

Automaten und Logik (SS 06)

Sibylle Schwarz

Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg

email: schwarz@s@informatik.uni-halle.de

url: <http://nirvana.informatik.uni-halle.de/~schwarz/>

14. Juli 2006

Semester

Beginn: 01.04.2006

Ende: 30.09.2006

Vorlesungszeit Beginn: 03.04.2006

Vorlesungszeit Ende: 14.07.2006

Unterbrechungen: alle gesetzlichen Feiertage (Ostern, Himmelfahrt, Pfingsten)

15 Wochen (3+1) = 20 V + 10 Ü

Literatur

- [1] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: <http://www.grappa.univ-lille3.fr/tata>, 1997. release October, 1rst 2002.
- [2] Dominique Perrin and Jean Éric Pin. *Infinite Words*, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004. ISBN 0-12-532111-2.
- [3] H. Straubing. *Finite automata, formal logic, and circuit complexity*. Boston, Basel, Berlin: Birkhäuser, 1994.
- [4] W. Thomas. Languages, automata and logic. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words. Springer, Berlin, 1997.
- [5] Wolfgang Thomas. Applied automata theory, 2003.
- [6] Igor Walukiewicz. Automata and logic, 2002.

Inhaltsverzeichnis

1	Motivation	4
2	Sprachen (Wiederholung)	5
3	Logiken $MSO_A[S, <]$ und $FO_A[S, <]$	6
3.1	Syntax $FO_A[S, <]$ und $MSO_A[S, <]$	6
3.2	Semantik $FO_A[S, <]$ und $MSO_A[S, <]$	6
3.3	Wahrheitswerte	7
3.4	$MSO_A[S, <]$ - und $FO_A[S, <]$ -definierbare Sprachen	8
3.5	Kodierung der Belegung	8
4	Sprachen endlicher Wörter	9
4.1	Wiederholung NFA	9
4.2	NFA-Erkennbarkeit und $MSO_A[S]$ -Definierbarkeit	11
4.3	Folgerungen	13
5	ω-Sprachen	14
5.1	ω -Automaten	15
5.2	Büchi-Automaten	15
5.3	Muller-Automaten	16
5.4	Satz von McNaughton	17
5.5	Satz von Büchi	19
5.6	$MSO[S]$ -definierbare Prädikate auf \mathbb{N}	20
5.7	Logik S1S	20
5.8	Weitere Logiken über \mathbb{N}	21
6	Baumsprachen	22
6.1	Endliche Terme und Bäume	22
6.2	Bottom-Up-Baumautomaten	24
6.3	Top-Down-Baumautomaten	25
6.4	Eigenschaften von REC_{NFTA}	26
6.5	Logik $MSO_{\Gamma}[S_1, \dots, S_k]$	29
6.6	Logik WSkS	30
6.7	Zusammenhänge zwischen Baum- und Wortsprachen	31
7	Sprachen unendlicher Bäume	32
7.1	ω -Baumautomaten	32
7.2	Logik über unendlichen Bäumen	34
7.3	Entscheidbarkeitskriterien für Logiken	35

8	Modallogik	37
8.1	Syntax	37
8.2	Kripke-Strukturen	37
8.3	Modellrelation	37
8.4	Einbettung in FO	38
8.5	Bisimulation	38
8.6	Baummodelleingenschaft von ML	39
9	Sprachen endlicher Bilder	40
9.1	Tiling-Systeme	40
9.2	Logik über Bildsprachen	41

1 Motivation

[4, 3, 6, 5, 2, 1] Beschreibung von Systemen durch Automaten und logische Formeln möglich

Logik statisch
 geeignet zur Spezifikation
 Negation einfach

Automat dynamisch
 geeignet zur Auswertung von Systemverhalten
 Minimierung

typische Fragen zu

	Sprachen	Automaten	Logik
spezielle	Sprache L $L \stackrel{?}{=} \emptyset, L \stackrel{?}{=} A^*$ $L_1 \stackrel{?}{=} L_2, L_1 \cap L_2 \stackrel{?}{=} \emptyset$	Automat \mathcal{A} erkannte Sprache $L(\mathcal{A})$ Äquivalenz von \mathcal{A}, \mathcal{B} minimaler äq. Automat	Formel φ Semantik $\llbracket \varphi \rrbracket$ $\varphi \equiv \psi$ „minimale“ äquivalente Formel Allgemeingültigkeit Erfüllbarkeit
		äq. <i>deterministischer</i> Automat	äq. Formel in Normalform
Klasse von	Sprachen Abgeschlossenheit	Automaten erkannte Sprachen	Formeln definierbare Sprachen Entscheidbarkeit

Wichtiger Satz aus Informatik 4

Satz 1.1 Für eine Sprache $L \subseteq A^*$ sind die folgenden Aussagen äquivalent:

1. L ist regulär.
2. Es existiert ein NFA \mathcal{A} mit $L(\mathcal{A}) = L$.
3. Es existiert ein DFA \mathcal{A} mit $L(\mathcal{A}) = L$.
4. Die Nerode-Rechtskongruenz \sim_L hat endlichen Index.

2 Sprachen (Wiederholung)

Wörter

endliches Alphabet A – Menge von Symbolen (Buchstaben)

kartesisches Produkt A^n – n -Tupel, Vektoren, Wörter der Länge n

Länge $|w| = n$

Wort $w_1 \dots w_{|w|}$ ist Funktion $w : \{1, \dots, |w|\} \rightarrow A$ (statt w_i auch $w(i)$)

$\text{dom}(w) = \{1, \dots, |w|\}$ Menge aller Positionen in w

Menge aller Wörter $A^* = \bigcup_{n \in \mathbb{N}} A^n$ mit $A^0 = \{\varepsilon\}$

Verkettung \cdot von Wörtern $u, v \in A^* : u : \{1, \dots, |u|\}, v : \{1, \dots, |v|\}$

$(u \cdot v) : \{1, \dots, |u| + |v|\}$ mit

$$(u \cdot v)(i) = \begin{cases} u(i) & \text{für } i \in \{1, \dots, |u|\} \\ v(i - |u|) & \text{sonst} \end{cases}$$

Verkettung \cdot ist assoziativ, aber nicht kommutativ

ε neutral für \cdot , d.h. für alle $w \in A^* : \varepsilon \cdot w = w \cdot \varepsilon = w$

$(A^*, \cdot, \varepsilon)$ ist Monoid

Sprachen

Sprachen $L \subseteq A^*$, $L \in 2^{A^*}$, charakteristische Funktion $\chi_L : A^* \rightarrow \{0, 1\}$

Boolesche Operationen $\cup, \cap, \bar{}$

$(2^{A^*}, \cup, \cap, \bar{}, \emptyset, A^*)$ ist Boolesche Algebra

Verkettung $L_1 \cdot L_2 = \{u \cdot v \mid u \in L_1, v \in L_2\}$

$L^0 = \{\varepsilon\}$, $L^{n+1} = L \cdot L^n$, $L^* = \bigcup_{n \in \mathbb{N}} L^n$

$(2^{A^*}, \cdot, \{\varepsilon\})$ ist Monoid

$(2^{A^*}, \cup, \cdot, \{\varepsilon\}, A^*)$ ist Halbring

Reguläre Ausdrücke:

REGEXP

$$r ::= \emptyset \mid \varepsilon \mid a \mid r \cup s \mid r \cdot s \mid r^*$$

wobei $a \in A$ und r, s REGEXP

reguläre Ausdrücke (als Terme) – rationale Sprache (RAT)

erweiterte reguläre Ausdrücke (nicht stärker, erlauben aber intuitivere Formalisierung)

3 Logiken $\text{MSO}_A[S, <]$ und $\text{FO}_A[S, <]$

Logik = Syntax + Semantik

FO – first order: nur Individuenvariablen

MSO – monadic second order: Individuen- und Mengenvariablen (Menge = einstelliges Prädikat)

	Junktoren	$\vee, \wedge, \neg, \rightarrow, \leftrightarrow$
syntaktische Elemente einer logischen Sprache	Quantoren	\forall, \exists
	Variablen	$\mathbb{X} = \mathbb{X}_{\text{FO}} \cup \mathbb{X}_{\text{MSO}}$
	allgemein	für Wörter
Relationssymbole	$\Sigma_R = \bigcup_{n \in \mathbb{N}} \Sigma_R^{(n)}$	$S, <$ (zweistellig), P_a für $a \in A$ (einstellig)
Konstanten	$\Sigma_F^{(0)}$	l, r (einstellig)

3.1 Syntax $\text{FO}_A[S, <]$ und $\text{MSO}_A[S, <]$

	allgemein	für Wörter
Terme	$\mathbb{X} \cup \Sigma_F^{(0)}$	$\mathbb{X} \cup \{l, r\}$
FO-Atome	$p(t_1, \dots, t_n)$ mit $p \in \Sigma_R^{(n)}, t, t_i$ Terme	$t_1 < t_2, S(t_1, t_2), P_a(t)$ mit t, t_1, t_2 Terme,
MSO-Atome	$X(t)$ mit $X \in \mathbb{X}_{\text{MSO}}$	$X(t)$ mit Term $t, X \in \mathbb{X}_{\text{MSO}}$

Formeln in $\text{FO}_A[S, <]$ induktiv definiert: $\varphi ::= P \mid \neg\varphi \mid \varphi * \psi \mid Qx\varphi$
mit FO-Atomen $P, x \in \mathbb{X}_{\text{FO}}$, $\text{MSO}_A[S, <]$ -Formeln φ, ψ ,
 $*$ $\in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\forall, \exists\}$,

Formeln in $\text{MSO}_A[S, <]$ induktiv definiert: $\varphi ::= P \mid \neg\varphi \mid \varphi * \psi \mid Qx\varphi$
mit FO- oder MSO-Atomen $P, x \in \mathbb{X}$, $\text{FO}_A[S, <]$ -Formeln φ, ψ ,
 $*$ $\in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$, $Q \in \{\forall, \exists\}$,

3.2 Semantik $\text{FO}_A[S, <]$ und $\text{MSO}_A[S, <]$

Σ -Struktur $S = (|S|, [\cdot]_S)$ mit

- Träger $|S| \neq \emptyset$
- für jedes Symbol $n \in \mathbb{N}$, jedes $p \in \Sigma_R^{(n)}$ eine Relation $[p]_S \subseteq |S|^n$ (als Funktion $[p]_S : |S|^n \rightarrow \{0, 1\}$)
- für jede Konstante $c \in \Sigma_F^{(0)}$ ein $[c]_S \in |S|$

Jedes Wort $w \in A^*$ definiert *Wortstruktur* $\underline{w} = (|\underline{w}|, [\cdot]_{\underline{w}})$ mit

- $|\underline{w}| = \text{dom}(w) = \{1, \dots, |w|\}$

•

$$\begin{aligned}
[P_a]_{\underline{w}} &= \{i \in \text{dom}(w) \mid w_i = a\} \\
[S]_{\underline{w}} &= \{(i, i+1) \mid i \in \text{dom}(w) \setminus \{|w|\}\} \\
[<]_{\underline{w}} &= \{(i, j) \in \text{dom}(w)^2 \mid i < j\} \\
[l]_{\underline{w}} &= 0 \\
[r]_{\underline{w}} &= |w|
\end{aligned}$$

Variablenbelegung $\sigma : \mathbb{X}_{\text{FO}} \longrightarrow \text{dom}(w), \mathbb{X}_{\text{MSO}} \longrightarrow 2^{\text{dom}(w)}$

Modifizierte Belegung

$$\sigma[x \mapsto c](y) = \begin{cases} c & \text{falls } y = x \\ \sigma(y) & \text{sonst} \end{cases}$$

Erweiterung auf Terme:

$$\sigma(t) = \begin{cases} \sigma(t) & \text{falls } t \in \mathbb{X}_{\text{FO}} \\ [c]_{\underline{w}} & \text{falls } c \in \Sigma_F^{(0)} \end{cases}$$

Interpretation (\underline{w}, σ) mit Wortstruktur \underline{w} und Belegung σ .

3.3 Wahrheitswerte

Wahrheitswert von *Atomen* in Interpretation (\underline{w}, σ)

$$\begin{aligned}
\llbracket P_a(t) \rrbracket_{(\underline{w}, \sigma)} &= \begin{cases} 1 & \text{falls } w_{\sigma(t)} = a \\ 0 & \text{sonst} \end{cases} \\
\llbracket t_1 < t_2 \rrbracket_{(\underline{w}, \sigma)} &= \begin{cases} 1 & \text{falls } \sigma(t_1) < \sigma(t_2) \\ 0 & \text{sonst} \end{cases} \\
\llbracket S(t_1, t_2) \rrbracket_{(\underline{w}, \sigma)} &= \begin{cases} 1 & \text{falls } \sigma(t_2) = \sigma(t_1) + 1 \\ 0 & \text{sonst} \end{cases} \\
\llbracket X(t) \rrbracket_{(\underline{w}, \sigma)} &= \begin{cases} 1 & \text{falls } \sigma(t) \in \sigma(X) \\ 0 & \text{sonst} \end{cases}
\end{aligned}$$

Wahrheitswert von *Formeln* in Interpretation $((\underline{w}, \sigma), \sigma)$

$$\begin{aligned}
\llbracket \neg \varphi \rrbracket_{(\underline{w}, \sigma)} &= 1 - \llbracket \varphi \rrbracket_{(\underline{w}, \sigma)} \\
\llbracket \varphi \vee \psi \rrbracket_{(\underline{w}, \sigma)} &= \max\{\llbracket \varphi \rrbracket_{(\underline{w}, \sigma)}, \llbracket \psi \rrbracket_{(\underline{w}, \sigma)}\} \\
\llbracket \varphi \wedge \psi \rrbracket_{(\underline{w}, \sigma)} &= \min\{\llbracket \varphi \rrbracket_{(\underline{w}, \sigma)}, \llbracket \psi \rrbracket_{(\underline{w}, \sigma)}\} \\
\llbracket \exists x \varphi \rrbracket_{(\underline{w}, \sigma)} &= \max\{\llbracket \varphi \rrbracket_{(\underline{w}, \sigma[x \mapsto c])} \mid c \in \text{dom}(w)\} \\
\llbracket \exists X \varphi \rrbracket_{(\underline{w}, \sigma)} &= \max\{\llbracket \varphi \rrbracket_{(\underline{w}, \sigma[X \mapsto C])} \mid C \subseteq \text{dom}(w)\} \\
\llbracket \forall x \varphi \rrbracket_{(\underline{w}, \sigma)} &= \min\{\llbracket \varphi \rrbracket_{(\underline{w}, \sigma[x \mapsto c])} \mid c \in \text{dom}(w)\} \\
\llbracket \forall X \varphi \rrbracket_{(\underline{w}, \sigma)} &= \min\{\llbracket \varphi \rrbracket_{(\underline{w}, \sigma[X \mapsto C])} \mid C \subseteq \text{dom}(w)\}
\end{aligned}$$

3.4 $\text{MSO}_A[S, <]$ - und $\text{FO}_A[S, <]$ -definierbare Sprachen

Interpretation (\underline{w}, σ) heißt *Modell* für φ gdw. $\llbracket \varphi \rrbracket_{(\underline{w}, \sigma)} = 1$

Für Sätze $\varphi \in \text{MSO}_A[S, <]$ (Formeln ohne freie Variablen) ist die Belegung irrelevant und wir schreiben $\llbracket \varphi \rrbracket_{\underline{w}} := \llbracket \varphi \rrbracket_{(\underline{w}, \sigma)}$ für eine beliebige Belegung σ .

Jeder Satz $\varphi \in \text{MSO}_A[S, <]$ definiert die Sprache

$$L(\varphi) = \{\underline{w} \mid \llbracket \varphi \rrbracket_{\underline{w}} = 1\}$$

Definition 3.1 Für eine Logik L heißt die Sprache $L \subseteq A^*$ genau dann L -definierbar, wenn ein Satz $\varphi \in L$ mit $L = L(\varphi)$ existiert.

Fakt 3.1 1. Zu jeder Formel in $\text{FO}_A[S, <]$ existiert eine äquivalente Formel in $\text{FO}_A[S, <]$, die die Konstanten l und r nicht enthält.

2. Zu jeder Formel in $\text{MSO}_A[S, <]$ existiert eine äquivalente Formel in $\text{MSO}_A[S, <]$, die die Konstanten l und r nicht enthält.

$$\varphi(l) \equiv \exists x (\varphi(x) \wedge \neg \exists y (y < x)) \quad \varphi(r) \equiv \exists x (\varphi(x) \wedge \neg \exists y (x < y))$$

Fakt 3.2 1. Zu jeder Formel in $\text{FO}_A[S, <]$ existiert eine äquivalente Formel in $\text{FO}_A[<]$.

2. Zu jeder Formel in $\text{MSO}_A[S, <]$ existiert eine äquivalente Formel in $\text{MSO}_A[<]$.

Beweis:

$$S(x, y) = x < y \wedge \neg \exists z (x < z \wedge z < y)$$

Fakt 3.3 Zu jeder Formel in $\text{MSO}_A[S, <]$ existiert eine äquivalente Formel in $\text{MSO}_A[S]$.

Beweis:

$$x < y \equiv \exists X (\neg X(x) \wedge X(y) \wedge \forall z \forall z' ((X(z) \wedge S(z, z')) \rightarrow X(z')))$$

Bemerkung 3.1 In $\text{FO}_A[S, <]$ läßt sich $<$ nicht eliminieren.

3.5 Kodierung der Belegung

Wort	$w : \text{dom}(w) \longrightarrow A$
Belegung für $X \in \mathbb{X}_{\text{MSO}}$	$\sigma(X) : \text{dom}(w) \longrightarrow \{0, 1\}$
Belegung für $x \in \mathbb{X}_{\text{FO}}$	$\sigma(x) : \text{dom}(w) \longrightarrow \{0, 1\}$ mit $ \sigma(x) = 1$
Interpretation	$(\underline{w}, \sigma) : \text{dom}(w) \longrightarrow A \times \{0, 1\}^n$

für n Variablen

Interpretation = Wort über erweitertem (aber endlichen) Alphabet $A \times \{0, 1\}^n$

$$(\underline{w}, \sigma) \in (A \times \{0, 1\}^n)^*$$

4 Sprachen endlicher Wörter

4.1 Wiederholung NFA

NFA $\mathcal{A} = (Q, \alpha, \delta, \beta)$ mit

- endliche Zustandsmenge $Q \neq \emptyset$
- Startzustände $\alpha \subseteq Q$ ($\alpha : Q \rightarrow \{0, 1\}$)
- Endzustände $\beta \subseteq Q$ ($\beta : Q \rightarrow \{0, 1\}$)
- Transitionen: Homomorphismus $\delta : A^* \rightarrow (Q^2 \rightarrow \{0, 1\})$ von Wörtern aus A^* auf $|Q|^2$ -Matrizen mit Einträgen in $\{0, 1\}$, eindeutig bestimmt durch Einschränkung auf Buchstaben $\delta : A \rightarrow (Q^2 \rightarrow \{0, 1\})$

Homomorphismus δ zwischen Monoiden $(A^*, \cdot, \varepsilon)$ und $(Q^2, \odot, E_{|Q|})$:

$E_{|Q|}$ ist $|Q| \times |Q|$ -Einheitsmatrix,

· ist Verkettung, \odot boolesche Matrixmultiplikation

$$\begin{aligned} \delta(\varepsilon) &= E_{|Q|} \quad \text{und} \quad \delta(u \cdot v) = \delta(u) \odot \delta(v) \quad \text{mit} \\ (\delta(u) \odot \delta(v))(i, j) &= \max\{\delta(u)(i, k)\delta(v)(k, j) \mid k \in Q\} \end{aligned}$$

Erkannte Sprache

elegant

$$L(\mathcal{A}) = \{w \mid \alpha \odot \delta(w) \odot \beta = 1\}$$

durch akzeptierenden Lauf

Lauf des NFA \mathcal{A} auf Wort $w \in A^*$: $(q_1, \dots, q_{|w|+1}) \in Q^{|w|+1}$, so daß für alle $i \in \text{dom}(w)$ gilt $(q_i, q_{i+1}) \in \delta(w_i)$.

akzeptierender Lauf des NFA \mathcal{A} auf Wort $w \in A^*$: Lauf $(q_0, \dots, q_{|w|}) \in Q^{|w|+1}$ mit $q_0 \in \alpha$ und $q_{|w|} \in \beta$.

$$L(\mathcal{A}) = \{w \mid \text{es ex. ein akzeptierender Lauf von } \mathcal{A} \text{ auf } w\}$$

durch Überdeckung

Transitionsfunktion δ definiert endliche Menge von *Kacheln* (Dominosteine)

$$T_{\mathcal{A}} = \{(a, p, q) \mid \delta(a)(p, q) = 1\} \quad (1)$$

Projektionen: $\pi_1(a, p, q) = a \in A$, $\pi_2(a, p, q) = p \in Q$, $\pi_3(a, p, q) = q \in Q$

Überdeckung für w : Folge $(T_1, \dots, T_n) \in T_{\mathcal{A}}^{|w|}$ von Kacheln, so daß für alle $i \in \text{dom}(w)$ gilt $\pi_1(T_i) = w_i$ und für alle $i \in \text{dom}(w) \setminus \{|w|\}$ gilt $\pi_3(T_i) = \pi_2(T_{i+1})$.

akzeptierende Überdeckung für w : $(T_1, \dots, T_n) \in T_{\mathcal{A}}^{|w|}$ mit $\pi_2(T_1) \in \alpha$ und $\pi_3(T_{|w|}) \in \beta$

$L(\mathcal{A}) = \{w \mid \text{es ex. eine akzeptierende Überdeckung von } w \text{ mit Kacheln aus } T(\mathcal{A})\}$

4.2 NFA-Erkennbarkeit und $\text{MSO}_A[S]$ -Definierbarkeit

Satz 4.1 (Büchi, Elgot 1960) Eine Sprache $L \subseteq A^*$ ist genau dann regulär, wenn sie $\text{MSO}_A[S]$ -definierbar ist.

verwenden Satz aus Informatik 4:

L regulär gdw. es existiert NFA $\mathcal{A} = (Q, \alpha, \delta, \beta)$ mit $L = L(\mathcal{A})$

Beweis: (\implies)

zu zeigen: Jede reguläre Sprache ist $\text{MSO}_A[S]$ -definierbar.

gegeben: NFA $\mathcal{A} = (Q, \alpha, \delta, \beta)$

Konstruktion eines Satzes $\varphi_{\mathcal{A}}$ mit $L(\mathcal{A}) = L(\varphi_{\mathcal{A}})$

Bezeichnung $\text{dom}'(w) = \text{dom}(w) \cup \{|w| + 1\} = \{1, \dots, |w| + 1\}$

Lauf $q : \text{dom}'(w) \longrightarrow Q$

$\varphi_{\mathcal{A}}$ ist logische Formulierung für s existiert ein akzeptierender Lauf $(q_i)_{i \in \text{dom}'(w)}$ von \mathcal{A} auf w .

- MSO -Variablen $\{X_q \mid q \in Q\}$ für Mengen von Positionen, an denen \mathcal{A} bei Lauf auf w in Zustand q ist.
- $\{X_q \mid q \in Q\}$ muß *Partition* von $\text{dom}'(w)$ sein.

$$\varphi_p = \forall x \left(\bigvee_{q \in Q} \left(X_q(x) \wedge \bigwedge_{p \in Q \setminus \{q\}} \neg X_q(x) \right) \right)$$

- q_1 ist Startzustand ($q_1 \in \alpha$),

$$\varphi_{\alpha} = \bigvee_{q \in \alpha} X_q(1)$$

- für alle $i \in \text{dom}(w)$ existiert ein w_i -Übergang von q_i zu q_{i+1} ($\delta(w_i)(q_i, q_{i+1}) = 1$).

$$\varphi_{\delta} = \forall x \forall y \left(\neg S(x, y) \vee \bigvee_{a \in A, (p, q) \in \delta(a)} (X_p(x) \wedge X_q(y) \wedge P_a(x)) \right)$$

- von $q_{|w|}$ wird ein Endzustand erreicht

$$\varphi_{\beta} = \bigvee_{\substack{a \in A, q \in \beta \\ (p, q) \in \delta(a)}} (X_p(r) \wedge P_a(r))$$

$$\varphi_{\mathcal{A}} = \exists X_1 \dots \exists X_{|Q|} (\varphi_P \wedge \varphi_{\alpha} \wedge \varphi_{\delta} \wedge \varphi_{\beta})$$

(\longleftarrow)

zu zeigen: Jede $\text{MSO}_A[S]$ -definierbare Sprache ist regulär.

gegeben: Satz $\varphi \in \text{MSO}_A[S]$

Konstruktion eines NFA $\mathcal{A} = (Q, \alpha, \delta, \beta)$ mit $L(\mathcal{A}) = L(\varphi)$

Definition einer Logik $\text{MSO}'_A[S, <]$ ohne first-order-Variablen

Atome:

$$\begin{aligned}
 X \subseteq Y & \quad \llbracket X \subseteq Y \rrbracket_{(w, \sigma)} = \begin{cases} 1 & \text{falls } \sigma(X) \subseteq \sigma(Y) \\ 0 & \text{sonst} \end{cases} \\
 X \subseteq P_a & \quad \llbracket X \subseteq P_a \rrbracket_{(w, \sigma)} = \begin{cases} 1 & \text{falls } \sigma(X) \subseteq [P_a]_w \\ 0 & \text{sonst} \end{cases} \\
 \text{Sing}(X) & \quad \llbracket \text{Sing}(X) \rrbracket_{(w, \sigma)} = \begin{cases} 1 & \text{falls } |\sigma(X)| = 1 \\ 0 & \text{sonst} \end{cases} \\
 S(X, Y) & \quad \llbracket S(X, Y) \rrbracket_{(w, \sigma)} = \begin{cases} 1 & \text{falls } \sigma(X) = \{i\}, \sigma(Y) = \{j\} \text{ und } j = i + 1 \\ 0 & \text{sonst} \end{cases} \\
 X < Y & \quad \llbracket X < Y \rrbracket_{(w, \sigma)} = \begin{cases} 1 & \text{falls } \sigma(X) = \{i\}, \sigma(Y) = \{j\} \text{ und } i < j \\ 0 & \text{sonst} \end{cases}
 \end{aligned}$$

induktiver Formalaufbau wie üblich

Lemma 4.1 Zu jeder Formel $\varphi \in \text{MSO}_A[S, <]$ existiert eine äquivalente Formel $\varphi' \in \text{MSO}'_A[S, <]$.

Beweis: Übersetzung $t : \text{MSO}_A[S, <] \longrightarrow \text{MSO}'_A[S, <]$ mit $t' : \mathbb{X}_{\text{FO}} \longrightarrow \mathbb{X}_{\text{MSO}}$

$$\begin{aligned}
 t(X(y)) & = \text{Sing}(t'(y)) \wedge t'(y) \subseteq X \\
 t(P_a(x)) & = \text{Sing}(t'(x)) \wedge t'(x) \subseteq P_a \\
 t(S(x, y)) & = S(t'(x), t'(y)) \\
 t(x < y) & = t'(x) < t'(y) \\
 t(\neg\varphi) & = \neg t(\varphi) \\
 t(\varphi * \psi) & = t(\varphi) * t(\psi) \quad \text{für } * \in \{\vee, \wedge\} \\
 t(Qx\varphi) & = Qt'(x)t(\varphi) \quad \text{für } Q \in \{\forall, \exists\}
 \end{aligned}$$

Fortsetzung Beweis für Satz 4.1 (\Leftarrow):

Codierung der Belegungen nach Abschnitt 3.5

zu zeigen: Für jede Formel $\varphi \in \text{MSO}'_A[S, <]$ mit den freien Variablen $\{X_1, \dots, X_n\}$ ist die Sprache $L(\varphi) \subseteq A \times \{0, 1\}^n$ regulär

Induktion über Formelaufbau

Induktionsanfang: Konstruktion von Automaten für Atome

Induktionsschritt: für mit Junktoren zusammengesetzte Formeln:

Abschlußeigenschaften von REG unter
Vereinigung, Schnitt und Komplement

im Induktionsschritt bleibt noch zu zeigen:

aus $L(\varphi) \in \text{REG}$ folgt $L(\exists X\varphi) \in \text{REG}$ und $L(\forall X\varphi) \in \text{REG}$

Wegen der logischen Äquivalenz $\forall X\varphi \equiv \neg\exists X\neg\varphi$ genügt es, zu zeigen:

aus $L(\varphi) \in \text{REG}$ folgt $L(\exists X\varphi) \in \text{REG}$

Lemma 4.2 (Projektionslemma) Für zwei Alphabete A und B , eine Funktion $h : A \rightarrow B$ und deren homomorpher Fortsetzung $h : A^* \rightarrow B^*$ gilt:
Ist $L \subseteq A^*$ regulär, dann ist auch $h(L) = \{h(w) \mid w \in L\}$ regulär.

REG ist also abgeschlossen unter Projektionen.

Projektion

$$f : (A \times \{0, 1\}^{n+1})^+ \rightarrow (A \times \{0, 1\}^n)^+ \quad \text{mit} \quad (2)$$

$$f(a, c_1, \dots, c_n, c_{n+1}) = (a, c_1, \dots, c_n)$$

Lemma 4.3 (Quantorenlemma) Für die von einer Formel φ definierte Sprache $L(\varphi)$ und die Projektion f aus (2) gilt

$$f(L(\varphi)) = L(\exists X_n\varphi)$$

Abschluß des Beweises von Satz 4.1:

für Induktionsschritt noch zu zeigen:

aus $L(\varphi) \in \text{REG}$ folgt $L(\exists X\varphi) \in \text{REG}$

o.B.d.A $X = X_n$

Wegen Lemma 4.3 gilt $L(\exists X_n\varphi) = f(L(\varphi))$ mit der Projektion f aus (2)

und wegen Lemma 4.2 folgt $L(\exists X_n\varphi) = f(L(\varphi)) \in \text{REG}$ aus $L(\varphi) \in \text{REG}$.

4.3 Folgerungen

(ungerichteter) Graph $G = (V, E)$ mit $E \subseteq \binom{V}{2}$

Hamiltonkreis in G : Folge (v_1, \dots, v_n) in der

1. jeder Knoten aus V genau einmal vorkommt und
2. $\{v_1, v_n\} \in E$ und für alle $i \in \{1, \dots, n-1\}$: $\{v_i, v_{i+1}\} \in E$

Folgerung 4.1 Es gibt keinen Satz $\varphi \in \text{MSO}[E]$, so daß für jeden Graphen G gilt:
 $\llbracket \varphi \rrbracket_G = 1$ gdw. G hat einen Hamiltonkreis

Folgerung 4.2 Es existiert keine Formel $\varphi \in \text{MSO}[<]$ mit den freien Variablen $\{x, y, z\}$, so daß $\llbracket \varphi \rrbracket_\sigma = 1$ gdw. $\sigma(x) + \sigma(y) = \sigma(z)$

5 ω -Sprachen

endliches Alphabet A

ω -Wörter

ω -Wort $\xi : \mathbb{N} \rightarrow A$ ($\xi : \omega \rightarrow A$)

Menge aller ω -Wörter: $A^\omega = \{\xi : \mathbb{N} \rightarrow A\}$

Verkettung $\cdot : A^* \times A^\omega \rightarrow A^\omega$: für $u : \{0, \dots, |u| - 1\} \rightarrow A, \xi : \mathbb{N} \rightarrow A$

$$(u\xi)(i) = \begin{cases} u(i) & \text{für } i \in \{0, |u| - 1\} \\ \xi(i - |u|) & \text{sonst} \end{cases}$$

Verkettung „assoziativ“ $(uv)\xi = u(v\xi)$

ω -Sprachen

ω -Sprache: $L \subseteq A^\omega$

Operationen auf ω -Sprachen:

- Boolesche Operationen $\cup, \cap, \overline{}$ (Komplement)
- Verkettung $\cdot : 2^{A^*} \times 2^{A^\omega} \rightarrow 2^{A^\omega}$ mit
für $L \subseteq A^*$ und $M \subseteq A^\omega$ ist $LM = \{u\xi \mid u \in L, \xi \in M\} \subseteq A^\omega$
- unendliche Iteration ${}^\omega : 2^{A^*} \rightarrow 2^{A^\omega}$ mit
 $L^\omega = \{u_1 u_2 \dots \mid i \in \mathbb{N}, u_i \in L \setminus \{\varepsilon\}\}$

$$\begin{aligned} (L \cup M)^\omega &= (L^* M)^\omega \cup (L \cup M)^\omega \\ (LM)^\omega &= L(ML)^\omega \\ L^\omega &= LL^\omega = L^n L^\omega = L^* L^\omega \\ (L^n)^\omega &= (L^+)^\omega = L^\omega \end{aligned}$$

Reguläre ω -Sprachen

Definition 5.1 Sprache $L \subseteq A^\omega$ heißt ω -regulär gdw. eine Zahl n und für jedes $i \in \{1, \dots, n\}$ Sprachen $N_i, M_i \in \text{REG}$ existieren, so daß gilt:

$$L = \bigcup_{i \in \{1, \dots, n\}} M_i N_i^\omega$$

Menge aller ω -regulären Sprachen: ωREG

Es gilt

- $\emptyset \in \omega\text{REG}$
- aus $L, M \in \omega\text{REG}$ folgt $L \cup M \in \omega\text{REG}$
- für alle $L \in \text{REG} \setminus \{\varepsilon\}$ gilt $L^\omega \in \omega\text{REG}$
- für alle $L \in \text{REG}, M \in \omega\text{REG}$ gilt $LM \in \omega\text{REG}$

5.1 ω -Automaten

$\mathcal{A} = (Q, \alpha, \delta, \text{Acc})$ mit $\text{Acc} \subseteq Q^\omega$ (Menge akzeptierender Läufe)

Akzeptanzbedingung Acc oft parametrisiert

\mathcal{A} akzeptiert $\xi \in A^\omega$ gdw. existiert Lauf $r \in Q^\omega$ von \mathcal{A} auf ξ , d.h.

- $r(0) \in \alpha$,
- für alle $i \in \mathbb{N}$ gilt $(q_i, q_{i+1}) \in \delta(w_i)$,
- $r \in \text{Acc}$.

ω -Automat \mathcal{A} erkennt Sprache $L(\mathcal{A}) =$ Menge aller akzeptierten Wörter $\xi \in A^\omega$

Definition 5.2 *Einen Sprache $L \subseteq A^\omega$ heißt von einem Automatentyp $*$ (z.B. NFA, DFA) erkennbar ($*$ -erkennbar) gdw. ein Automat \mathcal{A} vom Typ $*$ existiert, so daß $L = L(\mathcal{A})$ gilt.*

Die Menge aller $*$ -erkennbaren Sprachen heißt $\text{REC}_* \subseteq 2^{A^\omega}$.

Definition 5.3 *Für $r \in Q^\omega$ ist $\text{Inf}(r) = \{q \in Q \mid |\{i \mid r(i) = q\}| = \omega\}$.*

5.2 Büchi-Automaten

$\text{Acc} = \text{Büchi}(\beta)$ parametrisiert durch Menge $\beta \subseteq Q$

$$\text{Büchi}(\beta) = \{r \in Q^\omega \mid \text{Inf}(r) \cap \beta \neq \emptyset\}$$

nichtdeterministischer Büchi-Automat BNFA

$$\mathcal{A}_\beta = (Q, \alpha, \delta, \text{Büchi}(\beta))$$

zugeordneter NFA

$$\mathcal{A} = (Q, \alpha, \delta, \beta)$$

deterministischer Büchi-Automat BDFA:

alalog DFA: $|\alpha| = 1$ und δ ist Funktion (hier auch $\delta : A \rightarrow (Q \rightarrow Q)$)

Es gilt $\text{REC}_{\text{BDFA}} \subseteq \text{REC}_{\text{BNFA}}$

Beispiele:

- $L = (a^*b)^\omega \in \text{REC}_{\text{BDFA}}$
- $A^\omega \setminus L = A^*a^\omega \in \text{REC}_{\text{BNFA}}$

Fakt 5.1 *Es gilt $A^*a^\omega \notin \text{REC}_{\text{BDFA}}$*

Folgerung 5.1 $\text{REC}_{\text{BDFA}} \subset \text{REC}_{\text{BNFA}}$

Unterschied zu $\text{REC}_{\text{DFA}} = \text{REC}_{\text{NFA}}$ für *endliche* Wörter

Abschlußeigenschaften von REC_{BNFA}

- Aus $L, M \in \text{REC}_{\text{BNFA}}$ folgt $L \cup M \in \text{REC}_{\text{BNFA}}$.
- Aus $L, M \in \text{REC}_{\text{NFA}}$ folgt $LM^\omega \in \text{REC}_{\text{BNFA}}$.

Für jede Sprache $L = L(\mathcal{A})$ mit $\mathcal{A} = (Q, \alpha, \delta, \text{Büchi}(\beta))$ gilt

$$L = \bigcup_{\substack{i \in \alpha \\ f \in \beta}} L(\mathcal{A}_{i,f}) L(\mathcal{A}_{f,f})^\omega$$

mit den NFAs $\mathcal{A}_{p,q} = (Q, \{p\}, \delta, \text{Büchi}(\{q\}))$

Analogon zu Satz von Kleene (endliche Wörter) für ω -Wörter:

Satz 5.1 $\omega\text{REG} = \text{REC}_{\text{BNFA}}$

analog zu Projektionslemma

Lemma 5.1 Für zwei Alphabete A, B und $f : A \rightarrow B$ mit den homomorphen Fortsetzungen $f : A^* \rightarrow B^*$ und $f : A^\omega \rightarrow B^\omega$ gilt:
aus $L \subseteq A^\omega$ mit $L \in \omega\text{REG}$ folgt $f(L) \in \omega\text{REG}$.

Satz 5.2 Für jeden BNFA \mathcal{A} ist entscheidbar, ob $L(\mathcal{A}) = \emptyset$.

5.3 Muller-Automaten

Problem: $(a+b)^* a^\omega \notin \text{REC}_{\text{BNFA}}$

Akzeptanzbedingung $\text{Acc} = \text{Muller}(F)$ parametrisiert durch Menge $F \subseteq 2^Q$

$$\text{Muller}(F) = \{r \in Q^\omega \mid \text{Inf}(r) \in F\}$$

betrachten nur *deterministische* Muller-Automaten (MDFA): $\mathcal{A} = (Q, \{i\}, \delta, \text{Muller}(F))$

REC_{MDFA} = Menge aller durch einen Muller-Automaten erkennbaren Sprachen

Beispiele:

1. $(a^*b)^\omega \in \text{REC}_{\text{MDFA}}$
2. $A^\omega \setminus L = A^* a^\omega \in \text{REC}_{\text{MDFA}}$

Fakt 5.2 Für jede Sprache $L \in A^\omega$ gilt: $L \in \text{REC}_{\text{MDFA}}$ gdw. $A^\omega \setminus L \in \text{REC}_{\text{MDFA}}$

Lemma 5.2 $\text{REC}_{\text{MDFA}} \subseteq \omega\text{REG}$

5.4 Satz von McNaughton

Satz 5.3 (McNaughton 1968) $\omega\text{REG} \subseteq \text{REC}_{\text{MDFA}}$

Beweis: gezeigt wird $\text{REC}_{\text{BNFA}} \subseteq \text{REC}_{\text{MDFA}}$ durch Konstruktion eines MDFA $\mathcal{B} = (Q_B, s_0, \delta_B, \text{Muller}(T))$ zu einem gegebenen BNFA $\mathcal{A} = (Q, \{q_0\}, \delta, \text{Büchi}(\beta))$ (nach Safra)

Safra-Bäume

Baum s aus Knoten:

$v = (n_v, l_v, c_v, K_v) \in T = \mathbb{N} \times 2^Q \times \{w, g\} \times T^*$ mit

1. ID $n_v \in \mathbb{N}$
2. Markierung $l_v \in 2^Q$
3. Färbung $c_v \in \{w, g\}$
4. Liste der Kinder von v $K_v \in T^*$

mit den folgenden Eigenschaften

1. für alle v in s und alle $u, u' \in K_v$ mit $u \neq u'$ gilt $l_u \cap l_{u'} = \emptyset$
2. für alle v in s gilt $\bigcup_{u \in K_v} l_u \subset l_v$

Fakt 5.3 Die Menge aller Safra-Bäume über einer endlichen Menge Q ist endlich.

gegeben: $\mathcal{A} = (Q, \{q_0\}, \delta, \text{Büchi}(\beta))$

Konstruktion von $\mathcal{B} = (Q_B, s_0, \delta_B, \text{Muller}(T))$ mit

- $Q_B =$ Menge aller Safra-Bäume über Q

- $s_0 = (1, \{q_0\}, w, [])$

-

$$T = \left\{ M \subset Q_B \mid \exists i \in \mathbb{N} : \left(\begin{array}{l} \forall s \in M \exists v \in s : n_v = i \\ \wedge \exists s \in M \exists v \in s : (n_v = i \wedge c_v = g) \end{array} \right) \right\}$$

- $\delta' : A \longrightarrow (Q_B \longrightarrow Q_B)$ nach folgendem Verfahren

Für alle $a \in A$ und alle Safra-Bäume s wird der Safra-Baum $\delta'(a)(s) = s'$ bestimmt durch

Vorbereitung:

1. für alle v in s : $c_v := g$ (alle grünen Markierungen löschen)
2. für alle v in s : falls $l_v \cap \beta \neq \emptyset$ (l_v enthält akzeptierende Zustände):
 $i := \min\{n \in \mathbb{N} \mid n \notin \{n_{u'} \mid u' \in s\}$ (noch nicht vorkommende ID festlegen)
 $u := \{i, \{l_v \cap \beta\}, w, []\}$ (neuen Knoten erzeugen)
 $K_v := \text{append}(K_v, u)$ (neuen Knoten an Liste der Kinder von v anhängen)

Zustandsübergänge:

3. für alle v in s : $l_v := \{\delta(a)(q) \mid q \in l_v\}$

Aufräumen:

4. horizontal:
 - (a) für alle v in s : $L = K_v$
solange $L \neq []$
 - (1) $\text{head}(L) = u, \text{tail}(L) = L'$
 - (2) für alle $q \in u$ und $u' \in L'$ setze $l_{u'} := l_{u'} \setminus \{q\}$
 - (3) $L := L'$
(alte Zustände löschen, die schon in „älteren Bruder“ vorkommen)
 - (b) für alle v in s : lösche alle $u \in K_v$ mit $l_u = \emptyset$ aus K_v
(alte Knoten mit leerer Markierung löschen, Wurzel aber nicht)
5. vertikal: für alle v in s : falls $l_v = \bigcup_{u \in K_v} l_u$, setze
 - (a) $K_v = []$ (alle Kinder löschen)
 - (b) $c_v = g$ (v „blinkt“)

Bemerkung: Die Akzeptanzbedingung

$$\text{Muller}(T) = \{r \in Q_B^\omega \mid \text{Inf}(r) \in T\} \quad \text{mit}$$

$$T = \left\{ M \subset Q_B \mid \exists i \in \mathbb{N} \left(\begin{array}{l} \forall s \in M \exists v \in s : n_v = i \\ \wedge \exists s \in M \exists v \in s : (n_v = i \wedge c_v = g) \end{array} \right) \right\}$$

läßt sich auch beschreiben durch

$$\text{Rabin}(T') = \{r \in Q_B^\omega \mid \exists (F, G) \in T' : \text{Inf}(r) \cap F = \emptyset \wedge \text{Inf}(r) \cap G \neq \emptyset\}$$

mit

$$T' = \left\{ (F_i, G_i) \mid \exists i \in \mathbb{N} \left(\begin{array}{l} F_i = \{s \in Q_B \mid \forall v \in s : n_v \neq i\} \\ \wedge G_i = \{s \in Q_B \mid \exists v \in s (n_v = i \wedge c_v = g)\} \end{array} \right) \right\}$$

$\mathcal{A} = (Q, \alpha, \delta, \text{Rabin}(T'))$ mit $T' \subseteq (2^Q)^2$ heißt *Rabin-Automat*.

Für diese Konstruktion gilt $\text{Muller}(T) = \text{Rabin}(T')$ und damit

$$L(Q, \alpha, \delta, \text{Muller}(T)) = L(Q, \alpha, \delta, \text{Rabin}(T')).$$

5.5 Satz von Büchi

Abschlußeigenschaften von ωREG

Aus Satz 5.1 ($\omega\text{REG} = \text{REC}_{\text{BNFA}}$), Lemma 5.2 ($\text{REC}_{\text{MDFA}} \subseteq \omega\text{REG}$) und Satz 5.3 ($\omega\text{REG} \subseteq \text{REC}_{\text{MDFA}}$) folgt sofort:

Satz 5.4 $\omega\text{REG} = \text{REC}_{\text{BNFA}} = \text{REC}_{\text{MDFA}}$

Damit ist ωREG abgeschlossen unter

- endlicher Vereinigung
- Komplement (Fakt 5.2)
- endlichem Schnitt (deMorgan)
- Projektion (Lemma 5.1)

Analog zu Definition 3.1 für Sprachen endlicher Wörter:

Definition 5.4 Für eine Logik L heißt die ω -Sprache $L \subseteq A^\omega$ genau dann L -definierbar, wenn ein Satz $\varphi \in L$ mit $L = L(\varphi)$ existiert.

Uns interessieren hier $\text{MSO}_A[S, <]$ -definierbare ω -Sprachen.

Satz 5.5 (Büchi 1962) Für jede Sprache $L \subseteq A^\omega$ gilt $L \in \omega\text{REG}$ gdw. L ist $\text{MSO}_A[S, <]$ -definierbar.

Beweis:

(\implies) benutzen $\omega\text{REG} = \text{REC}_{\text{BNFA}}$,

Konstruktion eines Satzes $\varphi_{\mathcal{A}} \in \text{MSO}_A[S, <]$ zu gegebenem BNFA \mathcal{A}

Modifikation der Formel aus Beweis von Satz 4.1, $\varphi_{\text{Büchi}(\beta)}$ statt φ_β

$$\varphi_{\text{Büchi}(\beta)} = \bigvee_{q \in \beta} \varphi_{\text{Inf}}(X_p)$$

mit $\llbracket \varphi_{\text{Inf}}(X) \rrbracket_{(\xi, \sigma)} = 1$ gdw. $\sigma(X)$ unendlich

$$\varphi_{\text{Inf}}(X) = \forall x \exists y (x < y \wedge X(y)) \in \text{MSO}_A[S, <]$$

(\impliedby) Induktion über Formelaufbau (analog zum Beweis von Satz 4.1)

Induktionsschritt: Junktoren \neg, \vee, \exists durch Abschlußeigenschaften von ωREG unter Komplement, Vereinigung und Projektion.

Satz 5.6 Für ω -Sprachen $L, L_1, L_2 \in \omega\text{REG}$ ist entscheidbar, ob

1. $L = \emptyset$,
2. $L = A^\omega$,
3. $L_1 \subseteq L_2$,
4. $L_1 = L_2$.

5.6 MSO[S]-definierbare Prädikate auf \mathbb{N}

Für $A = \{a\}$ ist $A^\omega = \{a^\omega\}$ und es existieren genau zwei Sprachen $\subseteq A^\omega$, nämlich $L_0 = \emptyset$ und $L_1 = \{a^\omega\}$. Hier ist das Prädikatsymbol P_a irrelevant.

Jede Formel $\varphi \in \text{MSO}_\emptyset[S]$ mit den freien Variablen $\{x_1, \dots, x_n\} \subseteq \mathbb{X}_{\text{FO}}$ definiert eine n -stellige Relation $p_\varphi \subseteq \mathbb{N}^n$ durch

$$(m_1, \dots, m_n) \in p_\varphi \quad \text{gdw.} \quad \llbracket \varphi \rrbracket_{(a^\omega, [x_1 \mapsto m_1, \dots, x_n \mapsto m_n])} = 1$$

Notation $\text{MSO}[\Sigma]$ statt $\text{MSO}_\emptyset[\Sigma]$

Definition 5.5 Eine Relation p heißt genau dann MSO[S]-definierbar, wenn eine Formel $\varphi \in \text{MSO}[S]$ existiert, so daß $p = p_\varphi$ gilt.

Jede Funktion $f : M^n \rightarrow M$ läßt sich als Relation $p_f \subseteq M^{n+1}$ auffassen

$$(x_1, \dots, x_n, y) \in p_f \quad \text{gdw.} \quad f(x_1, \dots, x_n) = y$$

Definition 5.5 bestimmt damit auch die Menge aller MSO[S]-definierbaren Funktionen.

Jeder Satz $\varphi \in \text{MSO}[S]$ definiert eine (durch das Prädikat S) definierbare Eigenschaft der Menge \mathbb{N} .

5.7 Logik S1S

Logik S1S (monadic second order logic with one successor):

$$\text{S1S} = \text{Th}(\text{MSO}[<], \mathbb{N}) = \{\varphi \in \text{MSO}[<] \mid \mathbb{N} \models \varphi\}$$

(Menge aller über \mathbb{N} wahren Sätze $\varphi \in \text{MSO}[S]$)

Satz 5.7 (Büchi 1962) S1S ist entscheidbar.

Beweis: Entscheidungsverfahren für φ :

Konstruktion des Büchi-Automaten \mathcal{A}_φ wie im Beweis von Satz 5.5 über Alphabet $A = \{a\}$.

Es gilt $\mathbb{N} \models \varphi$ gdw. $L(\varphi) = L(\mathcal{A}_\varphi) \neq \emptyset$.

$L(\varphi) \neq \emptyset$ gilt genau dann, wenn in \mathcal{A}_φ ein Zustand $q \in \beta$ existiert, so daß

1. ein Pfad von einem Zustand aus α zu q und
2. ein Pfad von q nach q existiert.

Diese Eigenschaft von \mathcal{A}_φ ist entscheidbar.

5.8 Weitere Logiken über \mathbb{N}

$\text{FO}_A[<]$ und $\text{FO}[<]$

Fakt 5.4 $\text{Th}(\text{FO}[<], \mathbb{N})$ ist entscheidbar.

Beweis: Folgerung aus Satz 5.7 wegen $\text{FO}[<] \subseteq \text{MSO}[S]$

$\text{FO}_A[S]$ und $\text{FO}[S]$

Wegen Fakt 3.2 gilt:

Fakt 5.5 Jede $\text{FO}_A[S]$ -definierbare Sprache ist $\text{FO}_A[<]$ -definierbar.

Fakt 5.6 $\text{Th}(\text{FO}[S], \mathbb{N})$ ist entscheidbar.

Beweis: Folgerung aus Satz 5.7 wegen $\text{FO}[S] \subseteq \text{MSO}[S]$

Peano-Arithmetik

first-order-Logik über der Signatur $\Sigma_{\text{Peano}} = \{<, +, \cdot\}$

Fakt 5.7 $\text{Th}(\text{FO}[<, +, \cdot], \mathbb{N})$ ist unentscheidbar.

Presburger-Arithmetik

first-order-Logik über der Signatur $\Sigma_{\text{Pres}} = \{<, +\}$

Fakt 5.8 $\text{Th}(\text{FO}[<, +], \mathbb{N})$ ist entscheidbar.

Binärdarstellung: $n = \sum_{i \in \mathbb{N}} a_i 2^i$

Abbildung $n \in \mathbb{N} \mapsto M_n = \{i \in \mathbb{N} \mid a_i = 1\} \subseteq \mathbb{N}$

Relation $p \subseteq (2^{\mathbb{N}})^3$ mit

$$(M_i, M_j, M_k) \in p \quad \text{gdw.} \quad i + j = k$$

ist $\text{MSO}[S]$ -definierbar, weil von einem BNFA erkennbar.

6 Baumsprachen

6.1 Endliche Terme und Bäume

Terme

(Wiederholung Informatik 1)

(funktionale) Signatur $\Sigma \subseteq S \times \mathbb{N}$ (hier immer endlich) mit Symbolemenge S und jedem Symbol zugeordneter Stelligkeit.

$$\Sigma^{(n)} = \{f \in S \mid (f, n) \in \Sigma\} \quad \Sigma = \bigcup_{n \in \mathbb{N}} \Sigma^{(n)}$$

Definition 6.1 Die Menge $\text{Term}(\Sigma)$ aller Grundterme über Σ ist die kleinste Menge M , welche für jedes $(f, n) \in \Sigma$ und jedes n -Tupel $(t_1, \dots, t_n) \in M^n$ auch $f(t_1, \dots, t_n)$ enthält.

Damit gilt insbesondere $\Sigma^{(0)} \subseteq \text{Term}(\Sigma)$.

Variablen $x \in X$ wie Konstanten:

$\text{Term}(\Sigma, X) = \text{Term}(\Sigma')$ mit $\Sigma'^{(0)} = \Sigma^{(0)} \cup X$ und für alle $n > 0$: $\Sigma'^{(n)} = \Sigma^{(n)}$

hier oft $X = \{\square\}$ (Variable \square symbolisiert eine „Lücke“)

$t \in \text{Term}(\Sigma, X)$ heißt *linear*, falls jede Variable in t höchstens einmal vorkommt.

Beispiel 6.1 Für $\Sigma = \{(a, 1), (b, 1)\}$ ist $\text{Term}(\Sigma) = \emptyset$ und es existiert eine Bijektion zwischen $\text{Term}(\Sigma, \{\square\})$ und $\{a, b\}^*$.

Σ -Algebra $S = (|S|, \llbracket \cdot \rrbracket_S)$ mit

1. Träger $|S| \neq \emptyset$
2. für alle $(f, n) \in \Sigma$ ist $\llbracket f \rrbracket_S : S^n \longrightarrow S$

$\varphi : |A| \longrightarrow |B|$ heißt *Homomorphismus* zwischen Σ -Algebren $A = (|A|, \llbracket \cdot \rrbracket_A)$ und $B = (|B|, \llbracket \cdot \rrbracket_B)$, falls

für alle $(f, n) \in \Sigma$ und alle $(a_1, \dots, a_n) \in |A|^n$ gilt $\varphi(\llbracket f \rrbracket_A(a_1, \dots, a_n)) = \llbracket f \rrbracket_B(\varphi(a_1), \dots, \varphi(a_n))$

freie Termalgebra über Σ : $T(\Sigma) = (\text{Term}(\Sigma), \llbracket \cdot \rrbracket_{T(\Sigma)})$ mit

für alle $(f, n) \in \Sigma$ ist $\llbracket f \rrbracket_{T(\Sigma)}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$

in Informatik 1 bewiesen:

Satz 6.1 Für jede Σ -Algebra S existiert genau ein Homomorphismus von $T(\Sigma)$ nach S .

Bäume

(hier immer endlich, geordnet, markiert)

Definition 6.2 Ein endlicher geordneter Baum b mit Markierungen aus einer Signatur $\Sigma \subseteq S \times \mathbb{N}$ ist eine Abbildung

$$b : \text{dom}(b) \longrightarrow \Sigma$$

mit einer präfixabgeschlossenen (aus $wu \in \text{dom}(b)$ folgt $w \in \text{dom}(b)$) Menge $\emptyset \neq \text{dom}(b) \subseteq \mathbb{N}^*$ für welche aus $b(w) = (f, n)$ folgt $\{i \in \mathbb{N} \mid wi \in \text{dom}(b)\} = \{1, \dots, n\}$.

$w \in \text{dom}(b)$ heißen *Positionen* in b .

ε ist die Wurzel in jedem b .

$B(\Sigma)$ Menge aller Bäume über Σ

Größe des Baumes $b \in B(\Sigma)$: $\text{Größe}(b) = |\text{dom}(b)|$

Höhe des Baumes $b \in B(\Sigma)$: $\text{Höhe}(b) = \max\{|w| \mid w \in \text{dom}(b)\}$

Teilbaum von $b \in B(\Sigma)$ an Position w : Baum $b|_w : \{u \mid wu \in \text{dom}(b)\} \longrightarrow \Sigma$, so daß für alle $u \in \text{dom}(b|_w)$ gilt $b|_w(u) = b(wu)$.

Teilbaum-Relation $a \trianglelefteq b$ gdw. $\exists w \in \text{dom}(b) : a = b|_w$ (Halbordnung)

Bijektion Term – Baum $T : \text{Term}(\Sigma) \longrightarrow B(\Sigma)$ rekursiv definiert durch

$T(f(t_1, \dots, t_n)) = b$ mit $b(\varepsilon) = (f, n)$ und für alle $i \in \{1, \dots, n\} : b|i = T(t_i)$

(Wir benutzen die jeweils geeignete Repräsentation.)

Substitution $\theta : X \longrightarrow \text{Term}(\Sigma, X)$ definiert eindeutige homomorphe Fortsetzung

$\theta : \text{Term}(\Sigma, X) \longrightarrow \text{Term}(\Sigma, X)$

$\theta : X \longrightarrow \text{Term}(\Sigma)$ heißt Grundsubstitution

Kontext über Σ : $C \in \text{Term}(\Sigma, \{\square\})$ mit genau einem Vorkommen von \square

Menge aller Kontexte über Σ heißt $\mathcal{C}(\Sigma)$.

Kontext \square heißt *trivial*.

Für Kontext $C \in \mathcal{C}(\Sigma)$ und Term $t \in \text{Term}(\Sigma)$ ist $C[t] := \theta(C) \in \text{Term}(\Sigma)$ für

$\theta : \{\square\} \longrightarrow \text{Term}(\Sigma)$ mit $\theta(\square) = t$

analog ist $C[C'] := \theta(C) \in \mathcal{C}(\Sigma)$ für $\theta = [\square \mapsto C']$

iteriertes Einsetzen:

für $C \in \mathcal{C}(\Sigma)$ ist $C^0 = \square$, $C^1 = C$ und $C^{n+1} = C[C^n]$.

Was bedeuten diese Begriffe für $\mathcal{C}(\Sigma_1)$ aus Beispiel 6.1?

6.2 Bottom-Up-Baumautomaten

Nichtdeterministische Bottom-Up-Baumautomaten (NFTA)

Definition 6.3 $\mathcal{A} = (Q, \delta, \beta)$ mit

1. endliche Menge $Q \neq \emptyset$ von Zuständen
2. akzeptierende Zustände $\beta \subseteq Q$
3. $\delta : \Sigma \longrightarrow (Q^+ \longrightarrow \{0, 1\})$, so daß für alle $(f, n) \in \Sigma$ gilt
 $\delta(f) : Q^{n+1} \longrightarrow \{0, 1\}$

heißt (Σ) -NFTA.

Keine Anfangszustände definiert, aber für jedes $a \in \Sigma^{(0)}$ (Blattmarkierung) eine Funktion $\delta(a) : \Sigma \longrightarrow (Q \longrightarrow \{0, 1\})$, also $\delta(a) \subseteq Q$

NFTA erkennen Grundterme.

Definition 6.4 Ein akzeptierender Lauf eines Σ -NFTA (Q, δ, β) auf $t \in \text{Term}(\Sigma)$ ist eine Funktion $r : \text{dom}(t) \longrightarrow Q$ mit

1. für jede Position $w \in \text{dom}(t)$ mit $t(w) = (f, n)$ gilt
 $(r(w1), \dots, r(wn), r(w)) \in \delta(f, n)$
 (insbesondere gilt $r(w) \in \delta(a, 0) \subseteq Q$ für alle $(a, 0) \in \Sigma$)
2. $r(\varepsilon) \in \beta$

Σ -NFTA \mathcal{A} erkennt die Menge $L(\mathcal{A}) \subseteq \text{Term}(\Sigma)$ (Baumsprache)

Zwei Σ -NFTA \mathcal{A} und \mathcal{B} heißen äquivalent gdw. $L(\mathcal{A}) = L(\mathcal{B})$.

REC_{NFTA} ist die Menge aller von einem NFTA erkennbaren Baumsprachen.

Idee (analog Abschnitt 4.1):

Darstellung von δ im NFTA $\mathcal{A} = (Q, \delta, \beta)$ als endliche Menge von Kacheln

$$T_{\mathcal{A}} = \{(q_1, \dots, q_n, q, f) \mid (q_1, \dots, q_n, q) \in \delta(f, n)\} \quad (3)$$

akzeptierender Lauf von \mathcal{A} auf $t \in \text{Term}(\Sigma)$ als Überdeckung (der „Astgabeln“) von t mit Kacheln aus der Menge $T_{\mathcal{A}}$.

Ein NFTA (Q, δ, β) heißt *vollständig* gdw. für alle $(f, n) \in \Sigma$ und alle $(p_1, \dots, p_n) \in Q^n$ gilt $\{p \in Q \mid (p_1, \dots, p_n, p) \in \delta(f, n)\} \neq \emptyset$

Satz 6.2 Zu jedem NFTA existiert ein äquivalenter vollständiger NFTA.

Beweis: Vervollständigung (analog Wortautomaten)

Ein NFTA (Q, δ, β) heißt *deterministisch* (DFTA) gdw. für alle $(f, n) \in \Sigma$ gilt

$$|\{q \in Q \mid (q_1, \dots, q_n, q) \in \delta(f, n)\}| \leq 1$$

$(\delta(f, n) \subseteq Q^{n+1})$ ist partielle Funktion $\delta(f, n) : Q^n \longrightarrow Q$

Für einen DFTA $\mathcal{A} = (Q, \delta, \beta)$ existiert also für jedes $(f, n) \in \Sigma$ und jedes Tupel $(q_1, \dots, q_n) \in Q^n$ höchstens eine Kachel $(q_1, \dots, q_n, q, f) \in T_{\mathcal{A}}$.

Satz 6.3 Zu jedem NFTA existiert ein äquivalenter DFTA.

Beweis: Potenzmengen-Konstruktion

Folgerung 6.1 $\text{REC}_{\text{NFTA}} = \text{REC}_{\text{DFTA}}$

6.3 Top-Down-Baumautomaten

Definition 6.5 $\mathcal{A} = (Q, \alpha, \delta)$ mit

1. endliche Menge $Q \neq \emptyset$ von Zuständen
2. Startzustände $\alpha \subseteq Q$
3. $\delta : \Sigma \longrightarrow (Q^+ \longrightarrow \{0, 1\})$, so daß für alle $(f, n) \in \Sigma$ gilt
 $\delta(f) : Q^{n+1} \longrightarrow \{0, 1\}$

heißt $(\Sigma\text{-})\text{TD} - \text{NFTA}$.

Ein akzeptierender Lauf eines $\Sigma\text{-NFTA}$ (Q, α, δ) auf $t \in \text{Term}(\Sigma)$ ist eine Funktion $r : \text{dom}(t) \longrightarrow Q$ mit

1. $r(\varepsilon) \in \alpha$
2. für jede Position $w \in \text{dom}(t)$ gilt:
 $(r(w1), \dots, r(wn), r(w)) \in \delta(t(w))$
(insbesondere gilt $r(w) \in \delta(a, 0) \subseteq Q$ für alle $(a, 0) \in \Sigma$)

Der Typ der Übergangsrelation δ ist für NFTA und TD – NFTA gleich.

Damit definiert Gleichung 3 auch zu jedem TD – NFTA $\mathcal{A} = (Q, \alpha, \delta)$ eine Menge $T_{\mathcal{A}}$ von Kacheln. Somit entspricht auch jeder akzeptierende Lauf von \mathcal{A} auf einem Term $t \in \text{Term}(\Sigma)$ einer Überdeckung von t mit Kacheln aus $T_{\mathcal{A}}$.

Fakt 6.1 Für jeden NFTA $\mathcal{A} = (Q, \delta, \beta)$ ist $\mathcal{B} = (Q, \beta, \delta)$ ein TD – NFTA (und umgekehrt) mit $L(\mathcal{A}) = L(\mathcal{B})$.

Folgerung 6.2 $\text{REC}_{\text{NFTA}} = \text{REC}_{\text{TD-NFTA}}$

Ein TD – NFTA (Q, α, δ) heißt *deterministisch* (TD – DFTA) gdw. für alle $(f, n) \in \Sigma$ gilt

$$|\{(p_1, \dots, p_n) \in Q^n \mid (p_1, \dots, p_n, q) \in \delta(f, n)\}| \leq 1$$

$(\delta(f, n) \subseteq Q^{n+1})$ ist partielle Funktion $\delta(f, n) : Q \longrightarrow Q^n$

Für einen TD – DFTA $\mathcal{A} = (Q, \alpha, \delta)$ existiert also für jedes $(f, n) \in \Sigma$ und jeden Zustand $q \in Q$ höchstens eine Kachel $(q_1, \dots, q_n, q, f) \in T_{\mathcal{A}}$.

Fakt 6.2 $L = \{f(a, b), f(b, a)\} \in \text{REC}_{\text{TD-NFTA}} \setminus \text{REC}_{\text{TD-DFTA}}$

Folgerung 6.3 $\text{REC}_{\text{TD-DFTA}} \subset \text{REC}_{\text{TD-NFTA}}$

6.4 Eigenschaften von REC_{NFTA}

Homomorphismen zwischen Σ -Algebren

Für Signatur Σ definiert jeder vollständige DFTA $\mathcal{A} = (Q, \delta, \beta)$ mit der Transitionsfunktion $\delta : \Sigma \longrightarrow (Q^* \longrightarrow Q)$

1. eine Σ -Algebra $Z_{\mathcal{A}} = (Q, [\cdot]_{Z_{\mathcal{A}}})$ mit

$$\forall (f, n) \in \Sigma \forall (q_1, \dots, q_n) \in Q^n : [f]_{Z_{\mathcal{A}}}(q_1, \dots, q_n) = \delta(f, n)(q_1, \dots, q_n)$$

insbesondere $[a]_{Z_{\mathcal{A}}} = \delta(a)$ für $(a, 0) \in \Sigma$

Es gilt also $[f]_{Z_{\mathcal{A}}} = \delta(f, n)$.

2. einen Homomorphismus $h : \text{Term}(\Sigma) \longrightarrow Q$ mit

$$h(f(t_1, \dots, t_n)) = [f]_{Z_{\mathcal{A}}}(h(t_1), \dots, h(t_n)) \quad (= \delta(f, n)(h(t_1), \dots, h(t_n)))$$

zwischen den Σ -Algebren $T(\Sigma)$ und $Z_{\mathcal{A}}$

(homomorphe Fortsetzung von $\delta : \Sigma \longrightarrow (Q^* \longrightarrow Q)$ auf Grundterme)

Transitionsfunktion $\delta : \Sigma \longrightarrow (Q^+ \longrightarrow Q)$ läßt sich außerdem homomorph fortsetzen auf

- Terme mit Variablen $\delta : \text{Term}(\Sigma, \{x_1, \dots, x_n\}) \longrightarrow (Q^n \longrightarrow Q)$ durch $h(x_i)(q_1, \dots, q_n) = q_i$
- Spezialfall Kontexte $\delta : \mathcal{C}(\Sigma) \longrightarrow (Q \longrightarrow Q)$ durch $h(\square)(q) = q$

Pumping-Lemma

Notation für $(g, 1) \in \Sigma$: $g^{i+1}(t) = g(g^i(t))$

$L = \{f(g^i(a), g^i(a) \mid i \in \mathbb{N}\} \notin \text{REC}_{\text{NFTA}}$

Satz 6.4 Für jede Baumsprache $L \in \text{REC}_{\text{NFTA}}$ existiert eine Zahl $n \in \mathbb{N}$, so daß für jeden Grundterm $t \in L$ mit $\text{Höhe}(t) > n$ ein Kontext C' , ein Kontext $C \neq \square$ und ein Grundterm $s \in \text{Term}(\Sigma)$ existieren, so daß

$$t = C'[C[s]] \quad \text{und} \quad \{C'[C^k[s]] \mid k \in \mathbb{N}\} \subseteq L$$

Folgerung 6.4 Für einen NFTA $\mathcal{A} = (Q, \delta, \beta)$ gilt

- $L(\mathcal{A}) \neq \emptyset$ gdw. $\exists t \in L(\mathcal{A}) : \text{Höhe}(t) \leq |Q|$,
- $|L(\mathcal{A})|$ ist unendlich gdw. $\exists t \in L(\mathcal{A}) : |Q| \leq \text{Höhe}(t) \leq 2|Q|$.

Baum-Homomorphismen

Signaturen $\Sigma, \Gamma, \mathbb{X} = \{x_i \mid i \in \mathbb{N}\}$

Wie wird aus $\text{Term}(\Gamma)$ eine Σ -Algebra?

für jedes $(f, n) \in \Sigma$ Funktion $\llbracket f \rrbracket_{T(\Gamma)} : \text{Term}(\Gamma)^n \longrightarrow \text{Term}(\Gamma)$ definieren

Jeder Term $t \in \text{Term}(\Gamma, \{x_1, \dots, x_n\})$ repräsentiert eine Funktion

$$f_t : \text{Term}(\Gamma)^n \longrightarrow \text{Term}(\Gamma), \quad \text{wobei} \quad f_t(t_1, \dots, t_n) = \theta(t) \\ \text{mit} \quad \forall i \in \{1, \dots, n\} : \theta(x_i) = t_i$$

Diese Zuordnung $\varphi : \text{Term}(\Gamma, \{x_1, \dots, x_n\}) \longrightarrow (\text{Term}(\Gamma)^n \longrightarrow \text{Term}(\Gamma))$ mit $\varphi(t) = f_t$ ist bijektiv.

Der kürzeren Notation wegen identifizieren wir im Folgenden jeden Term $t \in \text{Term}(\Gamma, \{x_1, \dots, x_n\})$ mit der von ihm repräsentierten Funktion $f_t : \text{Term}(\Gamma)^n \longrightarrow \text{Term}(\Gamma)$.

Jede Funktion $h_\Sigma : \Sigma \longrightarrow \text{Term}(\Gamma, \mathbb{X})$ mit

$\forall (f, n) \in \Sigma : h_\Sigma(f) \in \text{Term}(\Gamma, \{x_1, \dots, x_n\})$

(insbesondere für $(a, 0) \in \Sigma$ Grundterm $h(a) \in \text{Term}(\Gamma)$) definiert:

1. die Σ -Struktur $S = (\text{Term}(\Gamma) \llbracket \cdot \rrbracket_S)$ mit $\forall (f, n) \in \Sigma : \llbracket f \rrbracket_S = h_\Sigma(f)$
2. den (Baum-)Homomorphismus $h : \text{Term}(\Sigma) \longrightarrow \text{Term}(\Gamma)$ zwischen den Σ -Strukturen $T(\Sigma)$ und S durch

$$h(f(t_1, \dots, t_n)) = h_\Sigma(f)(h(t_1), \dots, h(t_n))$$

Funktion $h_\Sigma : \Sigma \longrightarrow \text{Term}(\Gamma, \mathbb{X})$ (bzw. $h_\Sigma : \Sigma \longrightarrow (\text{Term}(\Gamma)^* \longrightarrow \text{Term}(\Gamma))$) läßt sich außerdem homomorph fortsetzen auf

- Terme mit Variablen $h : \text{Term}(\Sigma, \{x_1, \dots, x_n\}) \longrightarrow (\text{Term}(\Gamma)^n \longrightarrow \text{Term}(\Gamma))$ durch $h(x_i)(s_1, \dots, s_n) = s_i$
- Spezialfall Kontexte $\delta : \mathcal{C}(\Sigma) \longrightarrow (\text{Term}(\Gamma) \longrightarrow \text{Term}(\Gamma))$ durch $h(\square)(s) = s$

Beispiel 6.2 Für $\Sigma = \{(g, 1), (a, 0)\}$, $\Gamma = \{(f, 2), (b, 0)\}$ und

$h_\Sigma(a) = f(b, b) \in \text{Term}(\Gamma)$, $h_\Sigma(g) = f(x_1, x_1) \in \text{Term}(\Gamma, \{x_1\})$ gilt

$L = \{g^i(a) \mid i \in \mathbb{N}\} \in \text{REC}_{\text{NFTA}}$, aber

$h(L) = \{t \in \text{Term}(\Gamma) \mid \exists k \in \mathbb{N} : \text{dom}(t) = \{w \in \{1, 2\}^* \mid |w| < k\}\} \notin \text{REC}_{\text{NFTA}}$.

Dieses Beispiel zeigt:

Fakt 6.3 REC_{NFTA} ist nicht unter beliebigen Homomorphismen abgeschlossen.

Der durch eine Funktion $h_\Sigma : \Sigma \longrightarrow \text{Term}(\Gamma, \mathbb{X})$ definierte Homomorphismus heißt *linear* gdw. für jedes $(f, n) \in \Sigma$ der Term $h_\Sigma(f) \in \text{Term}(\Gamma, \{x_1, \dots, x_n\})$ linear ist (d.h. $\forall i \in \{1, \dots, n\} : |\{w \in \text{dom}(h_\Sigma(f)) \mid h_\Sigma(f)(w) = x_i\}| \leq 1$).

Satz 6.5 Für jede Baumsprache $L \subseteq \text{Term}(\Sigma)$ mit $L \in \text{REC}_{\text{NFTA}}$ und jeden linearen Baum-Homomorphismus $h : \text{Term}(\Sigma) \longrightarrow \text{Term}(\Gamma)$ gilt $h(L) \in \text{REC}_{\text{NFTA}}$.

Satz 6.6 Für jede Baumsprache $L \subseteq \text{Term}(\Gamma)$ mit $L \in \text{REC}_{\text{NFTA}}$ und jeden Baum-Homomorphismus $h : \text{Term}(\Sigma) \longrightarrow \text{Term}(\Gamma)$ gilt $h^{-1}(L) \in \text{REC}_{\text{NFTA}}$.

Zu jeder endlichen Signatur Σ existieren eine endliche Signatur $\Gamma = \Gamma^{(2)} \cup \Gamma^{(0)}$ und ein linearer Homomorphismus $h : \text{Term}(\Sigma) \longrightarrow \text{Term}(\Gamma)$, z.B. wie folgt definiert:

$$\begin{aligned} \Gamma_0 &= \Sigma^{(0)} \cup \{c\} & h(a) &= a \\ \Gamma_1 &= \{(f, 2) \mid (f, 1) \in \Sigma\} & h(f) &= f(x_1, c) \\ \Gamma_2 &= \Sigma^{(2)} & h(f) &= f(x_1, x_2) \\ \forall n > 2 : \Gamma_n &= \{(f_i, 2) \mid f \in \Sigma^{(n)}, i \in \{1, \dots, n-1\}\} & h(f) &= f_1(x_1, f_2(x_2, \dots, f_{n-1}(x_{n-1}, x_n) \dots)) \end{aligned}$$

$$\Gamma = \bigcup_{i \in \mathbb{N}} \Gamma_i$$

Für jede Baumsprache $L \subseteq \text{Term}(\Sigma)$ und einen (immer existierenden) linearen Homomorphismus $h : \text{Term}(\Sigma) \longrightarrow \text{Term}(\Gamma)$ mit $\Gamma = \Gamma^{(2)} \cup \Gamma^{(0)}$ gilt also

$$L \in \text{REC}_{\text{NFTA}} \text{ gdw. } h(L) \in \text{REC}_{\text{NFTA}}.$$

Untersuchung von Baumsprachen über Signaturen $\Sigma = \Sigma^{(2)} \cup \Sigma^{(0)}$ genügt also.

Abschlußeigenschaften von REC_{NFTA}

Satz 6.7 Für Baumsprachen $L, L' \in 2^{\text{Term}(\Sigma)}$ folgt aus $L, L' \in \text{REC}_{\text{NFTA}}$:

- $L \cup L' \in \text{REC}_{\text{NFTA}}$,
- $2^{\text{Term}(\Sigma)} \setminus L \in \text{REC}_{\text{NFTA}}$,
- $L \cap L' \in \text{REC}_{\text{NFTA}}$,
- $h^{-1}(L) \in \text{REC}_{\text{NFTA}}$ für Homomorphismus h ,
- $h(L) \in \text{REC}_{\text{NFTA}}$ für linearen Homomorphismus h .

6.5 Logik $\text{MSO}_\Gamma[S_1, \dots, S_k]$

Man vergleiche mit Abschnitt 3.4

Syntax

Für jede (funktionale) Signatur Γ mit $k = \max\{n \mid \Gamma^{(n)} \neq \emptyset\}$ wird eine relationale Signatur Σ_Γ definiert:

$$\Sigma_\Gamma = \{(P_f, 1) \mid f \in \Gamma\} \cup \{(S_i, 2) \mid i \in \{0, \dots, k\}\} \cup \{(\leq, 2)\}$$

Atome in $\text{MSO}_\Gamma[S_1, \dots, S_k]$

$$P_a(x) \mid S_i(x, y) \mid x \sqsubseteq y \mid X(x)$$

Erweiterung auf Formeln in $\text{MSO}_\Gamma[S_1, \dots, S_k, \leq]$ und $\text{FO}_\Gamma[S_1, \dots, S_k, \leq]$ wie üblich (Abschnitt 3.4)

Baumstrukturen

Jeder Term $t \in \text{Term}(\Gamma)$ (also $t : \text{dom}(t) \rightarrow \Gamma$) definiert die *Baumstruktur* $\underline{t} = (\text{dom}(t), [\cdot]_{\underline{t}})$ mit

$$\begin{aligned} \forall i \in \{1, \dots, k\} : \quad [S_i]_{\underline{t}} &= \{(w, wi) \in \text{dom}(t)^2\} \\ [\leq]_{\underline{t}} &= \{(v, w) \in \text{dom}(t)^2 \mid v \sqsubseteq w\} \\ \forall (f, n) \in \Gamma : \quad [P_f]_{\underline{t}} &= \{w \in \text{dom}(t) \mid t(w) = f\} \end{aligned}$$

Definierbare Prädikate

$$\begin{aligned} (X = \emptyset) &:= \forall x (\neg(X(x))) \\ (X \subseteq Y) &:= \forall x (X(x) \rightarrow Y(x)) \\ (X = Y) &:= X \subseteq Y \wedge Y \subseteq X \\ \text{Sing}(X) &:= \neg(X = \emptyset) \wedge \forall Y (Y \subseteq X \rightarrow (X = Y \vee Y = \emptyset)) \\ \varphi_P(X, X_1, \dots, X_n) &:= \forall x (X(x) \leftrightarrow \bigvee_{i=1}^{n-1} (X_i(x) \wedge \bigwedge_{j=i+1}^n \neg X_j(x))) \\ (x = y) &:= (x \leq y) \wedge (y \leq x) \\ (x < y) &:= (x \leq y) \wedge \neg(y \leq x) \\ \text{wurzel}(x) &:= \forall x (\neg y < x) \\ \text{blatt}(x) &:= \forall x (\neg x < y) \\ \text{laub}(X) &:= \forall y (X(y) \leftrightarrow \text{blatt}(y)) \end{aligned}$$

Präfixordnung (reflexiv-transitiver Abschluß von $\bigcup_{i=1}^k S_k$):

$$(x \leq y) := \forall X \left(X(y) \wedge \forall z \forall z' \left(\bigvee_{i=1}^k (S_i(z, z') \wedge X(z')) \rightarrow X(z) \right) \rightarrow X(x) \right)$$

\leq ist also in $\text{MSO}_\Gamma[S_1, \dots, S_k, \leq]$ eliminierbar (ähnlich Fakt 3.3).

Abgeschlossenheit unter Präfixrelation

$$\varphi_{\text{Pref}}(X) = \forall x \forall y ((X(x) \wedge y \leq x) \rightarrow X(y))$$

Codierung von Termen:

$$\begin{aligned} \varphi_{\text{Term}}(X, X_1, \dots, X_n) = & \neg(X = \emptyset) \\ & \wedge \varphi_P(X, X_1, \dots, X_n) \\ & \wedge \varphi_{\text{Pref}}(X) \\ & \wedge \bigwedge_{i=1}^k \bigwedge_{f \in \Gamma^{(i)}} \left(\bigwedge_{l=1}^i \forall x \forall y ((X(x) \wedge S_l(x, y)) \rightarrow X(y)) \right. \\ & \left. \wedge \bigwedge_{l=i+1}^k \forall x \forall y ((X(x) \wedge S_l(x, y)) \rightarrow \neg X(y)) \right) \end{aligned}$$

Modifizierte Logik $\text{MSO}'_\Gamma[S_1, \dots, S_k]$ ohne first-order Variablen hat die gleiche Ausdrucksstärke wie $\text{MSO}_\Gamma[S_1, \dots, S_k]$ (analog Lemma 4.1).

Satz 6.8 (Doner, Thatcher / Wright 1968) Für jede Baumsprache $L \subseteq \text{Term}(\Gamma)$ gilt $L \in \text{REC}_{\text{NFTA}}$ gdw. $\exists \varphi \in \text{MSO}$.

Beweis: analog Satz 4.1

6.6 Logik WSkS

(vergleiche Abschnitt 5.7)

Logik $\text{MSO}_\emptyset[S_1, \dots, S_k]$ heißt WSkS (**W**eak monadic **S**econd order with **k** **S**uccessors).

Sätze $\varphi \in \text{WSkS}$ definieren *Eigenschaften* von Bäumen, z.B.

$$\forall x (\exists y S_1(x, y) \rightarrow \exists z S_2(x, z))$$

$$\forall x \exists y S_1(x, y) \text{ (erfüllt kein endlicher Baum)}$$

$$\exists X (\forall x \forall y (X(x) \wedge S_1(x, y) \rightarrow X(y)) \text{ (erfüllt jeder endliche Baum)})$$

Formeln $\varphi \in \text{WSkS}$ mit Variablen definieren *Relationen* zwischen Bäumen, z.B.

„weak“, weil für alle $\sigma : \mathbb{X}_{\text{MSO}} \rightarrow 2^{\text{dom}(t)}$ gilt $|\sigma(X)| < \omega$.

Satz 6.9 $\{\varphi \in \text{WSkS} \mid L(\varphi) \neq \emptyset\}$ ist entscheidbar.

Beweis analog Satz 5.7

6.7 Zusammenhänge zwischen Baum- und Wortsprachen

Blattsprachen

Blattwort eines Termes (Blattmarkierungen von links nach rechts gelesen):

$l : \text{Term}(\Sigma) \longrightarrow (\Sigma^{(0)})^+$, induktiv definiert durch

$$\begin{aligned} \text{für } t = a \in \Sigma^{(0)}: \quad l(a) &= a \\ l(f(t_1, \dots, t_n)) &= l(t_1) \cdots l(t_n) \end{aligned}$$

Blattsprache für $L \subseteq \text{Term}(\Sigma)$:

$$l(L) = \{l(t) \mid t \in L\}$$

Satz 6.10 *Eine Wortsprache L ist kontextfrei gdw. L die Blattsprache einer Baumsprache $L' \in \text{REC}_{\text{NFTA}}$ ist.*

Pfadsprachen

Pfadsprache eines Termes (Knotenmarkierungen von Wurzel zu Blatt):

$P : \text{Term}(\Sigma) \longrightarrow 2^{\Sigma^+}$, induktiv definiert durch

$$\begin{aligned} \text{für } t = a \in \Sigma^{(0)}: \quad P(a) &= a \\ P(f(t_1, \dots, t_n)) &= \bigcup_{i=1}^n \{f \cdot w \mid w \in P(t_i)\} \end{aligned}$$

Pfadsprache der Baumsprache $L \subseteq \text{Term}(\Sigma)$:

$$P(L) = \{P(t) \mid t \in L\} \subseteq \Sigma^+$$

Satz 6.11 *Für jede Baumsprache $L \in \text{REC}_{\text{NFTA}}$ gilt $P(L) \in \text{REG}$.*

Beweis: Konstruktion eines NFA \mathcal{A} mit $L(\mathcal{A}) = P(L)$ aus NFTA \mathcal{B} mit $L = L(\mathcal{B})$

Die Umkehrung von Satz 6.11 gilt nicht.

Gegenbeispiel: für $\Sigma = \{(f, 2), (a, 0)\}$ und $L = \{f(t, t) \mid t \in \text{Term}(\Sigma)\} \notin \text{REC}_{\text{NFTA}}$ ist $P(L) = \{f^i(a) \mid i < 0\} \in \text{REG}$

7 Sprachen unendlicher Bäume

Unendliche Bäume

Wir betrachten hier nur Signaturen $\Sigma = \Sigma^{(2)}$.

Ein *unendlicher Baum* $t \in \omega\text{Baum}(\Sigma)$ ist eine Funktion $t : \{0, 1\}^* \rightarrow \Sigma$.

Jedes ω -Wort $\xi \in \{0, 1\}^\omega$ bezeichnet einen *Pfad* in $\text{dom}(t)$.

analog Pfadsprachen für endliche Terme:

Pfadsprache $P : \omega\text{Baum}(\Sigma) \rightarrow 2^{\Sigma^\omega}$

Für $t \in \omega\text{Baum}\Sigma$ ist $P(t)$ die Menge der Markierungen aller Pfade in t von der Wurzel aus gelesen (ω -Wörter).

$$P(t) = \{t(\varepsilon)t(\xi_1)t(\xi_2)\dots \mid \xi = \xi_1\xi_2\dots \in \{0, 1\}^\omega\}$$

Für Sprachen $L \subseteq \omega\text{Baum}(\Sigma)$ gilt $P(L) = \bigcup_{t \in L} P(t)$.

7.1 ω -Baumautomaten

(Top-down-) ω -Baumautomat $\mathcal{A} = (Q, \alpha, \delta, \text{Acc})$ mit

- endlicher Zustandsmenge $Q \neq \emptyset$
- Menge von Anganzuständen $\alpha \subseteq Q$
- Übergangsrelation $\delta : \Sigma \rightarrow 2^{Q^3}$

Definitionen von Übergangsrelation δ , Kacheln $T_{\mathcal{A}}$ und Überdeckungen wie für Automaten über endlichen Bäumen.

Für einen Lauf $r : \{0, 1\}^* \rightarrow Q$ von \mathcal{A} auf t definieren wir die Pfadsprache $P : \{0, 1\}^\omega \rightarrow Q$ von r analog zur Pfadsprache eines Baumes durch

$$P(r) = \{r(\varepsilon)r(\xi_1)r(\xi_2)\dots \mid \xi = \xi_1\xi_2\dots \in \{0, 1\}^\omega\}$$

Akzeptanzbedingungen Acc ähnlich zu ω -Automaten (Abschnitt 5.1)

- Büchi-Baumautomat (BNFTA) mit $\text{Acc} = \text{Büchi}(\beta)$ für $\beta \subseteq Q$ mit

$$\text{Büchi}(\beta) = \{r \in Q^\omega \mid \text{Inf}(r) \cap \beta \neq \emptyset\}$$

- Muller-Baumautomat (MNFTA) mit $\text{Acc} = \text{Muller}(F)$ mit $F \subseteq 2^Q$

$$\text{Muller}(F) = \{r \in Q^\omega \mid \text{Inf}(r) \in F\}$$

- Rabin-Baumautomat (RNFTA) mit $\text{Acc} = \text{Rabin}(T)$ mit $T \subseteq (2^Q)^2$

$$\text{Rabin}(T) = \{r \in Q^\omega \mid \exists (F, G) \in T : \text{Inf}(r) \cap F = \emptyset \wedge \text{Inf}(r) \cap G \neq \emptyset\}$$

- Street-Baumautomat (SNFTA) mit $\text{Acc} = \text{Streett}(T)$ mit $T \subseteq (2^Q)^2$
 $\text{Streett}(T) = \{r \in Q^\omega \mid \forall (F, G) \in T : \text{Inf}(r) \cap F = \emptyset \vee \text{Inf}(r) \cap G \neq \emptyset\}$

\mathcal{A} akzeptiert den Baum $t \in \omega\text{Baum}$ gdw. es existiert ein *akzeptierender Lauf* $r : \{0, 1\}^* \rightarrow Q$ von \mathcal{A} auf t , d.h.

- $r(\varepsilon) \in \alpha$,
- für alle $w \in \{0, 1\}^*$ gilt $(r(w0), r(w1), r(w)) \in \delta(t(w))$,
- $P(r) \subseteq \text{Acc}$.

ω -Baumautomat \mathcal{A} akzeptiert die Sprache $L(\mathcal{A})$ (Menge aller Bäume $t \in \omega\text{Baum}(\Sigma)$, für die ein akzeptierender Lauf von \mathcal{A} auf t existiert).

Mengen $\text{REC}_{\text{BNFTA}}$, $\text{REC}_{\text{MNFTA}}$, $\text{REC}_{\text{RNFTA}}$, $\text{REC}_{\text{SNFTA}}$ aller durch Büchi-, Muller-, Rabin- und Street- ω -Baumautomaten erkennbaren ω -Baumsprachen

Fakt 7.1 $\text{REC}_{\text{BNFTA}}$ ist nicht unter Komplement abgeschlossen.

Beispiel: für $\Sigma = \{(a, 2), (b, 2)\}$ ist $L = \{t \in \omega\text{Baum}(\Sigma) \mid \exists \xi \in P(t) : |a|_\xi = \omega\} \in \text{REC}_{\text{bnfta}}$ (durch BNFTA $\mathcal{A} = (\{q_a, q_b, p\}, \{q_a, q_b\}, \delta, \text{Büchi}(\{q_a, p\}))$), aber $2^{\omega\text{Baum}(\Sigma)} \setminus L \notin \text{REC}_{\text{BNFTA}}$

Für MNFTA $\mathcal{A} = (\{q_a, q_b, p\}, \{q_a, q_b\}, \delta, \text{Muller}(F))$ mit $F = \{\{q_a, q_b\}, \{q_a\}, \{p\}\}$ gilt $L = L(\mathcal{A})$

Für MNFTA $\mathcal{B} = (\{q_a, q_b\}, \{q_a, q_b\}, \delta, \text{Muller}(\{q_b\}))$ mit $\delta(a) = \{(q_a, q_a, q_a), (q_a, q_a, q_b)\}$, $\delta(b) = \{(q_b, q_b, q_a), (q_b, q_b, q_b)\}$ gilt $2^{\omega\text{Baum}(\Sigma)} \setminus L = L(\mathcal{B})$

Satz 7.1 $\text{REC}_{\text{MNFTA}} = \text{REC}_{\text{RNFTA}} = \text{REC}_{\text{SNFTA}}$

Beweisidee: Jeder ω -Baumautomat ist eine Kombination aus einem ω -Baumautomaten ohne Akzeptanzbedingung (zum Raten eines Laufes $r : \{0, 1\}^* \rightarrow Q$) und einem deterministischen ω (-Wort)-Automaten auf $P(r)$ (zum Testen der Akzeptanzbedingung auf jedem Pfad)

Abschlußeigenschaften von $\text{REC}_{\text{MNFTA}}$

Abschluß unter \cup und Projektion läßt sich analog zu ω -Wörtern zeigen.

Satz 7.2 (Rabin, 1969) Für jede Sprache $L \in \text{REC}_{\text{MNFTA}}$ gilt $2^{\omega\text{Baum}(\Sigma)} \setminus L \in \text{REC}_{\text{MNFTA}}$.

(hier ohne Beweis)

$\text{REC}_{\text{MNFTA}}$ ist abgeschlossen unter

- Vereinigung
- Komplement
- Projektion

7.2 Logik über unendlichen Bäumen

Definitionen nach Abschnitt 6.6

$\text{MSO}_\Sigma[S_1, \dots, S_k]$

Jeder Baum $t \in \omega\text{Baum}(\Sigma)$ definiert *Baumstruktur*

$$\underline{t} = (\{0, 1\}^\omega, \llbracket \underline{t} \rrbracket)$$

mit Interpretation von S_0, S_1, \leq, P_f wie in Abschnitt 6.6.

Jeder Satz $\varphi \in \text{MSO}_\Sigma[S_1, \dots, S_k]$ definiert die Sprache unendlicher Bäume $L^\omega(\varphi) = \{t \in \omega\text{Baum}\Sigma \mid \llbracket \varphi \rrbracket_{\underline{t}} = 1\}$.

Satz 7.3 Für jede ω -Baumsprache $L \subseteq 2^{\omega\text{Baum}(\Sigma)}$ gilt

$$L \in \text{REC}_{\text{MNFTA}} \text{ gdw. } \exists \varphi \in \text{MSO}_\Sigma[S_0, S_1] : L = L(\varphi)$$

Logiken ohne Buchstabenprädikate:

S1S = $\text{MSO}_\emptyset[S]$ (über endlichen und unendlichen Wörtern)

S2S = $\text{MSO}_\emptyset[S_0, S_1]$ (über endlichen und unendlichen Binärbäumen)

SkS = $\text{MSO}_\emptyset[S_1, \dots, S_k]$ (über endlichen und unendlichen Bäumen vom Grad k)

$S\omega S$ = $\text{MSO}_\emptyset[\{S_i \mid i \in \mathbb{N}\}]$ (über endlichen und unendlichen Bäumen, in denen jeder Knoten beliebigen

Die folgende Transformation erlaubt die Einbettung vollständiger Ternärbäume $t : \{1, 2, 3\}^* \rightarrow \Sigma$ in vollständige Binärbäume $t' : \{0, 1\}^* \rightarrow \Sigma$.

Die Menge der Positionen in t' , auf die die Knoten aus t abgebildet werden, ist

$$P = (10 + 110 + 1110)^* \subseteq \{0, 1\}^*$$

Übersetzung der Positionen $T : \{1, 2, 3\}^* \rightarrow \{0, 1\}^*$:

$$T(w_1 w_2, \dots, w_n) = 1^{w_1} 0 1^{w_2} 0 \dots 1^{w_n} 0$$

Übersetzung der Bäume $T : (\{1, 2, 3\}^* \rightarrow \Sigma) \rightarrow (\{0, 1\}^* \rightarrow \Sigma \cup \{*\})$

$$T(t) = t' \quad t'(w) = \begin{cases} t(w') & \text{falls } w \in P \wedge T(w') = w \\ * & \text{sonst} \end{cases}$$

Logische Beschreibung der Menge $P \subseteq \text{dom}(t')$:

$$P(x) = \forall Y (Y(x) \wedge \forall y ((Y(y10) \vee Y(y110) \vee Y(y1110)) \rightarrow Y(y)) \rightarrow Y(\varepsilon))$$

Übersetzung der Nachfolgerrelationen S_1, S_2, S_3 auf $\text{dom}(t) = \{1, 2, 3\}^*$ in Formeln aus $\text{FO}[S_0, S_1]$ auf $\text{dom}(T(t)) = \{0, 1\}^*$

$$\varphi_{S_1}(x, y) = \exists z (S_1(x, z) \wedge S_0(z, y))$$

$$\varphi_{S_2}(x, y) = \exists u \exists v (S_1(x, u) \wedge S_1(u, v) \wedge S_0(v, y))$$

$$\varphi_{S_3}(x, y) = \exists u \exists v \exists w (S_1(x, u) \wedge S_1(u, v) \wedge S_1(v, w) \wedge S_0(w, y))$$

Analog lassen sich Bäume mit Knoten anderer (auch verschiedener) Grade in vollständige Binärbäume einbetten.

Folgerung 7.1 Die Logiken S2S, SkS, S ω S sind entscheidbar.

7.3 Entscheidbarkeitskriterien für Logiken

Logik (L, M, \models) mit

- Menge L von Formeln
- Menge M von Interpretationen
- Modellrelation $\models \subseteq M \times L$

Bezeichnungen wie z.B. $\text{FO}_A[S]$, $\text{MSO}_\Sigma[S_0, S_1]$ definieren die Formelmenge L und mit „über ω -Wörtern“ „über endlichen Binärbaumen“ wird die Menge M angegeben.

Modelle einer Formel $\varphi \in L$: $\text{Mod}(\varphi) = \{m \in M \mid m \models \varphi\}$

Im folgenden nehmen wir an, daß die betrachtete Logik (L, M, \models) eine Prädikatenlogik oder in eine solche einbettbar ist.

(Prädikatenlogik: L definiert durch endliche Signatur Σ für Atome, M Menge von Σ -Strukturen)

Damit ist L abzählbar und jeder Interpretation $m \in M$ kann eine „Größe“ $|m|$ (Kardinalzahl) zugeordnet werden, so daß für jedes $n \in \mathbb{N}$ nur endlich viele $m \in M$ mit $|m| = n$ existieren.

Das gilt für alle hier bisher untersuchten Logiken.

Menge aller in M allgemeingültigen Formeln $\{\varphi \mid \text{Mod}(\varphi) = M\}$. Diese Menge ist aufzählbar (z.B. Sequenzenkalkül, Resolution).

Wiederholung Entscheidbarkeit:

Eine Logik (L, M, \models) heißt *entscheidbar*, wenn die charakteristische Funktion $\chi_{\{\varphi \mid \text{Mod}(\varphi) = M\}} : L \rightarrow \{0, 1\}$ berechenbar ist.

Entscheidbarkeit wird durch Angabe eines *Entscheidungsverfahrens* gezeigt.

Für manche Klassen von Logiken lassen sich allgemeine Entscheidungsverfahren angeben.

Beschränkte-Modell-Eigenschaft (BMP)

L erfüllt BMP gdw. $\forall \varphi \in L \exists k(\varphi) (\text{Mod}(\varphi) \neq \emptyset \rightarrow \exists m \in \text{Mod}(\varphi) : |m| \leq k(\varphi))$

Satz 7.4 Jede Logik mit BMP ist entscheidbar.

Entscheidungsverfahren: Suche nach Modell für $\neg\varphi$ für alle (das sind endliche viele) Modelle m mit $|m| < k(\varphi)$

Beispiel: Aussagenlogik, $\text{FO}[\Sigma_R^{(1)}]$ haben BMP

Endliche-Modell-Eigenschaft (FMP)

L erfüllt FMP gdw. $\forall \varphi \in L (\text{Mod}(\varphi) \neq \emptyset \rightarrow \exists m \in \text{Mod}(\varphi) : |m| \in \mathbb{N})$

Satz 7.5 *Jede Logik mit FMP ist entscheidbar.*

Wir wissen

- $\{\varphi \mid \text{Mod}(\varphi) = M\}$ ist aufzählbar (Beweise für Allgemeingültigkeit von φ).
- In Logiken mit FMP ist $\text{Mod}(\neg\varphi)$ (Menge der Gegenbeispiele gegen Allgemeingültigkeit von φ) aufzählbar (für $i \leftarrow [1..]$ teste alle $m \in M$ mit $|m| = i$).

Entscheidungsverfahren: Kombination von Suche nach Beweis für φ und Modell für $\neg\varphi$ (Gegenbeispiel).

schrittweise abwechselnde Ausführung

Baummodelleigenschaft (TMP)

L erfüllt TMP gdw. $\forall \varphi \in L (\text{Mod}(\varphi) \neq \emptyset \rightarrow \exists m \in \text{Mod}(\varphi) : m \text{ ist Baummodell})$.

Satz 7.6 *Jede Logik mit TMP ist entscheidbar.*

Entscheidungsverfahren: (Suche nach Baummodell für $\neg\varphi$)

berechne ω -Baumautomaten $\mathcal{A}_{\neg\varphi}$

φ ist allgemeingültig gdw. $L(\mathcal{A}_{\neg\varphi}) = \emptyset$ (entscheidbar)

8 Modallogik

Idee: Formale Beschreibung von Aussagen wie
Aussage φ ist „möglich“, „notwendig“, „erlaubt“, „bekannt“...

8.1 Syntax

Erweiterung der Aussagenlogik um *Modalitäten* \diamond, \square
Formeln in $ML(P)$ mit Menge P von Aussagenvariablen:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \diamond x\varphi \quad \text{mit Aussagenvariablen } p \in P$$

8.2 Kripke-Strukturen

Kripke-Frame (W, R) mit

- Menge W von *Welten*,
- *Erreichbarkeitsrelation* $R \subseteq W^2$

Kripke-Frames sind also (möglicherweise unendliche) Graphen.

Kripke-Struktur $K = (W, R, V)$ mit

- Kripke-Frame (W, R)
- Variablenbelegung $V : W \rightarrow (P \rightarrow \{0, 1\})$
ordnet jeder Welt eine Belegung der Aussagenvariablen zu

Kripke-Strukturen sind also Graphen, deren Knoten mit Mengen aus 2^P gefärbt sind.

8.3 Modellrelation

$ML(p)$ -Formeln werden in Kripke-Strukturen $(K, u) = (W, R, V, u)$ mit einer ausgezeichneten Welt $u \in W$ interpretiert.

induktive Definition für $(K, u) \models \varphi$

$$\begin{aligned} (K, u) \models p & \text{ gdw. } p \in V(u) \\ (K, u) \models \neg\varphi & \text{ gdw. } (K, u) \not\models \varphi \\ (K, u) \models \varphi \wedge \psi & \text{ gdw. } (K, u) \models \varphi \text{ und } (K, u) \models \psi \\ (K, u) \models \diamond\varphi & \text{ gdw. } \exists w \in W (R(u, w) \wedge (K, w) \models \varphi) \end{aligned}$$

8.4 Einbettung in FO

Für jede Menge P von Aussagenvariablen definieren wir die relationale Signatur $\Sigma(P) = \{(R, 2)\} \cup \{(a, 1) \mid a \in P\}$

Übersetzung der Kripke-Struktur $K = (W, R, V)$ in die $\Sigma(P)$ -Struktur $S_K = (W, [\cdot]_{S_K})$ mit

$$\begin{aligned} [R]_{S_K} &= R \\ \forall a \in P : [a]_{S_K} &= \{w \in W \mid a \in V(w)\} \end{aligned}$$

Übersetzung $T : \text{ML}(P) \times \{x_i \mid i \in \mathbb{N}\} \longrightarrow \text{FO}[\Sigma(P)]$

$$\begin{aligned} \text{für } a \in P : T(a, x_i) &= a(x_i) \\ T(\neg\varphi, x_i) &= \neg T(\varphi, x_i) \\ T(\varphi \wedge \psi, x_i) &= T(\varphi, x_i) \wedge T(\psi, x_i) \\ T(\diamond\varphi, x_i) &= \exists x_j (R(x_i, x_j) \wedge T(\varphi, x_j)) \end{aligned}$$

wobei x_j eine neue Variable ist, die in $T(\varphi, x_i)$ nicht vorkommt.

Satz 8.1 Für jede Formel $\varphi \in \text{ML}(P)$ und jede Kripke-Struktur (K, u) gilt

$$(K, u) \models \varphi \quad \text{gdw.} \quad \llbracket \theta(T(\varphi, x_0)) \rrbracket_{S_K} = 1 \quad \text{für } \theta(x_0) = u.$$

Damit lassen sich für jede Formel $\varphi \in \text{ML}(P)$ Erfüllbarkeit und Gültigkeit durch die entsprechenden Tests in $\text{FO}[\Sigma(P)]$ bestimmen.

Läßt sich auch jede Formel aus $\text{FO}[\Sigma(P)]$ in eine „äquivalente“ $\text{ML}(P)$ -Formel übersetzen?

8.5 Bisimulation

Für zwei Kripke-Strukturen (W, R, V, u) und (W', R', V', u') heißt eine Relation $Z \subseteq W \times W'$ *Bisimulation* gdw.

1. $(u, u') \in Z$
2. $\forall (w, w') \in Z (V(w) = V'(w'))$
3. $\forall (w, w') \in Z \forall (w, v) \in R \exists v' \in W' ((w', v') \in R' \wedge (v, v') \in Z)$
4. $\forall (w, w') \in Z \forall (w', v') \in R' \exists v \in W ((w, v) \in R \wedge (v, v') \in Z)$

Existiert eine Bisimulation zwischen zwei Kripke-Strukturen $(K, u), (K', u')$, dann heißen diese *bisimilar*.

Satz 8.2 Für jede Formel $\varphi \in \text{ML}(P)$ und zwei bisimulare Kripke-Strukturen $(K, u), (K', u')$ gilt

$$(K, u) \models \varphi \quad \text{gdw.} \quad (K', u') \models \varphi$$

ML(P)-Formeln können also bisimilare Kripke-Strukturen nicht unterscheiden.

Für $P = \emptyset$ sind die Kripke-Strukturen $K = (\{o\}, \{(o, o)\}, \emptyset, o)$ und $K' = (\mathbb{N}, \{(i, i+1) \mid i \in \mathbb{N}\}, \emptyset, 0)$ bisimilar durch die Bisimulation $Z = \{(o, i) \mid i \in \mathbb{N}\}$.

Für die Formel $\varphi = \exists x R(x, x)$ gilt $\llbracket T(\varphi, x_0)[x_0 \mapsto o] \rrbracket_{S_K} = 1$, aber $\llbracket T(\varphi, x_0)[x_0 \mapsto 0] \rrbracket_{S_{K'}} = 0$.

Daher gilt der folgende

Fakt 8.1 $\text{FO}[\Sigma(P)]$ läßt sich nicht in $\text{ML}(P)$ einbetten.

8.6 Baummodelleingenschaft von ML

Für jede Kripke-Struktur $K = (W, R, V, u)$ definieren wir die Kripke-Struktur $K' = (W', R', V', u')$ (Entfaltung von K) mit

$$\begin{aligned} W' &= \{w_0 \dots w_n \mid n \in \mathbb{N} \wedge w_0 = u \wedge \forall i \in \{0, \dots, n-1\} : (w_i, w_{i+1}) \in R\} \\ R' &= \{(w_0 \dots w_n, w_0 \dots w_n v) \in W' \mid (w_n, v) \in R\} \\ V'(w_0 \dots w_n) &= V(w_n) \\ u' &= \{u\} \end{aligned}$$

Fakt 8.2 Jede Kripke-Struktur $K = (W, R, V, u)$ ist bisimilar zur oben definierten Kripke-Struktur $K' = (W', R', V', u')$.

Der der oben definierten Kripke-Struktur zugrundeliegende Kripke-Frame (W', R') ist ein kreisfreier Graph.

Gilt für die Relation R aus K : $\forall x \exists y R(x, y)$, dann ist der Kripke-Frame (W', R') auch zusammenhängend und damit ein (unendlicher) Baum.

Die $\Sigma(P)$ -Struktur $S_{K'}$ läßt sich in ein Baummodell t mit $\text{dom}(t) = \mathbb{N}^\omega$ einbetten. Damit hat $ML(P)$ die TMP und es gilt

Satz 8.3 $ML(P)$ ist entscheidbar.

9 Sprachen endlicher Bilder

(Bild = zweidimensionales Wort)

Menge der Positionen $\text{dom}(p) = \{1, \dots, h(p)\} \times \{1, \dots, b(p)\}$ (Höhe $h(p)$, Breite $b(p)$)

Bild $p : \text{dom}(p) \longrightarrow \Sigma$ (Matrix mit Einträgen aus Σ)

$\Sigma^{m,n}$ Menge aller Bilder mit $\text{dom}(p) = \{1, \dots, m\} \times \{1, \dots, n\}$ über Σ

leeres Bild ϵ , $\Sigma^{++} = \bigcup_{(m,n) \in (\mathbb{N} \setminus \{0\})^2} \Sigma^{m,n}$

Bildsprache: Menge von Bildern, z.B.

$$\begin{aligned} L_w &= \{p \in \{a\}^{++} \mid h(p) = 1\} \\ L_{23} &= \{p \in \{a\}^{++} \mid p(2,3) = a\} \\ L_Q &= \{p \in \{a\}^{++} \mid h(p) = b(p)\} \\ L_{qq} &= \{pp \in \{a\}^{++} \mid p \in L_Q\} \end{aligned}$$

9.1 Tiling-Systeme

(analog zu nichtdeterministischen Automaten)

Alphabet A

$A' = A \cup \{\#\}$ (Randsymbol $\#$)

Tiling-System $\mathcal{A} = (Q, \delta)$, mit

1. endliche Menge $Q \neq \emptyset$ von Zuständen

2. Menge $\delta \subseteq (A \cup \{\#\}) \times Q$ von Kacheln $\begin{pmatrix} (a_1, q_1) & (a_2, q_2) \\ (a_3, q_3) & (a_4, q_4) \end{pmatrix}$

Akzeptierender Lauf von $\mathcal{A} = (Q, \delta)$ auf $p \in A^{++}$:

$r : \{0, \dots, h(p) + 1\} \times \{0, \dots, b(p) + 1\} \longrightarrow Q$ mit

$$\forall (i, j) \in \{0, \dots, h(p)\} \times \{0, \dots, b(p)\} : \begin{pmatrix} (a_{i,j}, q_{i,j}) & (a_{i,j+1}, q_{i,j+1}) \\ (a_{i+1,j}, q_{i+1,j}) & (a_{i+1,j+1}, q_{i+1,j+1}) \end{pmatrix} \in \delta$$

Das Tiling-System $\mathcal{A} = (Q, \delta)$ akzeptiert ein Bild $p \in A^{++}$ gdw. ein akzeptierender Lauf von \mathcal{A} auf p existiert.

$L(\mathcal{A})$ = Menge aller von \mathcal{A} akzeptierten Wörter.

$\text{REC}_{\text{Tiling}} = \{L \mid \exists \text{Tiling-System } \mathcal{A} : L = L(\mathcal{A})\}$ Menge aller erkennbaren Sprachen endlicher Bilder

Satz 9.1 $\text{REC}_{\text{Tiling}}$ ist unter \cup , \cap und Projektion abgeschlossen.

Fakt 9.1 Für jedes Alphabet A mit $|A| > 1$ gilt

1. $L_{qq} = \{pp \in A^{++} \mid p \in L_Q\} \notin \text{REC}_{\text{TS}}$
2. $A^{++} \setminus L_{qq} \in \text{REC}_{\text{TS}}$

Aber: für $|A| = 1$ gilt $L_{qq} = \{pp \mid p \in L_Q\} \in \text{REC}_{\text{TS}}$

Folgerung 9.1 REC_{TS} ist nicht unter Komplement abgeschlossen.

Satz 9.2 Für ein Tiling-System \mathcal{A} ist unentscheidbar, ob $L(\mathcal{A}) = \emptyset$ gilt.

Beweis: Reduktion auf Halteproblem für TM

9.2 Logik über Bildsprachen

Syntax

Alphabet A

Logik $\text{MSO}_A[S_1, S_2]$ wie in Abschnitt 6.6, aber mit anderer Interpretation der Nachfolgerrelationen S_1, S_2 .

Logik $\text{EMSO}_A[S_1, S_2]$ (Existentielle MSO):

$\varphi ::= P \mid \neg\varphi \mid \varphi * \psi \mid Qx\varphi \mid \exists X\eta$

mit MSO-Atomen $P, x \in \mathbb{X}_{\text{FO}}, X \in \mathbb{X}_{\text{MSO}}, \text{EMSO}_A[S_1, S_2]$ -Formeln φ, ψ ,
FO-Formeln $\eta, * \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}, Q \in \{\forall, \exists\}$

Semantik

Jedes Bild $p \in A^{++}$ (also $p : \text{dom}(p) \longrightarrow A$) definiert die *Bildstruktur*
 $\underline{p} = (\text{dom}(p), [\cdot]_{\underline{p}})$ mit

$$\begin{aligned} [S_1]_{\underline{p}} &= \{((i, j), (i+1, j)) \in \text{dom}(p)^2\} \\ [S_2]_{\underline{p}} &= \{((i, j), (i, j+1)) \in \text{dom}(p)^2\} \\ \forall a \in A: [P_a]_{\underline{p}} &= \{(i, j) \in \text{dom}(p) \mid p(i, j) = a\} \end{aligned}$$

$$\varphi_o(x) := \neg\exists x S_1(y, x)$$

$$\varphi_l(x) := \neg\exists x S_2(y, x)$$

$$\varphi_{lo}(x) := \varphi_o(x) \wedge \varphi_l(x)$$

analog $\varphi_u(x), \varphi_r(x), \varphi_{ro}(x), \dots$

von Satz $\varphi \in \text{MSO}_A[S_1, S_2]$ definierte Bildsprache

$$L(\varphi) = \{p \mid \llbracket \varphi \rrbracket_{\underline{p}} = 1\}$$

Beispiele: (für $A = \{a, b\}$)

$$\begin{aligned}\varphi &= \exists x \exists x' \exists y \exists y' (\varphi_{lo}(x) \wedge S_1(x, x') \wedge S_2(x', y) \wedge S_2(y, y') \wedge P_a(y')) \\ L(\varphi) &= L_{23}\end{aligned}$$

$$\begin{aligned}\varphi &= \exists X \left(\begin{array}{l} \exists x (\varphi_{lo}(x) \wedge X(x)) \wedge \\ \forall x \forall y \forall z (X(x) \wedge S_1(x, y) \wedge S_2(y, z) \rightarrow X(z)) \\ \wedge \forall x ((\varphi_u(x) \vee \varphi_r(x)) \rightarrow (\neg X(x) \wedge \varphi_{ru}(x))) \end{array} \right) \\ L(\varphi) &= L_Q\end{aligned}$$

Die Bildsprache $L \subseteq A^{++}$ heißt $\text{EMSO}_A[S_1, S_2]$ -definierbar, falls ein Satz $\varphi \in \text{EMSO}_A[S_1, S_2]$ mit $L = L(\varphi)$ existiert.

Satz 9.3 (Giammaresi, Restivo, Seibert, Thomas 1994) Für jede Bildsprache $L \subseteq A^{++}$ gilt $L \in \text{RECTS}$ gdw. $\exists \varphi \in \text{EMSO}_A[S_1, S_2] : L = L(\varphi)$.

L_{qq} ist $\text{MSO}_A[S_1, S_2]$ -definierbar, aber für $|A| > 1$ nicht $\text{EMSO}_A[S_1, S_2]$ -definierbar.