphi_2)$
- $(\varphi_1 \to \varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2)$

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

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\mathrm{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\mathrm{SO}[\sigma]$:
- $\neg\psi$
- $(\varphi_1 \vee \varphi_2)$
- $(\varphi_1 \wedge \varphi_2)$
- $(\varphi_1 \rightarrow \varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2)$

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

(Q2) Ist ψ eine Formel in $\mathrm{SO}[\sigma]$ und $X \in \mathrm{Var}_2$, so gehören auch folgende Formeln zu $\mathrm{SO}[\sigma]$:

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

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\text{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \text{ar}(R)$, $t_1, \ldots, t_k \in \text{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\text{SO}[\sigma]$
f.a. $t, t' \in \text{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\text{SO}[\sigma]$
f.a. $X \in \text{Var}_2$, $k := \text{ar}(X)$, $t_1, \ldots, t_k \in \text{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\text{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\text{SO}[\sigma]$:
- $\neg\psi$
- $(\varphi_1 \vee \varphi_2)$
- $(\varphi_1 \wedge \varphi_2)$
- $(\varphi_1 \rightarrow \varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2)$

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

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

(b) Die Formelmenge $\mathrm{SO}[\sigma]$ ist rekursiv wie folgt definiert:

(A1) $R(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. Relationssymbole $R \in \sigma$, $k := \mathrm{ar}(R)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(A2) $t{=}t'$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $t, t' \in \mathrm{T}_\sigma$

(A3) $X(t_1, \ldots, t_k)$ gehört zu $\mathrm{SO}[\sigma]$
f.a. $X \in \mathrm{Var}_2$, $k := \mathrm{ar}(X)$, $t_1, \ldots, t_k \in \mathrm{T}_\sigma$

(BC) Sind ψ, φ_1, φ_2 Formeln in $\mathrm{SO}[\sigma]$, so gehören auch die folgenden Formeln zu $\mathrm{SO}[\sigma]$:
- $\neg\psi$
- $(\varphi_1 \vee \varphi_2)$
- $(\varphi_1 \wedge \varphi_2)$
- $(\varphi_1 \rightarrow \varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2)$

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

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

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

Definition 2.2

(a) Für $\varphi \in \mathsf{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Definition 2.2

(a) Für $\varphi \in \mathsf{SO}[\sigma]$ bezeichnet frei(φ) die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und qr(φ) den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathsf{Var}_1 \cup \mathsf{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathsf{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

Definition 2.2

(a) Für $\varphi \in \mathrm{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathrm{Var}_1 \cup \mathrm{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathrm{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

$$\mathrm{frei}(R(t_1, \ldots, t_k)) \quad = \quad \{t_1, \ldots, t_k\} \cap \mathrm{Var}_1,$$

Definition 2.2

(a) Für $\varphi \in \mathrm{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathrm{Var}_1 \cup \mathrm{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathrm{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

$$\begin{aligned} \mathrm{frei}(R(t_1, \ldots, t_k)) &= \{t_1, \ldots, t_k\} \cap \mathrm{Var}_1, \\ \mathrm{frei}(t{=}t') &= \{t, t'\} \cap \mathrm{Var}_1, \end{aligned}$$

Definition 2.2

(a) Für $\varphi \in \mathrm{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathrm{Var}_1 \cup \mathrm{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathrm{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

$$\begin{aligned}
\mathrm{frei}(R(t_1, \ldots, t_k)) &= \{t_1, \ldots, t_k\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(t{=}t') &= \{t, t'\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(X(t_1, \ldots, t_k)) &= (\{t_1, \ldots, t_k\} \cap \mathrm{Var}_1) \cup \{X\},
\end{aligned}$$

Definition 2.2

(a) Für $\varphi \in \mathrm{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathrm{Var}_1 \cup \mathrm{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathrm{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

$$\begin{array}{rcl}
\mathrm{frei}(R(t_1, \ldots, t_k)) & = & \{t_1, \ldots, t_k\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(t{=}t') & = & \{t, t'\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(X(t_1, \ldots, t_k)) & = & (\{t_1, \ldots, t_k\} \cap \mathrm{Var}_1) \cup \{X\}, \\
\mathrm{frei}(\neg\psi) & = & \mathrm{frei}(\psi),
\end{array}$$

Definition 2.2

(a) Für $\varphi \in \mathrm{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathrm{Var}_1 \cup \mathrm{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathrm{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

$$\begin{aligned}
\mathrm{frei}(R(t_1,\ldots,t_k)) &= \{t_1,\ldots,t_k\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(t{=}t') &= \{t,t'\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(X(t_1,\ldots,t_k)) &= (\{t_1,\ldots,t_k\} \cap \mathrm{Var}_1) \cup \{X\}, \\
\mathrm{frei}(\neg\psi) &= \mathrm{frei}(\psi), \\
\mathrm{frei}((\varphi_1 * \varphi_2)) &= \mathrm{frei}(\varphi_1) \cup \mathrm{frei}(\varphi_2),
\end{aligned}$$

Definition 2.2

(a) Für $\varphi \in \mathrm{SO}[\sigma]$ bezeichnet $\mathrm{frei}(\varphi)$ die Menge aller Individuen- und Relationsvariablen, die frei in φ vorkommen, und $\mathrm{qr}(\varphi)$ den Quantorenrang (bzw. die Quantorentiefe) von φ, d.h. die maximale Anzahl ineinander geschachtelter Quantoren in φ.

Das heißt, f.a. $* \in \{\vee, \wedge, \to, \leftrightarrow\}$, $Q \in \{\exists, \forall\}$, $v \in \mathrm{Var}_1 \cup \mathrm{Var}_2$, σ-Terme $t, t', t_1, \ldots, t_k$, Relationssymbole $R \in \sigma$, Variablen $X \in \mathrm{Var}_2$ und Formeln $\psi, \varphi_1, \varphi_2$ gilt

$$
\begin{aligned}
\mathrm{frei}(R(t_1, \ldots, t_k)) &= \{t_1, \ldots, t_k\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(t{=}t') &= \{t, t'\} \cap \mathrm{Var}_1, \\
\mathrm{frei}(X(t_1, \ldots, t_k)) &= (\{t_1, \ldots, t_k\} \cap \mathrm{Var}_1) \cup \{X\}, \\
\mathrm{frei}(\neg\psi) &= \mathrm{frei}(\psi), \\
\mathrm{frei}((\varphi_1 * \varphi_2)) &= \mathrm{frei}(\varphi_1) \cup \mathrm{frei}(\varphi_2), \\
\mathrm{frei}(Qv\,\psi) &= \mathrm{frei}(\psi) \setminus \{v\}
\end{aligned}
$$

und

$$\begin{array}{rcl} \mathrm{qr}(\varphi) & = & 0 \quad \text{für atomare } \sigma\text{-Formeln } \varphi, \\ \mathrm{qr}(\neg\psi) & = & \mathrm{qr}(\psi), \\ \mathrm{qr}((\varphi_1 * \varphi_2)) & = & \max\{\mathrm{qr}(\varphi_1), \mathrm{qr}(\varphi_2)\}, \\ \mathrm{qr}(Qv\,\psi) & = & \mathrm{qr}(\psi) + 1. \end{array}$$

und

$$\begin{array}{rcl} \mathrm{qr}(\varphi) & = & 0 \quad \text{für atomare } \sigma\text{-Formeln } \varphi, \\ \mathrm{qr}(\neg\psi) & = & \mathrm{qr}(\psi), \\ \mathrm{qr}((\varphi_1 * \varphi_2)) & = & \max\{\mathrm{qr}(\varphi_1), \mathrm{qr}(\varphi_2)\}, \\ \mathrm{qr}(Qv\,\psi) & = & \mathrm{qr}(\psi) + 1. \end{array}$$

Wir schreiben oft $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ um anzudeuten, dass $\mathrm{frei}(\varphi) \subseteq \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ ist.

und

$$\begin{array}{rcl} \mathrm{qr}(\varphi) & = & 0 \quad \text{für atomare } \sigma\text{-Formeln } \varphi, \\ \mathrm{qr}(\neg\psi) & = & \mathrm{qr}(\psi), \\ \mathrm{qr}((\varphi_1 * \varphi_2)) & = & \max\{\mathrm{qr}(\varphi_1), \mathrm{qr}(\varphi_2)\}, \\ \mathrm{qr}(Qv\,\psi) & = & \mathrm{qr}(\psi) + 1. \end{array}$$

Wir schreiben oft $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ um anzudeuten, dass $\mathrm{frei}(\varphi) \subseteq \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ ist.

(b) Eine $\mathrm{SO}[\sigma]$-Formel φ heißt Satz, falls $\mathrm{frei}(\varphi) = \emptyset$ ist.

und

$$\begin{array}{rcl} \mathrm{qr}(\varphi) & = & 0 \quad \text{für atomare } \sigma\text{-Formeln } \varphi, \\ \mathrm{qr}(\neg\psi) & = & \mathrm{qr}(\psi), \\ \mathrm{qr}((\varphi_1 * \varphi_2)) & = & \max\{\mathrm{qr}(\varphi_1), \mathrm{qr}(\varphi_2)\}, \\ \mathrm{qr}(Qv\,\psi) & = & \mathrm{qr}(\psi) + 1. \end{array}$$

Wir schreiben oft $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ um anzudeuten, dass $\mathrm{frei}(\varphi) \subseteq \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ ist.

(b) Eine SO[σ]-Formel φ heißt Satz, falls $\mathrm{frei}(\varphi) = \emptyset$ ist.

Beispiel

Für die SO[σ]-Formel

$$\varphi := \exists X\, \forall Y\, \Big(\exists z\, \exists Z\, Z(z) \vee \forall u\, \big(Y(u) \vee u{=}z\big)\Big)$$

gilt

und

$$\begin{array}{rcl} \mathrm{qr}(\varphi) & = & 0 \quad \text{für atomare } \sigma\text{-Formeln } \varphi, \\ \mathrm{qr}(\neg\psi) & = & \mathrm{qr}(\psi), \\ \mathrm{qr}((\varphi_1 * \varphi_2)) & = & \max\{\mathrm{qr}(\varphi_1), \mathrm{qr}(\varphi_2)\}, \\ \mathrm{qr}(Qv\,\psi) & = & \mathrm{qr}(\psi) + 1. \end{array}$$

Wir schreiben oft $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ um anzudeuten, dass $\mathrm{frei}(\varphi) \subseteq \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ ist.

(b) Eine SO[σ]-Formel φ heißt Satz, falls $\mathrm{frei}(\varphi) = \emptyset$ ist.

Beispiel

Für die SO[σ]-Formel

$$\varphi := \exists X\, \forall Y \Big(\exists z\, \exists Z\, Z(z) \vee \forall u\, \big(Y(u) \vee u{=}z\big)\Big)$$

gilt $\mathrm{frei}(\varphi) = \{z\}$ und $\mathrm{qr}(\varphi) = 4$.

Definition 2.3 (SO: Semantik)

Die Semantik von SO[σ] ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von FO[σ]. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathrm{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathrm{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur

Definition 2.3 (SO: Semantik)

Die Semantik von SO[σ] ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von FO[σ]. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine SO[σ]-Formel.

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathrm{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathrm{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathrm{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A,

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathsf{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathsf{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathsf{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathsf{Var}_1$ und $\beta(X) \subseteq A^{\mathsf{ar}(X)}$ für alle $X \in M \cap \mathsf{Var}_2$.

Definition 2.3 (SO: Semantik)

Die Semantik von SO[σ] ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von FO[σ]. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine SO[σ]-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathrm{Var}_1$ und $\beta(X) \subseteq A^{\mathrm{ar}(X)}$ für alle $X \in M \cap \mathrm{Var}_2$.

Sei $\mathcal{I} := (\mathcal{A}, \beta)$ die zugehörige SO[σ]-Interpretation.

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathsf{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathsf{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathsf{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathrm{Var}_1$ und $\beta(X) \subseteq A^{\mathrm{ar}(X)}$ für alle $X \in M \cap \mathrm{Var}_2$.

Sei $\mathcal{I} := (\mathcal{A}, \beta)$ die zugehörige $\mathsf{SO}[\sigma]$-Interpretation.

Wir definieren $v^{\mathcal{I}} := \beta(v)$ für $v \in M$,

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathrm{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathrm{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathrm{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathrm{Var}_1$ und $\beta(X) \subseteq A^{\mathrm{ar}(X)}$ für alle $X \in M \cap \mathrm{Var}_2$.

Sei $\mathcal{I} := (\mathcal{A}, \beta)$ die zugehörige $\mathrm{SO}[\sigma]$-Interpretation.

Wir definieren $v^{\mathcal{I}} := \beta(v)$ für $v \in M$, und $c^{\mathcal{I}} := c^{\mathcal{A}}$ für Konstantensymbole $c \in \sigma$.

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathrm{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathrm{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathrm{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathrm{Var}_1$ und $\beta(X) \subseteq A^{\mathrm{ar}(X)}$ für alle $X \in M \cap \mathrm{Var}_2$.

Sei $\mathcal{I} := (\mathcal{A}, \beta)$ die zugehörige $\mathrm{SO}[\sigma]$-Interpretation.

Wir definieren $v^{\mathcal{I}} := \beta(v)$ für $v \in M$, und $c^{\mathcal{I}} := c^{\mathcal{A}}$ für Konstantensymbole $c \in \sigma$.

Für $X \in \mathrm{Var}_2$ und $S \subseteq A^{\mathrm{ar}(X)}$ bezeichne $\beta\frac{S}{X}$ die Belegung von $M \cup \{X\}$ in A mit

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathsf{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathsf{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathsf{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathsf{Var}_1$ und $\beta(X) \subseteq A^{\mathrm{ar}(X)}$ für alle $X \in M \cap \mathsf{Var}_2$.

Sei $\mathcal{I} := (\mathcal{A}, \beta)$ die zugehörige $\mathsf{SO}[\sigma]$-Interpretation.

Wir definieren $v^{\mathcal{I}} := \beta(v)$ für $v \in M$, und $c^{\mathcal{I}} := c^{\mathcal{A}}$ für Konstantensymbole $c \in \sigma$.

Für $X \in \mathsf{Var}_2$ und $S \subseteq A^{\mathrm{ar}(X)}$ bezeichne $\beta\frac{S}{X}$ die Belegung von $M \cup \{X\}$ in A mit $\beta\frac{S}{X}(X) = S$ und $\beta\frac{S}{X}(v) = \beta(v)$ für alle $v \in M \setminus \{X\}$.

Definition 2.3 (SO: Semantik)

Die Semantik von $\mathrm{SO}[\sigma]$ ist auf die offensichtliche Weise definiert, als Erweiterung der Semantik von $\mathrm{FO}[\sigma]$. Deshalb geben wir im Folgenden nur die zusätzlichen Fälle an.

Sei dazu $\mathcal{A}$ eine σ-Struktur und $\varphi(X_1, \ldots, X_t, x_1, \ldots, x_s)$ eine $\mathrm{SO}[\sigma]$-Formel.

Weiter sei β eine Belegung von $M := \{X_1, \ldots, X_t, x_1, \ldots, x_s\}$ in A, d.h. eine Zuordnung von M mit $\beta(x) \in A$ für alle $x \in M \cap \mathrm{Var}_1$ und $\beta(X) \subseteq A^{\mathrm{ar}(X)}$ für alle $X \in M \cap \mathrm{Var}_2$.

Sei $\mathcal{I} := (\mathcal{A}, \beta)$ die zugehörige $\mathrm{SO}[\sigma]$-Interpretation.

Wir definieren $v^{\mathcal{I}} := \beta(v)$ für $v \in M$, und $c^{\mathcal{I}} := c^{\mathcal{A}}$ für Konstantensymbole $c \in \sigma$.

Für $X \in \mathrm{Var}_2$ und $S \subseteq A^{\mathrm{ar}(X)}$ bezeichne $\beta\frac{S}{X}$ die Belegung von $M \cup \{X\}$ in A mit $\beta\frac{S}{X}(X) = S$ und $\beta\frac{S}{X}(v) = \beta(v)$ für alle $v \in M \setminus \{X\}$.

Wir definieren außerdem $\mathcal{I}\frac{S}{X} := \big(\mathcal{A}, \beta\frac{S}{X}\big)$.

(A3) Ist φ von der Form $X(t_1, \ldots, t_k)$, so gilt

(A3) Ist φ von der Form $X(t_1, \ldots, t_k)$, so gilt

$$\mathcal{I} \models X(t_1, \ldots, t_k) \iff$$

(A3) Ist φ von der Form $X(t_1, \dots, t_k)$, so gilt

$$\mathcal{I} \models X(t_1, \dots, t_k) \iff (t_1^{\mathcal{I}}, \dots, t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(A3) Ist φ von der Form $X(t_1, \ldots, t_k)$, so gilt

$$\mathcal{I} \models X(t_1, \ldots, t_k) \iff (t_1^{\mathcal{I}}, \ldots, t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(Q2) Ist φ von der Form $\exists X\,\psi$ bzw. $\forall X\,\psi$, so gilt

(A3) Ist φ von der Form $X(t_1, \ldots, t_k)$, so gilt

$$\mathcal{I} \models X(t_1, \ldots, t_k) \iff (t_1^{\mathcal{I}}, \ldots, t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(Q2) Ist φ von der Form $\exists X\, \psi$ bzw. $\forall X\, \psi$, so gilt

$$\mathcal{I} \models \exists X\, \psi \iff$$

(A3) Ist φ von der Form $X(t_1, \ldots, t_k)$, so gilt

$$\mathcal{I} \models X(t_1, \ldots, t_k) \iff (t_1^{\mathcal{I}}, \ldots, t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(Q2) Ist φ von der Form $\exists X\, \psi$ bzw. $\forall X\, \psi$, so gilt

$$\mathcal{I} \models \exists X\, \psi \iff \mathcal{I}\tfrac{S}{X} \models \psi \text{ für (mind.) eine Menge } S \subseteq A^{\mathrm{ar}(X)}$$

(A3) Ist φ von der Form $X(t_1,\dots,t_k)$, so gilt

$$\mathcal{I} \models X(t_1,\dots,t_k) \iff (t_1^{\mathcal{I}},\dots,t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(Q2) Ist φ von der Form $\exists X\,\psi$ bzw. $\forall X\,\psi$, so gilt

$$\mathcal{I} \models \exists X\,\psi \iff \mathcal{I}\tfrac{S}{X} \models \psi \text{ für (mind.) eine Menge } S \subseteq A^{\mathrm{ar}(X)}$$

bzw.

$$\mathcal{I} \models \forall X\,\psi \iff$$

(A3) Ist φ von der Form $X(t_1,\ldots,t_k)$, so gilt

$$\mathcal{I} \models X(t_1,\ldots,t_k) \iff (t_1^{\mathcal{I}},\ldots,t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(Q2) Ist φ von der Form $\exists X\,\psi$ bzw. $\forall X\,\psi$, so gilt

$$\mathcal{I} \models \exists X\,\psi \iff \mathcal{I}\tfrac{S}{X} \models \psi \text{ für (mind.) eine Menge } S \subseteq A^{\mathrm{ar}(X)}$$

bzw.

$$\mathcal{I} \models \forall X\,\psi \iff \mathcal{I}\tfrac{S}{X} \models \psi \text{ für jede Menge } S \subseteq A^{\mathrm{ar}(X)}.$$

(A3) Ist φ von der Form $X(t_1,\ldots,t_k)$, so gilt

$$\mathcal{I} \models X(t_1,\ldots,t_k) \iff (t_1^{\mathcal{I}},\ldots,t_k^{\mathcal{I}}) \in X^{\mathcal{I}}.$$

(Q2) Ist φ von der Form $\exists X\,\psi$ bzw. $\forall X\,\psi$, so gilt

$$\mathcal{I} \models \exists X\,\psi \iff \mathcal{I}\tfrac{S}{X} \models \psi \text{ für (mind.) eine Menge } S \subseteq A^{\mathrm{ar}(X)}$$

bzw.

$$\mathcal{I} \models \forall X\,\psi \iff \mathcal{I}\tfrac{S}{X} \models \psi \text{ für jede Menge } S \subseteq A^{\mathrm{ar}(X)}.$$

Allgemein schreiben wir statt $\mathcal{I} \models \varphi$ auch $(\mathcal{A}, X_1^{\mathcal{I}},\ldots,X_t^{\mathcal{I}}, x_1^{\mathcal{I}},\ldots,x_s^{\mathcal{I}}) \models \varphi$.

Beispiele 2.4

Sei $\sigma := \{E\}$ die Signatur für Graphen, das heißt E ist ein 2-stelliges Relationssymbol.

Beispiele 2.4

Sei $\sigma := \{E\}$ die Signatur für Graphen, das heißt E ist ein 2-stelliges Relationssymbol.

(a) Ein SO[σ]-Satz, der genau dann für einen Graphen gilt, wenn er 3-färbbar ist:

Beispiele 2.4

Sei $\sigma := \{E\}$ die Signatur für Graphen, das heißt E ist ein 2-stelliges Relationssymbol.

(a) Ein SO[σ]-Satz, der genau dann für einen Graphen gilt, wenn er 3-färbbar ist:

$$\begin{aligned}\Phi_{\text{3-col}} \;:=\; \exists R\, \exists G\, \exists B \Bigg(& \forall x \Big(R(x) \lor G(x) \lor B(x) \Big) \land \\ & \forall x\, \forall y \Big(E(x,y) \to \\ & \qquad \neg\Big((R(x) \land R(y)) \lor (G(x) \land G(y)) \lor \\ & \qquad\qquad (B(x) \land B(y)) \Big)\Big)\Bigg)\end{aligned}$$

(b) Ein SO[σ]-Satz, der besagt, dass der Graph symmetrisch und <u>nicht</u> zusammenhängend ist:

(b) Ein SO[σ]-Satz, der besagt, dass der Graph symmetrisch und <u>nicht</u> zusammenhängend ist:

$$\begin{aligned}\Phi_{\text{non-conn}} \ := \ \exists X \Big(& \exists x\, X(x) \land \exists y\, \neg X(y) \land \\ & \neg \exists u\, \exists v\, \big(E(u,v) \land X(u) \land \neg X(v) \big) \\ & \land \forall u\, \forall v\, \Big(E(u,v) \leftrightarrow E(v,u) \Big) \Big)\end{aligned}$$

(c) Eine SO[σ]-Formel $\Phi_{\text{reach}}(x, y)$, die besagt, dass es im Graph einen Weg von Knoten x zu Knoten y gibt:

(c) Eine SO[σ]-Formel $\Phi_{\text{reach}}(x, y)$, die besagt, dass es im Graph einen Weg von Knoten x zu Knoten y gibt:

$$\Phi_{\text{reach}}(x, y) \;:=\; \forall X \left(\left(X(x) \land \forall z \forall z' \left(\left(E(z, z') \land X(z) \right) \to X(z') \right) \right) \to X(y) \right)$$

(d) Ein SO[σ]-Satz, der in einem endlichen Graphen $G = (V, E)$ mit $n := |V|$ besagt, dass er einen Hamiltonkreis enthält,

(d) Ein SO[σ]-Satz, der in einem endlichen Graphen $G = (V, E)$ mit $n := |V|$ besagt, dass er einen Hamiltonkreis enthält, d.h. es gibt paarweise verschiedene Knoten $v_1, \ldots, v_n$ in V, s.d. $(v_i, v_{i+1}) \in E$ f.a. $1 \leqslant i < n$, und $(v_n, v_1) \in E$:

(d) Ein SO[σ]-Satz, der in einem endlichen Graphen $G = (V, E)$ mit $n := |V|$ besagt, dass er einen Hamiltonkreis enthält, d.h. es gibt paarweise verschiedene Knoten $v_1, \ldots, v_n$ in V, s.d. $(v_i, v_{i+1}) \in E$ f.a. $1 \leqslant i < n$, und $(v_n, v_1) \in E$:

$$
\begin{aligned}
\Phi_{\text{Ham}} \ := \ \exists R_< \, \exists R_{succ} \, \exists z_0 \, \exists z_{max} \Big(& E(z_{max}, z_0) \\
& \wedge \ \forall x \forall y \, \big(R_{succ}(x, y) \to E(x, y) \big) \\
& \underbrace{\wedge \ \varphi_{<,succ,0}(R_<, R_{succ}, z_0)}_{(*)} \\
& \underbrace{\wedge \ \forall x \, \big(R_<(x, z_{max}) \vee x = z_{max} \big)}_{z_{max} \text{ ist das größte Element bzgl. } R_<} \Big)
\end{aligned}
$$

$(*)$ Dies ist die Formel aus dem Beweis des Satzes von Trakhtenbrot, in der jedes Vorkommen von $<$ bzw. *succ* bzw. 0 durch $R_<$ bzw. R_{succ} bzw. z_0 ersetzt wurde. Sie besagt also, dass $R_<$ eine strikte lineare Ordnung auf V mit kleinstem Element z_0 und Nachfolger-Relation R_{succ} ist.

Definition 2.5

(a) Die monadische Logik zweiter Stufe MSO[σ] ist die Klasse aller SO[σ]-Formeln φ, s.d. alle in φ vorkommenden Relationsvariablen die Stelligkeit 1 besitzen (solche Relationsvariablen werden auch Mengenvariablen genannt).

Definition 2.5

(a) Die monadische Logik zweiter Stufe $\mathrm{MSO}[\sigma]$ ist die Klasse aller $\mathrm{SO}[\sigma]$-Formeln φ, s.d. alle in φ vorkommenden Relationsvariablen die Stelligkeit 1 besitzen (solche Relationsvariablen werden auch Mengenvariablen genannt).

(b) Die existenzielle Logik zweiter Stufe $\mathrm{ESO}[\sigma]$ (auch: $\exists\mathrm{SO}[\sigma]$, $\Sigma_1^1[\sigma]$) ist die Klasse aller $\mathrm{SO}[\sigma]$-Formeln der Form $\exists X_1 \ldots \exists X_d\, \varphi$, wobei gilt: $d \in \mathbb{N}$, $X_1, \ldots, X_d$ sind Relationsvariablen, und $\varphi \in \mathrm{FO}[\sigma \cup \mathrm{Var}_2]$. Letzteres bedeutet, dass φ ohne (Q2) gebildet wird.

Definition 2.5

(a) Die monadische Logik zweiter Stufe $\mathrm{MSO}[\sigma]$ ist die Klasse aller $\mathrm{SO}[\sigma]$-Formeln φ, s.d. alle in φ vorkommenden Relationsvariablen die Stelligkeit 1 besitzen (solche Relationsvariablen werden auch Mengenvariablen genannt).

(b) Die existenzielle Logik zweiter Stufe $\mathrm{ESO}[\sigma]$ (auch: $\exists\mathrm{SO}[\sigma]$, $\Sigma^1_1[\sigma]$) ist die Klasse aller $\mathrm{SO}[\sigma]$-Formeln der Form $\exists X_1 \ldots \exists X_d\, \varphi$, wobei gilt: $d \in \mathbb{N}$, $X_1, \ldots, X_d$ sind Relationsvariablen, und $\varphi \in \mathrm{FO}[\sigma \cup \mathrm{Var}_2]$. Letzteres bedeutet, dass φ ohne (Q2) gebildet wird.

(c) Die monadische existenzielle Logik zweiter Stufe $\mathrm{EMSO}[\sigma]$ (auch: $\exists\mathrm{MSO}[\sigma]$, $\mathrm{mon}\Sigma^1_1[\sigma]$, $\mathrm{monNP}[\sigma]$) ist die Klasse aller $\mathrm{MSO}[\sigma]$-Formeln, die zugleich $\mathrm{ESO}[\sigma]$-Formeln sind, d.h. aller $\mathrm{SO}[\sigma]$-Formeln der Form $\exists X_1 \ldots \exists X_d\, \varphi$, wobei gilt: $d \in \mathbb{N}$, $X_1, \ldots, X_d$ sind Mengenvariablen, und $\varphi \in \mathrm{FO}[\sigma \cup \{X \in \mathrm{Var}_2 \mid \mathrm{ar}(X) = 1\}]$. Letzteres bedeutet, dass φ ohne (Q2) gebildet wird, und bei Anwendungung von (A3) nur Mengenvariablen genutzt werden.

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu MSO[σ], ESO[σ] und EMSO[σ].

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu $\mathrm{MSO}[\sigma]$, $\mathrm{ESO}[\sigma]$ und $\mathrm{EMSO}[\sigma]$.
- $\Phi_{\text{non-conn}}$ gehört zu

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu $\mathsf{MSO}[\sigma]$, $\mathsf{ESO}[\sigma]$ und $\mathsf{EMSO}[\sigma]$.
- $\Phi_{\text{non-conn}}$ gehört zu $\mathsf{MSO}[\sigma]$, $\mathsf{ESO}[\sigma]$ und $\mathsf{EMSO}[\sigma]$.

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu MSO[σ], ESO[σ] und EMSO[σ].
- $\Phi_{\text{non-conn}}$ gehört zu MSO[σ], ESO[σ] und EMSO[σ].
- Φ_{reach} gehört zu

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu $\text{MSO}[\sigma]$, $\text{ESO}[\sigma]$ und $\text{EMSO}[\sigma]$.
- $\Phi_{\text{non-conn}}$ gehört zu $\text{MSO}[\sigma]$, $\text{ESO}[\sigma]$ und $\text{EMSO}[\sigma]$.
- Φ_{reach} gehört zu $\text{MSO}[\sigma]$, aber nicht zu $\text{ESO}[\sigma]$ und nicht zu $\text{EMSO}[\sigma]$.

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu $\text{MSO}[\sigma]$, $\text{ESO}[\sigma]$ und $\text{EMSO}[\sigma]$.
- $\Phi_{\text{non-conn}}$ gehört zu $\text{MSO}[\sigma]$, $\text{ESO}[\sigma]$ und $\text{EMSO}[\sigma]$.
- Φ_{reach} gehört zu $\text{MSO}[\sigma]$, aber nicht zu $\text{ESO}[\sigma]$ und nicht zu $\text{EMSO}[\sigma]$.
- Φ_{Ham} gehört zu

Beispiele 2.6

Betrachte die Formeln aus Beispiel 2.4:

- $\Phi_{\text{3-col}}$ gehört zu MSO[σ], ESO[σ] und EMSO[σ].
- $\Phi_{\text{non-conn}}$ gehört zu MSO[σ], ESO[σ] und EMSO[σ].
- Φ_{reach} gehört zu MSO[σ], aber nicht zu ESO[σ] und nicht zu EMSO[σ].
- Φ_{Ham} gehört zu ESO[σ], aber nicht zu MSO[σ] und nicht zu EMSO[σ].

Abschnitt 2.2:

MSO und der Satz von Büchi

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \dots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:
- einem 2-stelligen Relationssymbol $\leqslant$

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:
- einem 2-stelligen Relationssymbol $\leqslant$
- einem 1-stelligen Relationssymbol P_a für jeden Buchstaben $a \in \Sigma$.

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:

- einem 2-stelligen Relationssymbol $\leqslant$
- einem 1-stelligen Relationssymbol P_a für jeden Buchstaben $a \in \Sigma$.

(b) Einem endlichen Wort $w_1 \cdots w_n \in \Sigma^+$ der Länge $n \geqslant 1$ (mit $w_i \in \Sigma$ f.a. $i \in \{1, \ldots, n\}$) ordnen wir die σ_Σ-Struktur $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_{a_1}^{\mathcal{A}_w}, \ldots, P_{a_\ell}^{\mathcal{A}_w})$ zu, s.d.:

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:

- einem 2-stelligen Relationssymbol $\leqslant$
- einem 1-stelligen Relationssymbol P_a für jeden Buchstaben $a \in \Sigma$.

(b) Einem endlichen Wort $w_1 \cdots w_n \in \Sigma^+$ der Länge $n \geqslant 1$ (mit $w_i \in \Sigma$ f.a. $i \in \{1, \ldots, n\}$) ordnen wir die σ_Σ-Struktur $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_{a_1}^{\mathcal{A}_w}, \ldots, P_{a_\ell}^{\mathcal{A}_w})$ zu, s.d.:

- $A_w = \{1, \ldots, n\}$ ist die Menge aller Positionen in w

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:
- einem 2-stelligen Relationssymbol $\leqslant$
- einem 1-stelligen Relationssymbol P_a für jeden Buchstaben $a \in \Sigma$.

(b) Einem endlichen Wort $w_1 \cdots w_n \in \Sigma^+$ der Länge $n \geqslant 1$ (mit $w_i \in \Sigma$ f.a. $i \in \{1, \ldots, n\}$) ordnen wir die σ_Σ-Struktur $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_{a_1}^{\mathcal{A}_w}, \ldots, P_{a_\ell}^{\mathcal{A}_w})$ zu, s.d.:
- $A_w = \{1, \ldots, n\}$ ist die Menge aller Positionen in w
- $\leqslant^{\mathcal{A}_w}$ ist die natürliche lineare Ordnung auf A_w

Der Satz von Büchi besagt, dass die regulären Sprachen genau diejenigen Sprachen sind, die in MSO (bzw. EMSO) beschrieben werden können.

Sei $\Sigma := \{a_1, \ldots, a_\ell\}$ ein endliches, nicht-leeres Alphabet.

Nicht-leere Worte $w \in \Sigma^\star$ repräsentieren wir wie folgt.

Definition 2.7 (Wortstrukturen)

(a) Sei σ_Σ die Signatur bestehend aus:
- einem 2-stelligen Relationssymbol $\leqslant$
- einem 1-stelligen Relationssymbol P_a für jeden Buchstaben $a \in \Sigma$.

(b) Einem endlichen Wort $w_1 \cdots w_n \in \Sigma^+$ der Länge $n \geqslant 1$ (mit $w_i \in \Sigma$ f.a. $i \in \{1, \ldots, n\}$) ordnen wir die σ_Σ-Struktur $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_{a_1}^{\mathcal{A}_w}, \ldots, P_{a_\ell}^{\mathcal{A}_w})$ zu, s.d.:
- $A_w = \{1, \ldots, n\}$ ist die Menge aller Positionen in w
- $\leqslant^{\mathcal{A}_w}$ ist die natürliche lineare Ordnung auf A_w
- $P_a^{\mathcal{A}_w} := \{i \in A_w : w_i = a\}$ für jedes $a \in \Sigma$,
  d.h. $P_a^{\mathcal{A}_w}$ ist die Menge aller Positionen in w, an denen der Buchstabe a steht.

Beispiel 2.8

Für $\Sigma = \{a, b\}$ und $w = aaab$ gilt $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_a^{\mathcal{A}_w}, P_b^{\mathcal{A}_w})$ mit

Beispiel 2.8

Für $\Sigma = \{a, b\}$ und $w = aaab$ gilt $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_a^{\mathcal{A}_w}, P_b^{\mathcal{A}_w})$ mit

- $A_w = \{1, 2, 3, 4\}$

Beispiel 2.8

Für $\Sigma = \{a, b\}$ und $w = aaab$ gilt $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_a^{\mathcal{A}_w}, P_b^{\mathcal{A}_w})$ mit

- $A_w = \{1, 2, 3, 4\}$
- $\leqslant^{\mathcal{A}_w}$ ist die natürliche lineare Ordnung auf $\{1, 2, 3, 4\}$

Beispiel 2.8

Für $\Sigma = \{a, b\}$ und $w = aaab$ gilt $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_a^{\mathcal{A}_w}, P_b^{\mathcal{A}_w})$ mit

- $A_w = \{1, 2, 3, 4\}$
- $\leqslant^{\mathcal{A}_w}$ ist die natürliche lineare Ordnung auf $\{1, 2, 3, 4\}$
- $P_a^{\mathcal{A}_w} = \{1, 2, 3\}$

Beispiel 2.8

Für $\Sigma = \{a, b\}$ und $w = aaab$ gilt $\mathcal{A}_w = (A_w, \leqslant^{\mathcal{A}_w}, P_a^{\mathcal{A}_w}, P_b^{\mathcal{A}_w})$ mit

- $A_w = \{1, 2, 3, 4\}$
- $\leqslant^{\mathcal{A}_w}$ ist die natürliche lineare Ordnung auf $\{1, 2, 3, 4\}$
- $P_a^{\mathcal{A}_w} = \{1, 2, 3\}$
- $P_b^{\mathcal{A}_w} = \{4\}$.

Definition 2.9

- Sei $L \subseteq \Sigma^*$ und sei φ ein MSO[σ_Σ]-Satz.

Definition 2.9

- Sei $L \subseteq \Sigma^*$ und sei φ ein MSO[σ_Σ]-Satz. Wir sagen φ **beschreibt** L, falls

Definition 2.9

- Sei $L \subseteq \Sigma^*$ und sei φ ein MSO$[\sigma_\Sigma]$-Satz. Wir sagen φ **beschreibt** L, falls für jedes nicht-leere Wort $w \in \Sigma^\star$ gilt:

$$w \in L \iff \mathcal{A}_w \models \varphi.$$

Definition 2.9

- Sei $L \subseteq \Sigma^*$ und sei φ ein MSO$[\sigma_\Sigma]$-Satz. Wir sagen φ beschreibt L, falls für jedes nicht-leere Wort $w \in \Sigma^\star$ gilt:

$$w \in L \iff \mathcal{A}_w \models \varphi.$$

- $L \subseteq \Sigma^\star$ heißt MSO-definierbar (bzw. EMSO-definierbar), falls

Definition 2.9

- Sei $L \subseteq \Sigma^*$ und sei φ ein MSO[σ_Σ]-Satz. Wir sagen φ beschreibt L, falls für jedes nicht-leere Wort $w \in \Sigma^\star$ gilt:

$$w \in L \iff \mathcal{A}_w \models \varphi.$$

- $L \subseteq \Sigma^\star$ heißt MSO-definierbar (bzw. EMSO-definierbar), falls es einen MSO[σ_Σ]-Satz (bzw. EMSO[σ_Σ]-Satz) gibt, der L beschreibt.

Beispiel 2.10

Sei $\Sigma = \{a, b\}$. Die Sprache $L_{even} := \{w \in \Sigma^* \,:\, |w| \text{ ist gerade}\}$ wird durch folgenden EMSO$[\sigma_\Sigma]$-Satz φ_{even} beschrieben:

Beispiel 2.10

Sei $\Sigma = \{a, b\}$. Die Sprache $L_{even} := \{w \in \Sigma^* : |w| \text{ ist gerade}\}$ wird durch folgenden EMSO$[\sigma_\Sigma]$-Satz φ_{even} beschrieben:

$$\begin{aligned}\varphi_{\text{even}} \ := \ \exists X \Big(& \forall x \left(min(x) \to \neg X(x) \right) \land \\ & \forall x \left(max(x) \to X(x) \right) \land \\ & \forall x \forall y \Big(succ(x,y) \to \left(X(x) \leftrightarrow \neg X(y) \right) \Big) \Big)\end{aligned}$$

mit:
$$\begin{aligned} min(x) &:= \ \forall z \ x \leqslant z \\ max(x) &:= \ \forall z \ z \leqslant x \\ succ(x,y) &:= \ x \leqslant y \ \land \ \neg\, x{=}y \ \land \ \forall z \left(z \leqslant x \lor y \leqslant z \right) \end{aligned}$$

Idee: X soll genau die geraden Positionen und die letzte Position eines Wortes enthalten, in dessen Wortstruktur die Formel ausgewertet wird.

Beispiel 2.10

Sei $\Sigma = \{a, b\}$. Die Sprache $L_{even} := \{w \in \Sigma^* : |w| \text{ ist gerade}\}$ wird durch folgenden EMSO[σ_Σ]-Satz φ_{even} beschrieben:

$$\begin{aligned}\varphi_{\text{even}} \;:=\; \exists X \Big(& \forall x \,\big(min(x) \rightarrow \neg X(x)\big) \wedge \\ & \forall x \,\big(max(x) \rightarrow X(x)\big) \wedge \\ & \forall x \,\forall y \,\Big(succ(x,y) \rightarrow \big(X(x) \leftrightarrow \neg X(y)\big)\Big)\Big)\end{aligned}$$

mit:

$$\begin{aligned} min(x) &:= \forall z \; x \leqslant z \\ max(x) &:= \forall z \; z \leqslant x \\ succ(x,y) &:= x \leqslant y \;\wedge\; \neg\, x{=}y \;\wedge\; \forall z \,\big(z \leqslant x \vee y \leqslant z\big)\end{aligned}$$

<u>Idee</u>: X soll genau die geraden Positionen und die letzte Position eines Wortes enthalten, in dessen Wortstruktur die Formel ausgewertet wird.

Die Abkürzungen $min(x)$, $max(x)$ und $succ(x)$ für oben stehende Formeln werden wir im Verlauf dieser Vorlesung immer wieder benutzen.

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

(b) *L ist* EMSO*-definierbar.*

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

(b) *L ist* EMSO*-definierbar.*

(c) *L ist* MSO*-definierbar.*

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

(b) *L ist* EMSO*-definierbar.*

(c) *L ist* MSO*-definierbar.*

Der „Satz von Büchi“ ist auch unter dem Namen „Satz von Büchi, Elgot, Trakhtenbrot“ bekannt.

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

(b) *L ist* EMSO*-definierbar.*

(c) *L ist* MSO*-definierbar.*

Der „Satz von Büchi“ ist auch unter dem Namen „Satz von Büchi, Elgot, Trakhtenbrot“ bekannt.

Die Richtung „$(b) \Rightarrow (c)$“ gilt trivialerweise, da $\text{EMSO}[\sigma_\Sigma] \subseteq \text{MSO}[\sigma_\Sigma]$.

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

(b) *L ist* EMSO*-definierbar.*

(c) *L ist* MSO*-definierbar.*

Der „Satz von Büchi" ist auch unter dem Namen „Satz von Büchi, Elgot, Trakhtenbrot" bekannt.

Die Richtung „$(b) \Rightarrow (c)$" gilt trivialerweise, da $\text{EMSO}[\sigma_\Sigma] \subseteq \text{MSO}[\sigma_\Sigma]$.

Wir beweisen die Richtung „$(a) \Rightarrow (b)$" in dem folgenden Lemma.

Theorem 2.11 (Satz von Büchi)

Sei Σ ein endliches, nicht-leeres Alphabet und sei $L \subseteq \Sigma^*$.

Die folgenden Aussagen sind äquivalent:

(a) *L ist regulär (d.h. L wird von einem deterministischen endlichen Automaten erkannt).*

(b) *L ist* EMSO*-definierbar.*

(c) *L ist* MSO*-definierbar.*

Der „Satz von Büchi" ist auch unter dem Namen „Satz von Büchi, Elgot, Trakhtenbrot" bekannt.

Die Richtung „$(b) \Rightarrow (c)$" gilt trivialerweise, da $\mathrm{EMSO}[\sigma_\Sigma] \subseteq \mathrm{MSO}[\sigma_\Sigma]$.

Wir beweisen die Richtung „$(a) \Rightarrow (b)$" in dem folgenden Lemma.

Lemma 2.12

Jede reguläre Sprache $L \subseteq \Sigma^*$ ist EMSO*-definierbar.*

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar.

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO$[\sigma_\Sigma]$-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Die Richtung „$(c) \Rightarrow (a)$" des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO[σ_Σ]-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$.

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO$[\sigma_\Sigma]$-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO$[\sigma_\Sigma]$-Formeln:

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO[σ_Σ]-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO[σ_Σ]-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set“):

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO[σ_Σ]-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.
Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO[σ_Σ]-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set“):

$$\mathit{singl}(X) \;:=\; \exists y \Big(X(y) \land \forall z \,\big(X(z) \,\to\, z{=}y \big) \Big)$$

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO[σ_Σ]-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO[σ_Σ]-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set“):

$$\mathit{singl}(X) \;:=\; \exists y \Big(X(y) \land \forall z \big(X(z) \to z{=}y \big) \Big)$$

- $\mathit{le}(X, Y)$ besagt, dass keine Position in Y kleiner als eine Position in X ist:

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO$[\sigma_\Sigma]$-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.
Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO$[\sigma_\Sigma]$-Formeln:

- $singl(X)$ besagt $|X| = 1$ („X is a singleton set“):

$$singl(X) \;:=\; \exists y \Big(X(y) \wedge \forall z \big(X(z) \rightarrow z{=}y \big) \Big)$$

- $le(X, Y)$ besagt, dass keine Position in Y kleiner als eine Position in X ist:

$$le(X, Y) \;:= \forall x \, \forall y \Big(\big(X(x) \wedge Y(y) \big) \rightarrow x \leqslant y \Big)$$

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO$[\sigma_\Sigma]$-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO$[\sigma_\Sigma]$-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set“):

$$\mathit{singl}(X) \ := \ \exists y \Big(X(y) \land \forall z \, \big(X(z) \to z{=}y \big) \Big)$$

- $\mathit{le}(X, Y)$ besagt, dass keine Position in Y kleiner als eine Position in X ist:

$$\mathit{le}(X, Y) \ := \forall x \, \forall y \Big(\big(X(x) \land Y(y) \big) \to x \leqslant y \Big)$$

- $\mathit{sub}(X, Y)$ besagt $X \subseteq Y$:

Die Richtung „$(c) \Rightarrow (a)$" des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO$[\sigma_\Sigma]$-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO$[\sigma_\Sigma]$-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set"):

$$\mathit{singl}(X) \ := \ \exists y \Big(X(y) \land \forall z \left(X(z) \to z{=}y \right) \Big)$$

- $\mathit{le}(X, Y)$ besagt, dass keine Position in Y kleiner als eine Position in X ist:

$$\mathit{le}(X, Y) \ := \forall x \, \forall y \Big(\big(X(x) \land Y(y) \big) \to x \leqslant y \Big)$$

- $\mathit{sub}(X, Y)$ besagt $X \subseteq Y$:

$$\mathit{sub}(X, Y) \ := \ \forall x \left(X(x) \to Y(x) \right)$$

Die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO[σ_Σ]-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.

Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO[σ_Σ]-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set“):

$$\mathit{singl}(X) \ := \ \exists y \Big(X(y) \land \forall z \big(X(z) \rightarrow z{=}y \big) \Big)$$

- $\mathit{le}(X, Y)$ besagt, dass keine Position in Y kleiner als eine Position in X ist:

$$\mathit{le}(X, Y) \ := \forall x \, \forall y \Big(\big(X(x) \land Y(y) \big) \rightarrow x \leqslant y \Big)$$

- $\mathit{sub}(X, Y)$ besagt $X \subseteq Y$:

$$\mathit{sub}(X, Y) \ := \ \forall x \big(X(x) \rightarrow Y(x) \big)$$

- $\mathit{symb}_a(X)$ besagt, dass an allen Positionen in X der Buchstabe a steht:

Die Richtung „$(c) \Rightarrow (a)$" des Satzes von Büchi (Theorem 2.11) stellt den schwierigsten Teil des Beweises dar. Wir bringen hierfür einen gegebenen MSO[σ_Σ]-Satz zunächst in eine für die Übersetzung in einen endlichen Automaten besonders geeignete Form.
Seien X und Y Mengenvariablen und $a \in \Sigma$. Wir nutzen die folgenden MSO[σ_Σ]-Formeln:

- $\mathit{singl}(X)$ besagt $|X| = 1$ („X is a singleton set"):
$$\mathit{singl}(X) \ := \ \exists y \Big(X(y) \wedge \forall z \left(X(z) \to z{=}y \right) \Big)$$
- $\mathit{le}(X, Y)$ besagt, dass keine Position in Y kleiner als eine Position in X ist:
$$\mathit{le}(X, Y) \ := \forall x \, \forall y \Big(\big(X(x) \wedge Y(y) \big) \to x \leqslant y \Big)$$
- $\mathit{sub}(X, Y)$ besagt $X \subseteq Y$:
$$\mathit{sub}(X, Y) \ := \ \forall x \left(X(x) \to Y(x) \right)$$
- $\mathit{symb}_a(X)$ besagt, dass an allen Positionen in X der Buchstabe a steht:
$$\mathit{symb}_a(X) \ := \ \forall x \left(X(x) \to P_a(x) \right).$$

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und symb_a ersetzen, Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und symb_a ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \mathrm{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \mathrm{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \mathrm{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und symb_a ersetzen, s.d. für alle $x, y \in \mathrm{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \mathrm{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \mathrm{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, le(V_x, V_y) \wedge le(V_y, V_x)\,\big)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \mathrm{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \mathrm{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \mathrm{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(le(V_x, V_y) \wedge le(V_y, V_x) \big)$.
- Für $\varphi := P_a(x)$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und symb_a ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, \mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x) \,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und symb_a ersetzen, s.d. für alle $x, y \in \mathrm{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \mathrm{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \mathrm{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, \mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x) \,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(le(V_x, V_y) \wedge le(V_y, V_x) \big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, le(V_x, V_y) \wedge le(V_y, V_x)\,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und symb_a ersetzen, s.d. für alle $x, y \in \mathrm{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \mathrm{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \mathrm{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, \mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x)\,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := \mathit{sub}(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \,\big(\mathit{singl}(V_x) \wedge \psi^*\big)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \mathrm{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \mathrm{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \mathrm{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(le(V_x, V_y) \wedge le(V_y, V_x) \big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \big(singl(V_x) \wedge \psi^* \big)$.
- Für $\varphi := \forall x\, \psi$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, le(V_x, V_y) \wedge le(V_y, V_x)\,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \,\big(singl(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x \,\big(singl(V_x) \rightarrow \psi^*\big)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und *symb*$_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x)\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := \mathit{sub}(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x\, \big(\mathit{singl}(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x\, \big(\mathit{singl}(V_x) \rightarrow \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und *symb*$_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \to, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, \mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x) \,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := \mathit{sub}(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \,\big(\mathit{singl}(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x \,\big(\mathit{singl}(V_x) \to \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und *symb*$_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, le(V_x, V_y) \wedge le(V_y, V_x)\,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x\, \big(singl(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x\, \big(singl(V_x) \rightarrow \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.
- Für $\varphi := (\varphi_1 \,\Box\, \varphi_2)$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\, le(V_x, V_y) \wedge le(V_y, V_x)\,\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \,\big(singl(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x \,\big(singl(V_x) \rightarrow \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.
- Für $\varphi := (\varphi_1 \Box \varphi_2)$ ist $\varphi^* := (\varphi_1^* \Box \varphi_2^*)$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und $symb_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := le(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(le(V_x, V_y) \wedge le(V_y, V_x) \big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := symb_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := sub(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \big(singl(V_x) \wedge \psi^* \big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x \big(singl(V_x) \rightarrow \psi^* \big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.
- Für $\varphi := (\varphi_1 \Box \varphi_2)$ ist $\varphi^* := (\varphi_1^* \Box \varphi_2^*)$.
- Für $\varphi := QZ\, \psi$ ist

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente“ MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und *symb*$_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x)\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := \mathit{sub}(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x \big(\mathit{singl}(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x \big(\mathit{singl}(V_x) \rightarrow \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.
- Für $\varphi := (\varphi_1 \Box \varphi_2)$ ist $\varphi^* := (\varphi_1^* \Box \varphi_2^*)$.
- Für $\varphi := QZ\, \psi$ ist $\varphi^* := QZ\, \psi^*$.

Eine gegebene MSO[σ_Σ]-Formel φ transformieren wir nun in eine „äquivalente" MSO[σ_Σ]-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und *symb*$_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\Box \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x)\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := \mathit{sub}(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x\, \big(\mathit{singl}(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x\, \big(\mathit{singl}(V_x) \rightarrow \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.
- Für $\varphi := (\varphi_1 \,\Box\, \varphi_2)$ ist $\varphi^* := (\varphi_1^* \,\Box\, \varphi_2^*)$.
- Für $\varphi := QZ\,\psi$ ist $\varphi^* := QZ\,\psi^*$.

Lemma 2.13

Für jeden MSO[σ_Σ]-*Satz* φ *ist auch* φ^* *ein* MSO[σ_Σ]-*Satz, und* φ^* *beschreibt dieselbe Sprache wie* φ.

Eine gegebene MSO$[\sigma_\Sigma]$-Formel φ transformieren wir nun in eine „äquivalente" MSO$[\sigma_\Sigma]$-Formel φ^*, indem wir jede Individuenvariable x durch eine neue Mengenvariable V_x ersetzen und atomare Formeln durch geeignete Kombinationen der Formeln *singl*, *le*, *sub* und *symb*$_a$ ersetzen, s.d. für alle $x, y \in \text{Var}_1$, $a \in \Sigma$, Mengenvariablen $Z \in \text{Var}_2$, Formeln $\psi, \varphi_1, \varphi_2 \in \text{MSO}[\sigma_\Sigma]$, Junktoren $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ und Quantoren $Q \in \{\exists, \forall\}$ gilt:

- Für $\varphi := x \leqslant y$ ist $\varphi^* := \mathit{le}(V_x, V_y)$.
- Für $\varphi := x{=}y$ ist $\varphi^* := \big(\mathit{le}(V_x, V_y) \wedge \mathit{le}(V_y, V_x)\big)$.
- Für $\varphi := P_a(x)$ ist $\varphi^* := \mathit{symb}_a(V_x)$.
- Für $\varphi := Z(x)$ ist $\varphi^* := \mathit{sub}(V_x, Z)$.
- Für $\varphi := \exists x\, \psi$ ist $\varphi^* := \exists V_x\, \big(\mathit{singl}(V_x) \wedge \psi^*\big)$.
- Für $\varphi := \forall x\, \psi$ ist $\varphi^* := \forall V_x\, \big(\mathit{singl}(V_x) \rightarrow \psi^*\big)$.
- Für $\varphi := \neg\psi$ ist $\varphi^* := \neg\psi^*$.
- Für $\varphi := (\varphi_1 \,\square\, \varphi_2)$ ist $\varphi^* := (\varphi_1^* \,\square\, \varphi_2^*)$.
- Für $\varphi := QZ\,\psi$ ist $\varphi^* := QZ\,\psi^*$.

Lemma 2.13

Für jeden MSO$[\sigma_\Sigma]$*-Satz* φ *ist auch* φ^* *ein* MSO$[\sigma_\Sigma]$*-Satz, und* φ^* *beschreibt dieselbe Sprache wie* φ*.*

Beweis.

Übung. □

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten“ endlichen Automaten.

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten“ endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X, Y)$, $\mathit{sub}(X, Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar“.

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten" endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X,Y)$, $\mathit{sub}(X,Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar".

Insbesondere haben die Teilformeln, die wir betrachten müssen, keine freien Individuenvariablen, wohl aber freie Mengenvariablen.

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten" endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X, Y)$, $\mathit{sub}(X, Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar".

Insbesondere haben die Teilformeln, die wir betrachten müssen, keine freien Individuenvariablen, wohl aber freie Mengenvariablen. Belegungen dieser Mengenvariablen repräsentieren wir durch ein größeres Alphabet:

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten“ endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X, Y)$, $\mathit{sub}(X, Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar“.

Insbesondere haben die Teilformeln, die wir betrachten müssen, keine freien Individuenvariablen, wohl aber freie Mengenvariablen. Belegungen dieser Mengenvariablen repräsentieren wir durch ein größeres Alphabet:

Seien $X_1, \ldots, X_k$ (mit $k \in \mathbb{N}_{\geqslant 1}$) die Mengenvariablen, die in φ^* vorkommen.

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten“ endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X, Y)$, $\mathit{sub}(X, Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar“.

Insbesondere haben die Teilformeln, die wir betrachten müssen, keine freien Individuenvariablen, wohl aber freie Mengenvariablen. Belegungen dieser Mengenvariablen repräsentieren wir durch ein größeres Alphabet:

Seien $X_1, \ldots, X_k$ (mit $k \in \mathbb{N}_{\geqslant 1}$) die Mengenvariablen, die in φ^* vorkommen. Sei

$$\Sigma' \;:=\; \Sigma \times \{0,1\}^k.$$

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten" endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X, Y)$, $\mathit{sub}(X, Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar".

Insbesondere haben die Teilformeln, die wir betrachten müssen, keine freien Individuenvariablen, wohl aber freie Mengenvariablen. Belegungen dieser Mengenvariablen repräsentieren wir durch ein größeres Alphabet:

Seien $X_1, \ldots, X_k$ (mit $k \in \mathbb{N}_{\geqslant 1}$) die Mengenvariablen, die in φ^* vorkommen. Sei

$$\Sigma' \ := \ \Sigma \times \{0,1\}^k.$$

Ein Wort $w_1 \cdots w_n \in \Sigma^+$ (mit $w_i \in \Sigma$ f.a. $i \in [n]$) und Belegungen $S_1, \ldots, S_k \subseteq \{1, \ldots, n\}$ der Mengenvariablen $X_1, \ldots, X_k$ repräsentieren wir

Als nächstes konstruieren wir induktiv für jede Teilformel von φ^* einen „äquivalenten" endlichen Automaten. Formeln der Form $\mathit{singl}(X)$, $\mathit{le}(X, Y)$, $\mathit{sub}(X, Y)$ und $\mathit{symb}_a(X)$ behandeln wir dabei als „atomar".

Insbesondere haben die Teilformeln, die wir betrachten müssen, keine freien Individuenvariablen, wohl aber freie Mengenvariablen. Belegungen dieser Mengenvariablen repräsentieren wir durch ein größeres Alphabet:

Seien $X_1, \ldots, X_k$ (mit $k \in \mathbb{N}_{\geqslant 1}$) die Mengenvariablen, die in φ^* vorkommen. Sei

$$\Sigma' \;:=\; \Sigma \times \{0,1\}^k.$$

Ein Wort $w_1 \cdots w_n \in \Sigma^+$ (mit $w_i \in \Sigma$ f.a. $i \in [n]$) und Belegungen $S_1, \ldots, S_k \subseteq \{1, \ldots, n\}$ der Mengenvariablen $X_1, \ldots, X_k$ repräsentieren wir durch das Wort

$$u \;:=\; u_1 \cdots u_n \;\in\; \left(\Sigma'\right)^+$$

mit

$$u_i \;:=\; \left(w_i,\, b_{i,1}, \ldots,\, b_{i,k}\right) \in \Sigma' \text{ für alle } i \in \{1, \ldots, n\},$$

wobei für alle $i \in \{1, \ldots, n\}$ und $j \in \{1, \ldots, k\}$ gilt:

$$b_{i,j} := \begin{cases} 1 & \text{falls } i \in S_j \\ 0 & \text{falls } i \notin S_j. \end{cases}$$

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \left(\Sigma'\right)^+$

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \left(\Sigma'\right)^+$ ein Wort

$$w = w_1 \cdots w_n \in \Sigma^+$$

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \left(\Sigma'\right)^+$ ein Wort

$$w \;=\; w_1 \cdots w_n \;\in\; \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt:

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \left(\Sigma'\right)^+$ ein Wort

$$w \;=\; w_1 \cdots w_n \;\in\; \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt: Ist, für alle $i \in \{1, \ldots, n\}$, $u_i = (\sigma_i,\, b_{i,1}, \ldots, b_{i,k})$, so ist

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \left(\Sigma'\right)^+$ ein Wort

$$w \;=\; w_1 \cdots w_n \;\in\; \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt: Ist, für alle $i \in \{1, \ldots, n\}$, $u_i = (\sigma_i, b_{i,1}, \ldots, b_{i,k})$, so ist

$$w_i = \sigma_i \;\text{ für alle } i \in [n]$$

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \left(\Sigma'\right)^+$ ein Wort

$$w \;=\; w_1 \cdots w_n \;\in\; \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt: Ist, für alle $i \in \{1, \ldots, n\}$, $u_i = (\sigma_i,\, b_{i,1}, \ldots, b_{i,k})$, so ist

$$w_i = \sigma_i \;\text{ für alle } i \in [n]$$

und

$$S_j \;=\; \big\{ i \in [n] \,:\, b_{i,j} = 1 \big\} \;\text{ für alle } j \in [k].$$

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \big(\Sigma'\big)^+$ ein Wort

$$w \;=\; w_1 \cdots w_n \;\in\; \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt: Ist, für alle $i \in \{1, \ldots, n\}$, $u_i = (\sigma_i,\, b_{i,1}, \ldots, b_{i,k})$, so ist

$$w_i \;=\; \sigma_i \;\; \text{für alle } i \in [n]$$

und

$$S_j \;=\; \big\{ i \in [n] \,:\, b_{i,j} = 1 \big\} \;\; \text{für alle } j \in [k].$$

Setze dann $\mathcal{I}_u := \big(\mathcal{A}_w, S_1, \ldots, S_k\big)$.

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \big(\Sigma'\big)^+$ ein Wort

$$w = w_1 \cdots w_n \in \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt: Ist, für alle $i \in \{1, \ldots, n\}$, $u_i = (\sigma_i,\ b_{i,1}, \ldots, b_{i,k})$, so ist

$$w_i = \sigma_i \ \text{ für alle } i \in [n]$$

und

$$S_j = \big\{ i \in [n] \ : \ b_{i,j} = 1 \big\} \ \text{ für alle } j \in [k].$$

Setze dann $\mathcal{I}_u := \big(\mathcal{A}_w, S_1, \ldots, S_k\big)$.

Eine Teilformel ψ von φ^* definiert die Sprache

Umgekehrt repräsentiert jedes Wort $u = u_1 \cdots u_n \in \big(\Sigma'\big)^+$ ein Wort

$$w \;=\; w_1 \cdots w_n \;\in\; \Sigma^+$$

und Belegungen

$$S_1, \ldots, S_k \subseteq [n]$$

der Mengenvariablen $X_1, \ldots, X_k$ wie folgt: Ist, für alle $i \in \{1, \ldots, n\}$, $u_i \;=\; (\sigma_i,\, b_{i,1}, \ldots, b_{i,k})$, so ist

$$w_i \;=\; \sigma_i \;\text{ für alle } i \in [n]$$

und

$$S_j \;=\; \big\{ i \in [n] \,:\, b_{i,j} = 1 \big\} \;\text{ für alle } j \in [k].$$

Setze dann $\mathcal{I}_u := \big(\mathcal{A}_w, S_1, \ldots, S_k\big)$.

Eine Teilformel ψ von φ^* definiert die Sprache

$$L'(\psi) \;:=\; \big\{ u \in \big(\Sigma'\big)^+ \,:\, \mathcal{I}_u \models \psi \big\}.$$

Lemma 2.14

Sei $k \in \mathbb{N}_{\geqslant 1}$, $\Sigma' := \Sigma \times \{0,1\}^k$, *seien* $i, j \in [k]$, *sei* $a \in \Sigma$.

Lemma 2.14

Sei $k \in \mathbb{N}_{\geqslant 1}$, $\Sigma' := \Sigma \times \{0,1\}^k$, seien $i, j \in [k]$, sei $a \in \Sigma$.

Es gibt nichtdeterminstische endliche Automaten $\mathbb{A}_{singl(X_i)}$, $\mathbb{A}_{le(X_i,X_j)}$, $\mathbb{A}_{sub(X_i,X_j)}$ und $\mathbb{A}_{symb_a(X_i)}$ über dem Alphabet Σ', so dass gilt:

Lemma 2.14

Sei $k \in \mathbb{N}_{\geqslant 1}$, $\Sigma' := \Sigma \times \{0,1\}^k$, seien $i, j \in [k]$, sei $a \in \Sigma$.

Es gibt nichtdeterminstische endliche Automaten $\mathbb{A}_{singl(X_i)}$, $\mathbb{A}_{le(X_i,X_j)}$, $\mathbb{A}_{sub(X_i,X_j)}$ und $\mathbb{A}_{symb_a(X_i)}$ über dem Alphabet Σ', so dass gilt:

(a) *$\mathbb{A}_{singl(X_i)}$ erkennt die Sprache $L'\big(singl(X_i)\big)$.*

Lemma 2.14

Sei $k \in \mathbb{N}_{\geqslant 1}$, $\Sigma' := \Sigma \times \{0,1\}^k$, seien $i, j \in [k]$, sei $a \in \Sigma$.

Es gibt nichtdeterminstische endliche Automaten $\mathbb{A}_{singl(X_i)}$, $\mathbb{A}_{le(X_i,X_j)}$, $\mathbb{A}_{sub(X_i,X_j)}$ und $\mathbb{A}_{symb_a(X_i)}$ über dem Alphabet Σ', so dass gilt:

(a) $\mathbb{A}_{singl(X_i)}$ *erkennt die Sprache* $L'\big(singl(X_i)\big)$.

(b) $\mathbb{A}_{le(X_i,X_j)}$ *erkennt die Sprache* $L'\big(le(X_i, X_j)\big)$.

Lemma 2.14

Sei $k \in \mathbb{N}_{\geqslant 1}$, $\Sigma' := \Sigma \times \{0,1\}^k$, *seien* $i, j \in [k]$, *sei* $a \in \Sigma$.

Es gibt nichtdeterminstische endliche Automaten $\mathbb{A}_{singl(X_i)}$, $\mathbb{A}_{le(X_i,X_j)}$, $\mathbb{A}_{sub(X_i,X_j)}$ *und* $\mathbb{A}_{symb_a(X_i)}$ *über dem Alphabet* Σ', *so dass gilt:*

(a) $\mathbb{A}_{singl(X_i)}$ *erkennt die Sprache* $L'\big(singl(X_i)\big)$.

(b) $\mathbb{A}_{le(X_i,X_j)}$ *erkennt die Sprache* $L'\big(le(X_i,X_j)\big)$.

(c) $\mathbb{A}_{sub(X_i,X_j)}$ *erkennt die Sprache* $L'\big(sub(X_i,X_j)\big)$.

Lemma 2.14

Sei $k \in \mathbb{N}_{\geqslant 1}$, $\Sigma' := \Sigma \times \{0,1\}^k$, seien $i, j \in [k]$, sei $a \in \Sigma$.

Es gibt nichtdeterminstische endliche Automaten $\mathbb{A}_{singl(X_i)}$, $\mathbb{A}_{le(X_i,X_j)}$, $\mathbb{A}_{sub(X_i,X_j)}$ und $\mathbb{A}_{symb_a(X_i)}$ über dem Alphabet Σ', so dass gilt:

(a) *$\mathbb{A}_{singl(X_i)}$ erkennt die Sprache $L'\big(singl(X_i)\big)$.*

(b) *$\mathbb{A}_{le(X_i,X_j)}$ erkennt die Sprache $L'\big(le(X_i, X_j)\big)$.*

(c) *$\mathbb{A}_{sub(X_i,X_j)}$ erkennt die Sprache $L'\big(sub(X_i, X_j)\big)$.*

(d) *$\mathbb{A}_{symb_a(X_i)}$ erkennt die Sprache $L'\big(symb_a(X_i)\big)$.*

Lemma 2.15

Sei φ *ein* MSO$[\sigma_\Sigma]$*-Satz, seien* $X_1, \ldots, X_k$ *(für* $k \in \mathbb{N}_{\geqslant 1}$*) die in* φ^* *vorkommenden Mengenvariablen, und sei* $\Sigma' := \Sigma \times \{0,1\}^k$*.*

Lemma 2.15

Sei φ ein $\mathrm{MSO}[\sigma_\Sigma]$-Satz, seien $X_1, \ldots, X_k$ (für $k \in \mathbb{N}_{\geqslant 1}$) die in φ^* vorkommenden Mengenvariablen, und sei $\Sigma' := \Sigma \times \{0,1\}^k$.

Für jede Teilformel ψ von φ^* (wobei die Formeln $singl(X_i)$, $sub(X_i, X_j)$, $le(X_i, X_j)$, $symb_a(X_i)$ als „atomar“ betrachtet werden)

Lemma 2.15

Sei φ ein $\mathrm{MSO}[\sigma_\Sigma]$-Satz, seien $X_1, \ldots, X_k$ (für $k \in \mathbb{N}_{\geqslant 1}$) die in φ^* vorkommenden Mengenvariablen, und sei $\Sigma' := \Sigma \times \{0,1\}^k$.

Für jede Teilformel ψ von φ^* (wobei die Formeln $singl(X_i)$, $sub(X_i, X_j)$, $le(X_i, X_j)$, $symb_a(X_i)$ als „atomar" betrachtet werden) gibt es einen nichtdeterministischen Automaten $\mathbb{A}_\psi$ über dem Alphabet Σ', sodass gilt:

Lemma 2.15

Sei φ ein $\mathrm{MSO}[\sigma_\Sigma]$-Satz, seien $X_1, \ldots, X_k$ (für $k \in \mathbb{N}_{\geqslant 1}$) die in φ^* vorkommenden Mengenvariablen, und sei $\Sigma' := \Sigma \times \{0,1\}^k$.

Für jede Teilformel ψ von φ^* (wobei die Formeln $singl(X_i)$, $sub(X_i, X_j)$, $le(X_i, X_j)$, $symb_a(X_i)$ als „atomar" betrachtet werden) gibt es einen nichtdeterministischen Automaten $\mathbb{A}_\psi$ über dem Alphabet Σ', sodass gilt:

$$\mathbb{A}_\psi \text{ erkennt die Sprache } L'(\psi).$$

Das folgende Lemma beweist die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11).

Das folgende Lemma beweist die Richtung „$(c) \Rightarrow (a)$“ des Satzes von Büchi (Theorem 2.11).

Lemma 2.16

Für jeden $\mathrm{MSO}[\sigma_\Sigma]$*-Satz* φ *gibt es einen nichtdeterministischen endlichen Automaten* $\mathbb{A}_\varphi$*, der die von* φ *beschriebene Sprache erkennt.*

Bemerkung 2.17

Der Satz von Büchi besagt insbesondere, dass existenzielle monadische Logik zweiter Stufe auf Worten dieselbe Ausdrucksstärke besitzt wie die „volle“ monadische Logik zweiter Stufe (und genau die regulären Sprachen beschreiben kann).

Bemerkung 2.17

Der Satz von Büchi besagt insbesondere, dass existenzielle monadische Logik zweiter Stufe auf Worten dieselbe Ausdrucksstärke besitzt wie die „volle“ monadische Logik zweiter Stufe (und genau die regulären Sprachen beschreiben kann).

Man kann sogar beweisen, dass jede reguläre Sprache L durch einen EMSO$[\sigma_\Sigma]$-Satz beschrieben werden kann, der nur <u>eine</u> Mengenvariable benutzt. Somit ist auf Worten die existentielle monadische Logik zweiter Stufe mit nur einer einzigen Mengenvariablen äquivalent zur vollen monadischen Logik zweiter Stufe.

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Insbesondere wissen wir für $\Sigma = \{a, b\}$, dass die Sprache

$$L_{a^n b^n} := \{a^n b^n \,:\, n \in \mathbb{N}_{\geqslant 1}\}$$

nicht regulär ist.

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Insbesondere wissen wir für $\Sigma = \{a, b\}$, dass die Sprache

$$L_{a^n b^n} := \{a^n b^n \, : \, n \in \mathbb{N}_{\geqslant 1}\}$$

nicht regulär ist.

Gemäß dem Satz von Büchi kann $L_{a^n b^n}$ also auch nicht durch einen MSO-Satz beschrieben werden.

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Insbesondere wissen wir für $\Sigma = \{a, b\}$, dass die Sprache

$$L_{a^n b^n} := \{ a^n b^n \, : \, n \in \mathbb{N}_{\geqslant 1} \}$$

nicht regulär ist.

Gemäß dem Satz von Büchi kann $L_{a^n b^n}$ also auch nicht durch einen MSO-Satz beschrieben werden.

Durch Anwenden einer logischen Reduktion können wir daraus folgern, dass z.B. bestimmte Graphen-Eigenschaften nicht in MSO beschrieben werden können:

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Insbesondere wissen wir für $\Sigma = \{a, b\}$, dass die Sprache

$$L_{a^n b^n} := \{a^n b^n \,:\, n \in \mathbb{N}_{\geqslant 1}\}$$

nicht regulär ist.

Gemäß dem Satz von Büchi kann $L_{a^n b^n}$ also auch nicht durch einen MSO-Satz beschrieben werden.

Durch Anwenden einer logischen Reduktion können wir daraus folgern, dass z.B. bestimmte Graphen-Eigenschaften nicht in MSO beschrieben werden können:

Satz 2.18 („Hamiltonkreis ist nicht MSO-definierbar“)

Sei $\sigma_{\text{Graph}} = \{E\}$ für ein 2-stelliges Relationssymbol E.

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Insbesondere wissen wir für $\Sigma = \{a, b\}$, dass die Sprache

$$L_{a^n b^n} := \{a^n b^n \,:\, n \in \mathbb{N}_{\geqslant 1}\}$$

nicht regulär ist.

Gemäß dem Satz von Büchi kann $L_{a^n b^n}$ also auch nicht durch einen MSO-Satz beschrieben werden.

Durch Anwenden einer logischen Reduktion können wir daraus folgern, dass z.B. bestimmte Graphen-Eigenschaften nicht in MSO beschrieben werden können:

Satz 2.18 („Hamiltonkreis ist nicht MSO-definierbar")

Sei $\sigma_{\text{Graph}} = \{E\}$ *für ein 2-stelliges Relationssymbol* E*.*

Es gibt keinen $\text{MSO}[\sigma_{\text{Graph}}]$*-Satz* φ*, s.d.*

Aus dem Gebiet der formalen Sprachen sind viele Methoden bekannt, mit denen man nachweisen kann, dass bestimmte Sprachen nicht regulär sind (z.B. Pumping Lemma, Satz von Myhill und Nerode, Abschlusseigenschaften, ...). Man siehe dazu einführende Vorlesungen in Theoretische Informatik, formale Sprachen bzw. Automatentheorie.

Insbesondere wissen wir für $\Sigma = \{a, b\}$, dass die Sprache

$$L_{a^n b^n} := \{a^n b^n \, : \, n \in \mathbb{N}_{\geqslant 1}\}$$

nicht regulär ist.

Gemäß dem Satz von Büchi kann $L_{a^n b^n}$ also auch nicht durch einen MSO-Satz beschrieben werden.

Durch Anwenden einer logischen Reduktion können wir daraus folgern, dass z.B. bestimmte Graphen-Eigenschaften nicht in MSO beschrieben werden können:

Satz 2.18 („Hamiltonkreis ist nicht MSO-definierbar“)

Sei $\sigma_{\text{Graph}} = \{E\}$ für ein 2-stelliges Relationssymbol E.

Es gibt keinen MSO[σ_{Graph}]*-Satz φ, s.d. für jeden endlichen Graphen $\mathcal{G}$ gilt:*

$$\mathcal{G} \models \varphi \iff \mathcal{G} \text{ besitzt einen Hamiltonkreis.}$$

Abschnitt 2.3:

ESO und der Satz von Fagin

Zur Erinnerung:
ESO-Formeln sind SO-Formeln der Form

$$\exists X_1 \cdots \exists X_\ell \, \varphi,$$

wobei $\ell \geqslant 0$ ist, $X_1, \ldots, X_\ell$ Relationsvariablen beliebiger Stelligkeit sind und φ eine FO-Formel ist, deren atomare Teilformeln auch Relationsvariablen nutzen können.

Zur Erinnerung:
ESO-Formeln sind SO-Formeln der Form

$$\exists X_1 \cdots \exists X_\ell \, \varphi,$$

wobei $\ell \geqslant 0$ ist, $X_1, \ldots, X_\ell$ Relationsvariablen beliebiger Stelligkeit sind und φ eine FO-Formel ist, deren atomare Teilformeln auch Relationsvariablen nutzen können.

Definition 2.19 (Logische Beschreibung von Komplexitätsklassen)

Sei K eine Komplexitätsklasse (z.B. NP), sei $\mathcal{L}$ eine Logik (z.B. ESO) und sei $\mathcal{S}$ eine unter Isomorphie abgeschlossene Klasse endlicher Strukturen (z.B. FIN := die Klasse aller endlichen Strukturen über allen endlichen funktionenfreien Signaturen).

Zur Erinnerung:
ESO-Formeln sind SO-Formeln der Form

$$\exists X_1 \cdots \exists X_\ell \, \varphi,$$

wobei $\ell \geqslant 0$ ist, $X_1, \ldots, X_\ell$ Relationsvariablen beliebiger Stelligkeit sind und φ eine FO-Formel ist, deren atomare Teilformeln auch Relationsvariablen nutzen können.

Definition 2.19 (Logische Beschreibung von Komplexitätsklassen)

Sei K eine Komplexitätsklasse (z.B. NP), sei $\mathcal{L}$ eine Logik (z.B. ESO) und sei $\mathcal{S}$ eine unter Isomorphie abgeschlossene Klasse endlicher Strukturen (z.B. FIN := die Klasse aller endlichen Strukturen über allen endlichen funktionenfreien Signaturen).

Wir sagen „$\mathcal{L}$ beschreibt K auf $\mathcal{S}$“, falls die folgenden beiden Bedingungen erfüllt sind:

(1) Für jeden Satz $\varphi \in \mathcal{L}$ gehört das folgende Problem zur Komplexitätsklasse K:

EVAL$_\varphi(\mathcal{S})$ *(Das Auswertungsproblem für φ auf $\mathcal{S}$)*

Eingabe: Eine Struktur $\mathcal{A} \in \mathcal{S}$

Frage: Gilt $\mathcal{A} \models \varphi$?

(1) Für jeden Satz $\varphi \in \mathcal{L}$ gehört das folgende Problem zur Komplexitätsklasse K:

$EVAL_\varphi(\mathcal{S})$ (Das Auswertungsproblem für φ auf $\mathcal{S}$)	
Eingabe:	Eine Struktur $\mathcal{A} \in \mathcal{S}$
Frage:	Gilt $\mathcal{A} \models \varphi$?

(2) Für jede endliche, funktionenfreie Signatur σ und jede unter Isomorphie abgeschlossene Klasse $\mathcal{C} \subseteq \mathcal{S}$ von σ-Strukturen gilt:

Falls das Problem

Zugehörigkeit zu $\mathcal{C}$ in $\mathcal{S}$	
Eingabe:	Eine σ-Struktur $\mathcal{A} \in \mathcal{S}$
Frage:	Ist $\mathcal{A} \in \mathcal{C}$?

zur Komplexitätsklasse K gehört,

(1) Für jeden Satz $\varphi \in \mathcal{L}$ gehört das folgende Problem zur Komplexitätsklasse K:

EVAL$_\varphi$($\mathcal{S}$) (Das Auswertungsproblem für φ auf $\mathcal{S}$)
Eingabe: Eine Struktur $\mathcal{A} \in \mathcal{S}$
Frage: Gilt $\mathcal{A} \models \varphi$?

(2) Für jede endliche, funktionenfreie Signatur σ und jede unter Isomorphie abgeschlossene Klasse $\mathcal{C} \subseteq \mathcal{S}$ von σ-Strukturen gilt:

Falls das Problem

Zugehörigkeit zu $\mathcal{C}$ in $\mathcal{S}$
Eingabe: Eine σ-Struktur $\mathcal{A} \in \mathcal{S}$
Frage: Ist $\mathcal{A} \in \mathcal{C}$?

zur Komplexitätsklasse K gehört,

so gibt es einen $\mathcal{L}[\sigma]$-Satz φ, sodass gilt:

$$\mathcal{C} = \underbrace{\{\mathcal{A} \in \mathcal{S} : \mathcal{A} \text{ ist eine } \sigma\text{-Struktur mit } \mathcal{A} \models \varphi\}}_{=:\,\mathrm{Mod}_{\mathcal{S}}(\varphi)}$$

Theorem 2.20 (Der Satz von Fagin, 1974)

ESO *beschreibt* NP *auf der Klasse FIN aller endlicher Strukturen.*

Theorem 2.20 (Der Satz von Fagin, 1974)

ESO *beschreibt* NP *auf der Klasse FIN aller endlicher Strukturen.*

Der Beweis von Theorem 2.20 erfolgt in zwei Teilen:

Theorem 2.20 (Der Satz von Fagin, 1974)

ESO *beschreibt* NP *auf der Klasse FIN aller endlicher Strukturen.*

Der Beweis von Theorem 2.20 erfolgt in zwei Teilen:

Im ersten Teil zeigen wir, dass jeder ESO-Satz Ψ bei Eingabe einer Struktur $\mathcal{A}$ nichtdeterministisch in Zeit polynomiell in der Größe von $\mathcal{A}$ ausgewertet werden kann (dies ist der „leichte" Teil des Beweises).

Theorem 2.20 (Der Satz von Fagin, 1974)

ESO *beschreibt* NP *auf der Klasse FIN aller endlicher Strukturen.*

Der Beweis von Theorem 2.20 erfolgt in zwei Teilen:

Im ersten Teil zeigen wir, dass jeder ESO-Satz Ψ bei Eingabe einer Struktur $\mathcal{A}$ nichtdeterministisch in Zeit polynomiell in der Größe von $\mathcal{A}$ ausgewertet werden kann (dies ist der „leichte" Teil des Beweises).

Im zweiten Teil zeigen wir, dass jedes Problem, das in NP liegt, durch einen ESO-Satz beschrieben werden kann.

Theorem 2.20 (Der Satz von Fagin, 1974)

ESO *beschreibt* NP *auf der Klasse FIN aller endlicher Strukturen.*

Der Beweis von Theorem 2.20 erfolgt in zwei Teilen:

Im ersten Teil zeigen wir, dass jeder ESO-Satz Ψ bei Eingabe einer Struktur $\mathcal{A}$ nichtdeterministisch in Zeit polynomiell in der Größe von $\mathcal{A}$ ausgewertet werden kann (dies ist der „leichte“ Teil des Beweises).

Im zweiten Teil zeigen wir, dass jedes Problem, das in NP liegt, durch einen ESO-Satz beschrieben werden kann.

Das Berechnungsmodell, mit dem wir arbeiten, sind Turingmaschinen. Um die Details des Beweises von Theorem 2.20 ausarbeiten zu können, müssen wir festlegen, wie eine Struktur $\mathcal{A}$ als Eingabe einer Turingmaschine repräsentiert wird. Wir benutzen dazu die sogenannte Standardkodierung, die im Folgenden eingeführt wird.

Definition 2.21

Sei $A = \{a_0, a_1, \ldots, a_{n-1}\}$ eine durch die Relation $<$ linear geordnete endliche Menge, so dass gilt: $a_0 < a_1 < \cdots < a_{n-1}$.

(a) Der Rang $\mathrm{rg}_<(a)$ eines Elements $a \in A$ ist

$$\mathrm{rg}_<(a) \ := \ |\{b \in A \ : \ b < a\}|.$$

Somit gilt $\mathrm{rg}_<(a_i) = i$ (f.a. $i \in \{0, \ldots, n{-}1\}$).

Definition 2.21

Sei $A = \{a_0, a_1, \ldots, a_{n-1}\}$ eine durch die Relation $<$ linear geordnete endliche Menge, so dass gilt: $a_0 < a_1 < \cdots < a_{n-1}$.

(a) Der Rang $\mathrm{rg}_<(a)$ eines Elements $a \in A$ ist

$$\mathrm{rg}_<(a) \ := \ |\{b \in A \ : \ b < a\}|.$$

Somit gilt $\mathrm{rg}_<(a_i) = i$ (f.a. $i \in \{0, \ldots, n-1\}$).

(b) Sei $r \in \mathbb{N}_{\geqslant 1}$. Die lexikographische Ordnung $<_{\text{lex}}$ auf A^r ist wie folgt definiert:

Für $\overline{b} = (b_1, \ldots, b_r) \in A^r$ und $\overline{c} = (c_1, \ldots, c_r) \in A^r$ ist

$$\overline{b} <_{\text{lex}} \overline{c} \iff \text{ex. } i \in \{1, \ldots, r\} \text{ s.d. } b_i < c_i \text{ und f.a. } j < i \text{ ist } b_j = c_j.$$

Definition 2.21

Sei $A = \{a_0, a_1, \ldots, a_{n-1}\}$ eine durch die Relation $<$ linear geordnete endliche Menge, so dass gilt: $a_0 < a_1 < \cdots < a_{n-1}$.

(a) Der Rang $\text{rg}_<(a)$ eines Elements $a \in A$ ist

$$\text{rg}_<(a) \ := \ |\{b \in A \ : \ b < a\}|.$$

Somit gilt $\text{rg}_<(a_i) = i$ (f.a. $i \in \{0, \ldots, n-1\}$).

(b) Sei $r \in \mathbb{N}_{\geqslant 1}$. Die lexikographische Ordnung $<_{\text{lex}}$ auf A^r ist wie folgt definiert:

Für $\overline{b} = (b_1, \ldots, b_r) \in A^r$ und $\overline{c} = (c_1, \ldots, c_r) \in A^r$ ist

$$\overline{b} <_{\text{lex}} \overline{c} \iff \text{ex. } i \in \{1, \ldots, r\} \text{ s.d. } b_i < c_i \text{ und f.a. } j < i \text{ ist } b_j = c_j.$$

Man kann sich leicht davon überzeugen, dass $<_{\text{lex}}$ eine strikte lineare Ordnung auf A^r ist .

Beispiel 2.22

Sei $A = \{0, 1, 2\}$ mit $0 < 1 < 2$. Bezüglich der lexikographischen Ordnung $<_{\text{lex}}$ auf A^2 gilt f.a. $\overline{b} \in A^2$:

Beispiel 2.22

Sei $A = \{0, 1, 2\}$ mit $0 < 1 < 2$. Bezüglich der lexikographischen Ordnung $<_{\text{lex}}$ auf A^2 gilt f.a. $\overline{b} \in A^2$:

$\overline{b}$	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
$\text{rg}_{<_{\text{lex}}}(\overline{b})$	0	1	2	3	4	5	6	7	8

Beispiel 2.22

Sei $A = \{0, 1, 2\}$ mit $0 < 1 < 2$. Bezüglich der lexikographischen Ordnung $<_{\text{lex}}$ auf A^2 gilt f.a. $\overline{b} \in A^2$:

$\overline{b}$	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
$\text{rg}_{<_{\text{lex}}}(\overline{b})$	0	1	2	3	4	5	6	7	8

Allgemein gilt: Ist $n \geqslant 1$, $A := \{0, \ldots, n{-}1\}$ mit $0 < \cdots < n{-}1$ und $r \geqslant 1$, so gilt für jedes $\overline{b} = (b_{r-1}, \ldots, b_0) \in A^r$

Beispiel 2.22

Sei $A = \{0, 1, 2\}$ mit $0 < 1 < 2$. Bezüglich der lexikographischen Ordnung $<_{\text{lex}}$ auf A^2 gilt f.a. $\overline{b} \in A^2$:

$\overline{b}$	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
$\mathrm{rg}_{<_{\text{lex}}}(\overline{b})$	0	1	2	3	4	5	6	7	8

Allgemein gilt: Ist $n \geqslant 1$, $A := \{0, \ldots, n{-}1\}$ mit $0 < \cdots < n{-}1$ und $r \geqslant 1$, so gilt für jedes $\overline{b} = (b_{r-1}, \ldots, b_0) \in A^r$

$$\mathrm{rg}_{<_{\text{lex}}}(\overline{b}) \;=\; \sum_{i=0}^{r-1} b_i n^i \;=\; b_{r-1}n^{r-1} + b_{r-2}n^{r-2} + \cdots + b_0 n^0,$$

Beispiel 2.22

Sei $A = \{0, 1, 2\}$ mit $0 < 1 < 2$. Bezüglich der lexikographischen Ordnung $<_{\text{lex}}$ auf A^2 gilt f.a. $\overline{b} \in A^2$:

$\overline{b}$	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
$\text{rg}_{<_{\text{lex}}}(\overline{b})$	0	1	2	3	4	5	6	7	8

Allgemein gilt: Ist $n \geqslant 1$, $A := \{0, \ldots, n-1\}$ mit $0 < \cdots < n-1$ und $r \geqslant 1$, so gilt für jedes $\overline{b} = (b_{r-1}, \ldots, b_0) \in A^r$

$$\text{rg}_{<_{\text{lex}}}(\overline{b}) \;=\; \sum_{i=0}^{r-1} b_i n^i \;=\; b_{r-1}n^{r-1} + b_{r-2}n^{r-2} + \cdots + b_0 n^0,$$

d.h. für $n \geqslant 2$ ist $\text{rg}_{<_{\text{lex}}}(\overline{b})$ die von der Darstellung $b_{r-1} \cdots b_0$ zur Basis n repräsentierte natürliche Zahl.

Definition 2.23 (Standardkodierung $\mathrm{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur,

Definition 2.23 (Standardkodierung $\mathrm{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \mathrm{ar}(R_j)$.

Definition 2.23 (Standardkodierung $\text{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \text{ar}(R_j)$.

  Sei $r := \max\{1, r_1, \ldots, r_\ell\}$

Definition 2.23 (Standardkodierung $\text{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \text{ar}(R_j)$.

  Sei $r := \max\{1, r_1, \ldots, r_\ell\}$ — d.h. wenn σ mindestens ein Relationssymbol enthält, ist r einfach die maximale Stelligkeit der Relationssymbole in σ, und ansonsten ist $r = 1$.

Definition 2.23 (Standardkodierung $\text{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \text{ar}(R_j)$.

  Sei $r := \max\{1, r_1, \ldots, r_\ell\}$ — d.h. wenn σ mindestens ein Relationssymbol enthält, ist r einfach die maximale Stelligkeit der Relationssymbole in σ, und ansonsten ist $r = 1$.

- Sei $\mathcal{A}$ eine endliche σ-Struktur, sei $n := |A|$.

Definition 2.23 (Standardkodierung $\text{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \text{ar}(R_j)$.

  Sei $r := \max\{1, r_1, \ldots, r_\ell\}$ — d.h. wenn σ mindestens ein Relationssymbol enthält, ist r einfach die maximale Stelligkeit der Relationssymbole in σ, und ansonsten ist $r = 1$.

- Sei $\mathcal{A}$ eine endliche σ-Struktur, sei $n := |A|$. Sei $<$ eine beliebige lineare Ordnung auf A und sei $A = \{a_0, \ldots, a_{n-1}\}$ mit $a_0 < a_1 < \cdots < a_{n-1}$.

Definition 2.23 (Standardkodierung $\text{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \text{ar}(R_j)$.

  Sei $r := \max\{1, r_1, \ldots, r_\ell\}$ — d.h. wenn σ mindestens ein Relationssymbol enthält, ist r einfach die maximale Stelligkeit der Relationssymbole in σ, und ansonsten ist $r = 1$.

- Sei $\mathcal{A}$ eine endliche σ-Struktur, sei $n := |A|$. Sei $<$ eine beliebige lineare Ordnung auf A und sei $A = \{a_0, \ldots, a_{n-1}\}$ mit $a_0 < a_1 < \cdots < a_{n-1}$.

  Im Folgenden identifizieren wir jedes $a_i \in A$ mit seinem Rang $i = \text{rg}_<(a_i)$

Definition 2.23 (Standardkodierung $\text{enc}_<(\mathcal{A})$)

- Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ (mit $\ell, \ell' \geqslant 0$) eine endliche funktionenfreie Signatur, wobei für jedes $j \in \{1, \ldots, \ell\}$ gilt: R_j ist ein Relationssymbol der Stelligkeit $r_j := \text{ar}(R_j)$.

  Sei $r := \max\{1, r_1, \ldots, r_\ell\}$ — d.h. wenn σ mindestens ein Relationssymbol enthält, ist r einfach die maximale Stelligkeit der Relationssymbole in σ, und ansonsten ist $r = 1$.

- Sei $\mathcal{A}$ eine endliche σ-Struktur, sei $n := |A|$. Sei $<$ eine beliebige lineare Ordnung auf A und sei $A = \{a_0, \ldots, a_{n-1}\}$ mit $a_0 < a_1 < \cdots < a_{n-1}$.

  Im Folgenden identifizieren wir jedes $a_i \in A$ mit seinem Rang $i = \text{rg}_<(a_i)$, und für $j \in \{1, \ldots, \ell\}$ identifizieren wir jedes r_j-Tupel $\overline{b} \in A^{r_j}$ mit seinem Rang $\text{rg}_{<_{\text{lex}}}(\overline{b}) \in \{0, \ldots, n^{r_j} - 1\}$.

- Für jedes $j \in \{1, \ldots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(R_j^{\mathcal{A}}) \quad := \quad w_0\, w_1 \cdots w_{n^r - 1} \quad \in \quad \{0,1\}^{(n^r)},$$

- Für jedes $j \in \{1, \ldots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(R_j^{\mathcal{A}}) \quad := \quad w_0\, w_1 \cdots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $\overline{b} \in A^{r_j}$ gilt: $\quad \overline{b} \in R_j^{\mathcal{A}} \iff w_{\mathrm{rg}_{<_{\mathrm{lex}}}(\overline{b})} = 1$
  - f.a. $m \geqslant n^{r_j}$ gilt: $\quad w_m = 0$.

- Für jedes $j \in \{1, \ldots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\text{enc}_<(R_j^{\mathcal{A}}) \quad := \quad w_0\, w_1 \cdots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $\overline{b} \in A^{r_j}$ gilt: $\quad \overline{b} \in R_j^{\mathcal{A}} \iff w_{\text{rg}_{<_{\text{lex}}}(\overline{b})} = 1$
  - f.a. $m \geqslant n^{r_j}$ gilt: $\quad w_m = 0.$
- Für jedes $j \in \{1, \ldots, \ell'\}$ kodieren wir die Konstante $c_j^{\mathcal{A}}$ durch das Wort

$$\text{enc}_<(c_j^{\mathcal{A}}) \quad := \quad w_0 w_1 \ldots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

- Für jedes $j \in \{1, \ldots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(R_j^{\mathcal{A}}) \quad := \quad w_0\, w_1 \cdots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $\overline{b} \in A^{r_j}$ gilt: $\quad \overline{b} \in R_j^{\mathcal{A}} \iff w_{\mathrm{rg}_{<_{\mathrm{lex}}}(\overline{b})} = 1$
  - f.a. $m \geqslant n^{r_j}$ gilt: $\quad w_m = 0$.
- Für jedes $j \in \{1, \ldots, \ell'\}$ kodieren wir die Konstante $c_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(c_j^{\mathcal{A}}) \quad := \quad w_0 w_1 \ldots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $m \in \{0, \ldots, n^r-1\}$ gilt: $\quad w_m = 1 \iff m = \mathrm{rg}_<(c_j^{\mathcal{A}})$.

- Für jedes $j \in \{1, \dots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\text{enc}_<(R_j^{\mathcal{A}}) \quad := \quad w_0\, w_1 \cdots w_{n^r - 1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $\overline{b} \in A^{r_j}$ gilt: $\overline{b} \in R_j^{\mathcal{A}} \iff w_{\text{rg}_{<_{\text{lex}}}(\overline{b})} = 1$
  - f.a. $m \geqslant n^{r_j}$ gilt: $w_m = 0$.
- Für jedes $j \in \{1, \dots, \ell'\}$ kodieren wir die Konstante $c_j^{\mathcal{A}}$ durch das Wort

$$\text{enc}_<(c_j^{\mathcal{A}}) \quad := \quad w_0 w_1 \dots w_{n^r - 1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $m \in \{0, \dots, n^r - 1\}$ gilt: $w_m = 1 \iff m = \text{rg}_<(c_j^{\mathcal{A}})$.
- Die Mächtigkeit n des Universums von $\mathcal{A}$ kodieren wir durch das Wort

$$\text{enc}_<(|A|) \quad := \quad 1^n 0^{n^r - n} \quad \in \quad \{0,1\}^{(n^r)}.$$

- Für jedes $j \in \{1, \ldots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(R_j^{\mathcal{A}}) \quad := \quad w_0\, w_1 \cdots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $\overline{b} \in A^{r_j}$ gilt: $\quad \overline{b} \in R_j^{\mathcal{A}} \iff w_{\mathrm{rg}_{<_{\mathrm{lex}}}(\overline{b})} = 1$
  - f.a. $m \geqslant n^{r_j}$ gilt: $\quad w_m = 0$.
- Für jedes $j \in \{1, \ldots, \ell'\}$ kodieren wir die Konstante $c_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(c_j^{\mathcal{A}}) \quad := \quad w_0 w_1 \ldots w_{n^r-1} \quad \in \quad \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $m \in \{0, \ldots, n^r - 1\}$ gilt: $\quad w_m = 1 \iff m = \mathrm{rg}_<(c_j^{\mathcal{A}})$.
- Die Mächtigkeit n des Universums von $\mathcal{A}$ kodieren wir durch das Wort

$$\mathrm{enc}_<(|A|) \quad := \quad 1^n 0^{n^r - n} \quad \in \quad \{0,1\}^{(n^r)}.$$

- Die Standardkodierung von $\mathcal{A}$ bzgl. $<$ ist das Wort

$$\mathrm{enc}_<(\mathcal{A}) \;:=\; \mathrm{enc}_<(|A|)\,\mathrm{enc}_<(R_1^{\mathcal{A}}) \cdots \mathrm{enc}_<(R_\ell^{\mathcal{A}})\,\mathrm{enc}_<(c_1^{\mathcal{A}}) \cdots \mathrm{enc}_<(c_{\ell'}^{\mathcal{A}}).$$

- Für jedes $j \in \{1, \dots, \ell\}$ kodieren wir die Relation $R_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(R_j^{\mathcal{A}}) \;:=\; w_0\, w_1 \cdots w_{n^r-1} \;\in\; \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $\overline{b} \in A^{r_j}$ gilt: $\overline{b} \in R_j^{\mathcal{A}} \iff w_{\mathrm{rg}_{<_{\mathrm{lex}}}(\overline{b})} = 1$
  - f.a. $m \geqslant n^{r_j}$ gilt: $w_m = 0$.
- Für jedes $j \in \{1, \dots, \ell'\}$ kodieren wir die Konstante $c_j^{\mathcal{A}}$ durch das Wort

$$\mathrm{enc}_<(c_j^{\mathcal{A}}) \;:=\; w_0 w_1 \dots w_{n^r-1} \;\in\; \{0,1\}^{(n^r)},$$

  wobei
  - f.a. $m \in \{0, \dots, n^r-1\}$ gilt: $w_m = 1 \iff m = \mathrm{rg}_<(c_j^{\mathcal{A}})$.
- Die Mächtigkeit n des Universums von $\mathcal{A}$ kodieren wir durch das Wort

$$\mathrm{enc}_<(|A|) \;:=\; 1^n 0^{n^r-n} \;\in\; \{0,1\}^{(n^r)}.$$

- Die Standardkodierung von $\mathcal{A}$ bzgl. $<$ ist das Wort

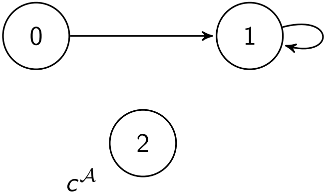
$$\mathrm{enc}_<(\mathcal{A}) \;:=\; \mathrm{enc}_<(|A|)\,\mathrm{enc}_<(R_1^{\mathcal{A}}) \cdots \mathrm{enc}_<(R_\ell^{\mathcal{A}})\,\mathrm{enc}_<(c_1^{\mathcal{A}}) \cdots \mathrm{enc}_<(c_{\ell'}^{\mathcal{A}}).$$

  Es gilt: $\mathrm{enc}_<(\mathcal{A}) \in \{0,1\}^{(1+\ell+\ell')\cdot n^r}$.

Beispiel 2.24

Sei $\sigma = \{E, c\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E und einem Konstantensymbol c besteht.

Wir betrachten die σ-Struktur $\mathcal{A}$:

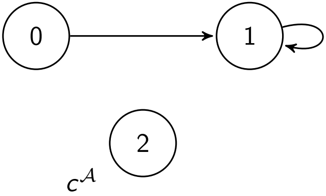


d.h. $\mathcal{A} = (A, E^{\mathcal{A}}, c^{\mathcal{A}})$ mit $A = \{0, 1, 2\}$, $E^{\mathcal{A}} = \{(0, 1), (1, 1)\}$ und $c^{\mathcal{A}} = 2$.

Beispiel 2.24

Sei $\sigma = \{E, c\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E und einem Konstantensymbol c besteht.

Wir betrachten die σ-Struktur $\mathcal{A}$:



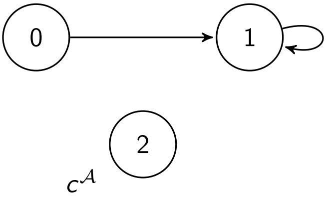
d.h. $\mathcal{A} = (A, E^{\mathcal{A}}, c^{\mathcal{A}})$ mit $A = \{0, 1, 2\}$, $E^{\mathcal{A}} = \{(0, 1), (1, 1)\}$ und $c^{\mathcal{A}} = 2$.

Es gilt $r = 2$, $n = 3$ und somit $n^r = 9$.

Beispiel 2.24

Sei $\sigma = \{E, c\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E und einem Konstantensymbol c besteht.

Wir betrachten die σ-Struktur $\mathcal{A}$:



d.h. $\mathcal{A} = (A, E^{\mathcal{A}}, c^{\mathcal{A}})$ mit $A = \{0,1,2\}$, $E^{\mathcal{A}} = \{(0,1), (1,1)\}$ und $c^{\mathcal{A}} = 2$.

Es gilt $r = 2$, $n = 3$ und somit $n^r = 9$.

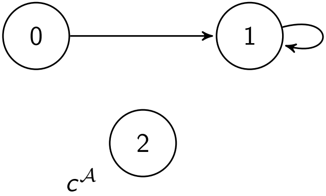
Sei $<$ die natürliche lineare Ordnung auf A. Dann gilt

$$\mathrm{enc}_<(\mathcal{A}) \quad =$$

Beispiel 2.24

Sei $\sigma = \{E, c\}$ die Signatur, die aus einem 2-stelligen Relationssymbol E und einem Konstantensymbol c besteht.

Wir betrachten die σ-Struktur $\mathcal{A}$:



d.h. $\mathcal{A} = (A, E^{\mathcal{A}}, c^{\mathcal{A}})$ mit $A = \{0, 1, 2\}$, $E^{\mathcal{A}} = \{(0, 1), (1, 1)\}$ und $c^{\mathcal{A}} = 2$.

Es gilt $r = 2$, $n = 3$ und somit $n^r = 9$.

Sei $<$ die natürliche lineare Ordnung auf A. Dann gilt

$$\text{enc}_<(\mathcal{A}) \quad = \quad \underbrace{111|000000}_{\text{enc}_<(|A|)}\ \underbrace{010|010|000}_{\text{enc}_<(E^{\mathcal{A}})}\ \underbrace{001|000000}_{\text{enc}_<(c^{\mathcal{A}})}.$$

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\text{enc}_<(\mathcal{A})| \quad =$$

Bemerkung 2.25

(a) Die Standardkodierung $\mathrm{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\mathrm{enc}_<(\mathcal{A})| \quad = \quad (1+\ell+\ell') \cdot |A|^r.$$

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\text{enc}_<(\mathcal{A})| \quad = \quad (1 + \ell + \ell') \cdot |A|^r.$$

Insbes. ist für jede feste Signatur σ also die Länge von $\text{enc}_<(\mathcal{A})$ polynomiell in der Mächtigkeit $|A|$ der Struktur $\mathcal{A}$.

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\text{enc}_<(\mathcal{A})| \quad = \quad (1 + \ell + \ell') \cdot |A|^r.$$

Insbes. ist für jede feste Signatur σ also die Länge von $\text{enc}_<(\mathcal{A})$ polynomiell in der Mächtigkeit $|A|$ der Struktur $\mathcal{A}$.

(c) Sei

$$\text{enc}_<(\mathcal{A}) \quad = \quad w_0 w_1 \ldots w_{(1+\ell+\ell')\cdot n^r - 1}$$

mit $n := |A|$ und $w_i \in \{0, 1\}$ f.a. $i < n^r$.

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\text{enc}_<(\mathcal{A})| \quad = \quad (1 + \ell + \ell') \cdot |A|^r.$$

Insbes. ist für jede feste Signatur σ also die Länge von $\text{enc}_<(\mathcal{A})$ polynomiell in der Mächtigkeit $|A|$ der Struktur $\mathcal{A}$.

(c) Sei

$$\text{enc}_<(\mathcal{A}) \quad = \quad w_0 w_1 \ldots w_{(1+\ell+\ell')\cdot n^r - 1}$$

mit $n := |A|$ und $w_i \in \{0, 1\}$ f.a. $i < n^r$.

Für $j \in \{1, \ldots, \ell\}$ und ein Tupel $\overline{b} \in A^{r_j}$ lässt sich die Information, ob $\overline{b} \in R_j^{\mathcal{A}}$ ist, in $\text{enc}_<(\mathcal{A})$ am Buchstaben w_m ablesen, für

$$m \quad :=$$

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\text{enc}_<(\mathcal{A})| \quad = \quad (1+\ell+\ell') \cdot |A|^r.$$

Insbes. ist für jede feste Signatur σ also die Länge von $\text{enc}_<(\mathcal{A})$ polynomiell in der Mächtigkeit $|A|$ der Struktur $\mathcal{A}$.

(c) Sei

$$\text{enc}_<(\mathcal{A}) \quad = \quad w_0 w_1 \ldots w_{(1+\ell+\ell')\cdot n^r - 1}$$

mit $n := |A|$ und $w_i \in \{0,1\}$ f.a. $i < n^r$.

Für $j \in \{1,\ldots,\ell\}$ und ein Tupel $\overline{b} \in A^{r_j}$ lässt sich die Information, ob $\overline{b} \in R_j^{\mathcal{A}}$ ist, in $\text{enc}_<(\mathcal{A})$ am Buchstaben w_m ablesen, für

$$m \quad := \quad j \cdot n^r \ + \ \text{rg}_{<_{\text{lex}}}(\overline{b}).$$

Bemerkung 2.25

(a) Die Standardkodierung $\text{enc}_<(\mathcal{A})$ hängt von der gewählten linearen Ordnung $<$ ab.

(b) Es gilt:

$$|\text{enc}_<(\mathcal{A})| \quad = \quad (1+\ell+\ell') \cdot |A|^r.$$

Insbes. ist für jede feste Signatur σ also die Länge von $\text{enc}_<(\mathcal{A})$ polynomiell in der Mächtigkeit $|A|$ der Struktur $\mathcal{A}$.

(c) Sei

$$\text{enc}_<(\mathcal{A}) \quad = \quad w_0 w_1 \dots w_{(1+\ell+\ell')\cdot n^r - 1}$$

mit $n := |A|$ und $w_i \in \{0,1\}$ f.a. $i < n^r$.

Für $j \in \{1,\dots,\ell\}$ und ein Tupel $\overline{b} \in A^{r_j}$ lässt sich die Information, ob $\overline{b} \in R_j^{\mathcal{A}}$ ist, in $\text{enc}_<(\mathcal{A})$ am Buchstaben w_m ablesen, für

$$m \quad := \quad j \cdot n^r \ + \ \text{rg}_{<_{\text{lex}}}(\overline{b}).$$

Wir können nun den Beweis des Satzes von Fagin (Theorem 2.20) führen.

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d\, \varphi$ ein ESO[σ]-Satz. Wir müssen zeigen, dass das folgende Problem zu NP gehört:

$\mathit{EVAL}_\Psi(\mathit{FIN})$: Auswertungsproblem für Ψ auf FIN

Eingabe: Eine endliche σ-Struktur $\mathcal{A}$

Frage: Gilt $\mathcal{A} \models \Psi$?

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d \, \varphi$ ein ESO[σ]-Satz. Wir müssen zeigen, dass das folgende Problem zu NP gehört:

$EVAL_\Psi(FIN)$: Auswertungsproblem für Ψ auf FIN
Eingabe: Eine endliche σ-Struktur $\mathcal{A}$
Frage: Gilt $\mathcal{A} \models \Psi$?

Ein nichtdeterministischer Polynomialzeit-Algorithmus kann bei Eingabe einer endlichen σ-Struktur $\mathcal{A}$ wie folgt vorgehen, um zu testen, ob $\mathcal{A} \models \Psi$:

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d \, \varphi$ ein ESO[σ]-Satz. Wir müssen zeigen, dass das folgende Problem zu NP gehört:

$\mathrm{EVAL}_\Psi(FIN)$: Auswertungsproblem für Ψ auf FIN
Eingabe: Eine endliche σ-Struktur $\mathcal{A}$
Frage: Gilt $\mathcal{A} \models \Psi$?

Ein nichtdeterministischer Polynomialzeit-Algorithmus kann bei Eingabe einer endlichen σ-Struktur $\mathcal{A}$ wie folgt vorgehen, um zu testen, ob $\mathcal{A} \models \Psi$:

(1) Wenn $d > 0$: Rate Belegungen der Relationsvariablen $X_1, \ldots, X_d$, d.h. rate Relationen $S_1 \subseteq A^{\mathrm{ar}(X_1)}, \ldots, S_d \subseteq A^{\mathrm{ar}(X_d)}$.

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d \, \varphi$ ein ESO[σ]-Satz.
Wir müssen zeigen, dass das folgende Problem zu NP gehört:

$\mathit{EVAL}_\Psi(\mathit{FIN})$: Auswertungsproblem für Ψ auf FIN

Eingabe: Eine endliche σ-Struktur $\mathcal{A}$

Frage: Gilt $\mathcal{A} \models \Psi$?

Ein nichtdeterministischer Polynomialzeit-Algorithmus kann bei Eingabe einer endlichen σ-Struktur $\mathcal{A}$ wie folgt vorgehen, um zu testen, ob $\mathcal{A} \models \Psi$:

(1) Wenn $d > 0$: Rate Belegungen der Relationsvariablen $X_1, \ldots, X_d$, d.h. rate Relationen $S_1 \subseteq A^{\mathrm{ar}(X_1)}, \ldots, S_d \subseteq A^{\mathrm{ar}(X_d)}$.

Das geht in Zeit

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d\, \varphi$ ein ESO[σ]-Satz.
Wir müssen zeigen, dass das folgende Problem zu NP gehört:

> *$\mathrm{EVAL}_\Psi(\mathit{FIN})$: Auswertungsproblem für Ψ auf FIN*
>
> *Eingabe:* Eine endliche σ-Struktur $\mathcal{A}$
>
> *Frage:* Gilt $\mathcal{A} \models \Psi$?

Ein nichtdeterministischer Polynomialzeit-Algorithmus kann bei Eingabe einer endlichen σ-Struktur $\mathcal{A}$ wie folgt vorgehen, um zu testen, ob $\mathcal{A} \models \Psi$:

(1) Wenn $d > 0$: Rate Belegungen der Relationsvariablen $X_1, \ldots, X_d$, d.h. rate Relationen $S_1 \subseteq A^{\mathrm{ar}(X_1)}, \ldots, S_d \subseteq A^{\mathrm{ar}(X_d)}$.

Das geht in Zeit $\mathcal{O}(d \cdot n^r)$, wobei $n = |A|$ und r die maximale Stelligkeit der Relationsvariablen ist.

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d \, \varphi$ ein ESO[σ]-Satz. Wir müssen zeigen, dass das folgende Problem zu NP gehört:

$EVAL_\Psi(FIN)$: Auswertungsproblem für Ψ auf FIN
Eingabe: Eine endliche σ-Struktur $\mathcal{A}$
Frage: Gilt $\mathcal{A} \models \Psi$?

Ein nichtdeterministischer Polynomialzeit-Algorithmus kann bei Eingabe einer endlichen σ-Struktur $\mathcal{A}$ wie folgt vorgehen, um zu testen, ob $\mathcal{A} \models \Psi$:

(1) Wenn $d > 0$: Rate Belegungen der Relationsvariablen $X_1, \ldots, X_d$, d.h. rate Relationen $S_1 \subseteq A^{\mathrm{ar}(X_1)}, \ldots, S_d \subseteq A^{\mathrm{ar}(X_d)}$.

Das geht in Zeit $\mathcal{O}(d \cdot n^r)$, wobei $n = |A|$ und r die maximale Stelligkeit der Relationsvariablen ist.

(2) Teste (deterministisch, in Polynomialzeit), ob $(\mathcal{A}, S_1, \ldots, S_d) \models \varphi$.

Unser Ziel ist es zu zeigen, dass ESO die Klasse NP auf *FIN* beschreibt.

Wir führen den Beweis in zwei Schritten.

1. Schritt: Sei σ eine Signatur und sei $\Psi := \exists X_1 \cdots \exists X_d \, \varphi$ ein ESO[σ]-Satz. Wir müssen zeigen, dass das folgende Problem zu NP gehört:

$\mathit{EVAL}_\Psi(\mathit{FIN})$: Auswertungsproblem für Ψ auf FIN
Eingabe: Eine endliche σ-Struktur $\mathcal{A}$
Frage: Gilt $\mathcal{A} \models \Psi$?

Ein nichtdeterministischer Polynomialzeit-Algorithmus kann bei Eingabe einer endlichen σ-Struktur $\mathcal{A}$ wie folgt vorgehen, um zu testen, ob $\mathcal{A} \models \Psi$:

(1) Wenn $d > 0$: Rate Belegungen der Relationsvariablen $X_1, \ldots, X_d$, d.h. rate Relationen $S_1 \subseteq A^{\mathrm{ar}(X_1)}, \ldots, S_d \subseteq A^{\mathrm{ar}(X_d)}$.

Das geht in Zeit $\mathcal{O}(d \cdot n^r)$, wobei $n = |A|$ und r die maximale Stelligkeit der Relationsvariablen ist.

(2) Teste (deterministisch, in Polynomialzeit), ob $(\mathcal{A}, S_1, \ldots, S_d) \models \varphi$.

Dafür können wir den naiven Algorithmus benutzen, der rekursiv entlang der Definition der Semantik von FO vorgeht; siehe Vorlesung „Logik in der Informatik“.

2. Schritt: Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ eine Signatur und sei $\mathcal{C}$ eine unter Isomorphie abgeschlossene Klasse endlicher σ-Strukturen, sodass das Problem

> *Zugehörigkeit zu $\mathcal{C}$ in FIN*
>
> *Eingabe:* Eine endliche Struktur $\mathcal{A}$.
>
> *Frage:* Ist $\mathcal{A} \in \mathcal{C}$?

in NP liegt.

2. Schritt: Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ eine Signatur und sei $\mathcal{C}$ eine unter Isomorphie abgeschlossene Klasse endlicher σ-Strukturen, sodass das Problem

Zugehörigkeit zu $\mathcal{C}$ in FIN

Eingabe: Eine endliche Struktur $\mathcal{A}$.

Frage: Ist $\mathcal{A} \in \mathcal{C}$?

in NP liegt.

Das heißt, es gibt eine NTM

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

mit $F = F_{\text{akz}} \,\dot\cup\, F_{\text{verw}}$ und eine Konstante $k \in \mathbb{N}$, s.d. M bei Eingabe (der Kodierung) einer endlichen σ-Struktur $\mathcal{A}$ entscheidet, ob $\mathcal{A} \in \mathcal{C}$ ist und dabei weniger als n^k Schritte macht, für $n := |A|$.

2. Schritt: Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ eine Signatur und sei $\mathcal{C}$ eine unter Isomorphie abgeschlossene Klasse endlicher σ-Strukturen, sodass das Problem

Zugehörigkeit zu $\mathcal{C}$ in FIN
Eingabe: Eine endliche Struktur $\mathcal{A}$.
Frage: Ist $\mathcal{A} \in \mathcal{C}$?

in NP liegt.

Das heißt, es gibt eine NTM

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

mit $F = F_{\text{akz}} \dot\cup F_{\text{verw}}$ und eine Konstante $k \in \mathbb{N}$, s.d. M bei Eingabe (der Kodierung) einer endlichen σ-Struktur $\mathcal{A}$ entscheidet, ob $\mathcal{A} \in \mathcal{C}$ ist und dabei weniger als n^k Schritte macht, für $n := |A|$.

D.h., für jede endliche σ-Struktur $\mathcal{A}$ und jede lineare Ordnung $<$ auf A gilt:

- Jeder Lauf von M bei Eingabe $\text{enc}_<(\mathcal{A})$ endet nach weniger als n^k Schritten,

2. Schritt: Sei $\sigma = \{R_1, \ldots, R_\ell, c_1, \ldots, c_{\ell'}\}$ eine Signatur und sei $\mathcal{C}$ eine unter Isomorphie abgeschlossene Klasse endlicher σ-Strukturen, sodass das Problem

Zugehörigkeit zu $\mathcal{C}$ in FIN
Eingabe: Eine endliche Struktur $\mathcal{A}$.
Frage: Ist $\mathcal{A} \in \mathcal{C}$?

in NP liegt.

Das heißt, es gibt eine NTM

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

mit $F = F_{\text{akz}} \,\dot{\cup}\, F_{\text{verw}}$ und eine Konstante $k \in \mathbb{N}$, s.d. M bei Eingabe (der Kodierung) einer endlichen σ-Struktur $\mathcal{A}$ entscheidet, ob $\mathcal{A} \in \mathcal{C}$ ist und dabei weniger als n^k Schritte macht, für $n := |A|$.

D.h., für jede endliche σ-Struktur $\mathcal{A}$ und jede lineare Ordnung $<$ auf A gilt:

- Jeder Lauf von M bei Eingabe $\text{enc}_<(\mathcal{A})$ endet nach weniger als n^k Schritten, und
- $\mathcal{A} \in \mathcal{C} \iff$
  es gibt einen akzeptierenden Lauf von M bei Eingabe $\text{enc}_<(\mathcal{A})$.