FGdI 1

Donnerstag, 23. Juli 2015 13:05

Alphabet	Nicht leere, endliche Menge von Buchstaben, Zeichen, Symbolen (Σ)
Σ-Wort	Endliche Sequenz von Buchstaben aus dem Alphabet
Σ^*	Menge aller möglichen Wörter
Σ^+	Menge aller möglichen Wörter ohne ε
Σ^n	Menge der Wörter mit Länge n
ε	leere Wort
$\lvert u \rvert$	Länge eines Wortes
Präfixrelation $\preccurlyeq$	$u \preccurlyeq v$ gdw. v mit dem Wort u beginnt
Äquivalenzrelation	reflexiv, symmetrisch, transitiv (Bsp. Alle Wörter der Länge 5)
Ordnungsrelation	reflexiv, antisymmetrisch, transitiv
Äquivalenzklasse	R sei Äquiv.-Rel. auf A. Dann $[A]_R = \{b \in A : aRb\}$
Monoid	Trägermenge; assoziative 2-stellige Operation; neutrales Element; Kein Inverses
	Bsp: Wortmonoid $(\Sigma^*, \cdot, \varepsilon)$ Σ^* mit d. Konkatination u. dem leeren Wort
Gruppe	Trägermenge; assoziative 2-stellige Operation; neutrales Element; Inverses
	Bsp: $(\mathbb{Z}, +, 0)$ Ganze Zahlen mit d. Addition und Null
Boolesche Algebra	Trägermenge; 2 2-stellige assoziative und kommutative Operationen; 1 einstellige Operat.; 2 Konstanten
	$\mathcal{B} = (B, \cdot, +, ', 0, 1)$
	Bsp: $(\{0,1\}, \wedge, \vee, \neg, 0, 1)$ Wahr-Falsch-Logik
	Bsp: $(P(M), \cap, \cup, ^-, \emptyset, M)$ Mengenlogik
Homomorphismus	In Struktur A xRy dann auch in B $fxRfy$
Isomorphismus	Bijektiver Homomorphismus. Umkehrung auch Homomorphismus
Kontraposition	$\neg B \Rightarrow \neg A$ beweisen um $A \Rightarrow B$ zu zeigen
Indirekter Beweis	$\neg A$ unmöglich; also A
Äquivalenzen / Biimplikation	$A \Leftrightarrow B$ ist gleich $A \Rightarrow B \wedge B \Rightarrow A$
Implikationskette	$A \Rightarrow B$ zeigen, indem $A \Rightarrow C$ und $C \Rightarrow B$
Vollständige Induktion.	Es gilt an Stelle A; Wenn es an Stelle X gilt, dann auch bei X+1
Sprache	Menge von Σ-Wörtern $L \subseteq \Sigma^*$
Leere Sprache	$\emptyset$
Volle Sprache	Σ^*
Sprachoperationen	Komplement $\bar{L} = \Sigma^* \setminus L$
	Schnitt $(L_1, L_2) \to L_1 \cap L_2$
	Vereinigung $(L_1, L_2) \to L_1 \cup L_2$
	Konkatination $(L_1, L_2) \to L_1 \cdot L_2 = \{u \cdot v : u \in L_1, v \in L_2\}$
	↳ Spezialfall: $L \cdot L = L^2$ oder $L^3, L^4, \ldots$
	Sternoperation: $L \to L^* := \{u_1 \cdot \ldots \cdot u_n : u_1, \ldots, u_n \in L, n \in \mathbb{N}\}$ Auch das leere Wort ist drin
Reguläre Sprachen	Gebildet aus einfachen Sprachen $\emptyset$ und $\{a\}$ für $a \in \Sigma$
	und den Operationen Vereinigung $(\cap, +)$, Konkatination $(\cdot)$ und Stern $(^*)$
	Bsp: $\{a,b\} \cdot \{c\} = \{ac, bc\}$
	$\emptyset^* = \{\varepsilon\}$
Reguläre Ausdrücke	Syntax zur Beschreibung von reg. Sprachen

$\emptyset^* = \{\varepsilon\}$

Reguläre Ausdrücke Syntax zur Beschreibung von reg. Sprachen

Bsp.: $a(a+b)^*$ beschreibt alle ab-Wörter, die mit a beginnen.

Endliche Automaten

DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$ Von jedem Knoten muss man mit jeder Eingabe weg kommen

Σ Alphabet der Kantensymbole

Q endl. nicht leere Zustandsmenge

$q_0 \in Q$ Startzustand

$\delta: Q \times \Sigma \to Q$ Übergangsfunktion $\hat{\delta}: Q \times \Sigma^* \to Q$: Abarbeiten eines Wortes

$A \subseteq Q$ akzeptierende Endzustände

NFA $\mathcal{A} = (\Sigma, Q, q_0, \Delta, A)$

Σ Alphabet der Kantensymbole

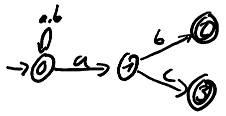
Q endl. nicht leere Zustandsmenge

$q_0 \in Q$ Startzustand

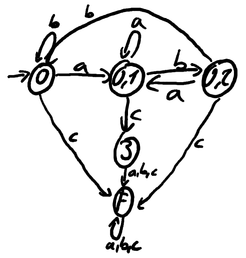
$\Delta \subseteq Q \times \Sigma \times Q$ Übergangsrelationen

$A \subseteq Q$ akzeptierende Endzustände

Potenzmengenkonstruktion

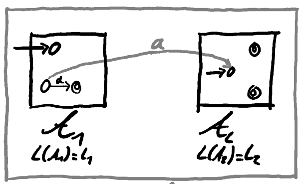


	a	b	c
0	0,1	0	/
0,1	0,1	0,2	3
0,2	0,1	0	/
3	/	/	/
/	/	/	/



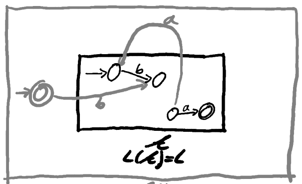
Produktautomat

Konkatination von Sprachen



$L(\mathcal{A}) = L_1 \cdot L_2$

Sternoperation



$L(\mathcal{A}^*) = L^*$

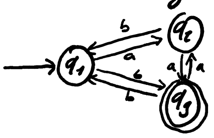
$\Rightarrow$ Abschlusseigenschaft der reg. Sprachen unter Konkat. und Stern

Satz von Kleene Auch Durchschnitt / Komplement von reg. Ausdrücken möglich, da bei Automaten möglich

Reguläre Sprache = Von endl. Automaten erkannte Sprache

= Es gibt einen reg. Ausdruck

Finden eines reg. Ausdrucks zu einem Automaten:



$\alpha^{(k)}_{\ell,m}$

ℓ: Von Zustand

m: Ziel Zustand

k: Über Zustände $\leq k$

$1 \leq \ell, m \leq n$

$0 \leq k \leq n$

n: Anz. Zustände

$$\alpha^{(k+1)}_{\ell,m} = \underbrace{\alpha^{(k)}_{\ell,m}}_{\text{Nur über alle Knoten}} + \underbrace{\alpha^{(k)}_{\ell,k+1}}_{\text{Zu neuem Knoten}} \underbrace{\left(\alpha^{(k)}_{k+1,k+1}\right)^*}_{\text{Ewige Zyklen}} \underbrace{\alpha^{(k)}_{k+1,m}}_{\text{Zum Ziel}}$$

→ Eintrag in … → Eintrag in … → Eintrag in … → Eintrag in …

Nur über alte Knoten → Eintrag in Zeile l, Spalte m, Iteration k

Zu neuem Knoten → Eintrag in k-ter Spalte

Einzige Zyklus → Eintrag in k-ter Diagonal-eintrag

Zum Ziel → Eintrag in Zeile k

$k+1=0$

Start \ Ziel	q_1	q_2	q_3
q_1	ε	a	b
q_2	b	ε	a
q_3	b	a	ε

$k+1=1$ Ich darf q_1 als Zwischenzustand nutzen

Start \ Ziel	q_1	q_2	q_3
q_1	$\varepsilon+\varepsilon\cdot(\varepsilon)^*\cdot\varepsilon$	$a+\varepsilon\cdot\varepsilon^*\cdot a$	$b+\varepsilon\cdot\varepsilon^* b$
q_2	$b+b\varepsilon^*\varepsilon$	$\varepsilon+b\cdot\varepsilon^* a$	$a+b\varepsilon^* b$
q_3	$b+b\varepsilon^*\varepsilon$	$a+b\varepsilon^* a$	$\varepsilon+b\varepsilon^* b$

$k+1=2$

Start \ Ziel	q_1	q_2	q_3
q_1	$\varepsilon+a\cdot(ba)^* b$	$a+a(ba)^*(ba)$	$b+a(ba)^*(a+bb)$
q_2	$b+(ba)(ba)^* b$	$ba+(ba)(ba)^*(ba)$	$a+bb+(ba)(ba)^*(a+bb)$
q_3	$b+(a+ba)(ba)^* b$	$a+ba+(a+ba)(ba)^*(ba)$	$bb+(ba)(ba)^*(a+bb)$

usw.

Lösung $\alpha_{1,3}^{(3)} = \alpha_{1,3}^{(2)} + \alpha_{1,3}^{(2)} (\alpha_{3,3}^{(2)})^* \alpha_{3,3}^{(2)}$

Startzust. Endzust. $= (b+a(ba)^*(a+bb))(bb+(a+ba)(ba^*)(a\ldots)$

Satz von Myhill-Nerode
Minimalautomat

$\sim_L$ hat endlichen Index gdw. L regulär

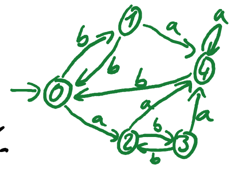
Dabei ist $w \sim_L w'$ gdw. $\forall x \in \Sigma^*: wx \in L \Leftrightarrow w'x \in L$

Die Zustände eines Automaten sind $\sim_A$ Äquivalenzklassen.

$\text{index}(\sim_A) \geq \text{index}(\sim_L)$ Bei Minimalautomat: gleich

Aufstellen von $\sim_L$-Äquivalenzklassen:

1) NFA → DFA



2)

0					
1	X				
2	X	X			
3	X	X			
4	X	X	X	X	
	0	1	2	3	4

- Tabelle mit allen Zuständen
- Unten links anfangen und zeilenweise nach oben: Zwei sind unterschiedlich, wenn
  ↳ Einer akzeptierend ist, der andere nicht
  ↳ Es eine Eingabe ist, so dass in zwei Zustände gelangt wird, die untersch. sind
  ⇒ Wenn untersch. Kreuz ins Feld
- So lange wdh. bis keine Änderung

/// Endzust. vs. kein Endzust.

/// Erste Iteration

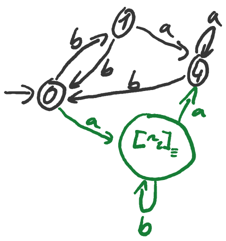
/// Zweite Iteration

3) Minimalautomat aufstellen

In dem Bsp. können 2 und 3 zusammengefasst werden.

Kontrolle: Beim Zusammenlegen stimmen die ausgehenden Kanten überein.

(Mit a kommt man immer zur 4)



Syntaktisches Monoid

Anstelle von $\sim_L$ betrachtet man $\approx_L$

$w \approx_L w'$ gdw. $\forall x,y \in \Sigma^*: xwy \in L \Leftrightarrow xw'y \in L$

Syntaktisches Monoid Anstelle von $\sim_L$ betrachtet man $\approx_L$

$w \approx_L w'$ gdw. $\forall x,y \in \Sigma^*: xwy \in L \Leftrightarrow xw'y \in L$

$\approx_L$ ist eine Verfeinerung von $\sim_L$: $x \approx_L y \Rightarrow x \sim_L y$

$\approx_L$ ist verträglich mit Konkatination $u \approx_L u' \wedge w \approx_L w' \Rightarrow uw \approx_L u'w'$

Pumping Lemma Nachweis, dass eine Sprache nicht regulär ist.

$\exists n \in \mathbb{N}\ \forall x \in L\ (|x| \geq n \Rightarrow \exists\, uvw : x = uvw$

$\wedge\ v \neq \varepsilon$

Muss auf reg. Sprachen zutreffen $\wedge\ |uv| \leq n$

$\forall m \in \mathbb{N}: uv^m w \in L$

Nicht-Regularität durch Kontraposition:

Zeige das $L = \{ w \in \{a,b\}^* : |w|_a = |w|_b \}$ nicht reg. ist.

Wir nehmen an, L sei regulär und wählen $n > 0$ beliebig

Dazu betrachten wir $x = \underbrace{aaa\ldots aa}_{n\text{-mal}} \underbrace{bbb\ldots bb}_{n\text{-mal}} = a^n b^n \in L$ Wort wählen

Es sei $x = uvw$ eine beliebige Zerlegung, so dass

$|uv| \leq n$ und $v \neq \varepsilon$

Dann muss die Zerlegung die Form

$u = a^x \quad v = a^y \quad w = a^{n-x-y} b^n$ mit $y \geq 1 \quad x \geq 0 \quad x+y \leq n$

Wir wählen $m = 2$ und erhalten

$uv^m w = uv^2 w = a^x a^{2y} a^{n-x-y} b^n = a^{n+y} b^n \notin L$ da $n+y \neq n$

Also kann das Wort nicht gepumpt werden und L ist nicht regulär.

Grammatik $G = (\Sigma, V, P, X_0)$

Σ : endl. Terminalalphabet ; $\Sigma \neq \emptyset$

V : endl. Variablenmenge ; $\Sigma \cap V = \emptyset$

$P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$: Produktionen Links kein leeres Wort

$X_0 \in V$: Startsymbol

Bsp. Klammerung: $G = (\{(,)\}, \{X\}, \{X \to (), X \to (X), X \to XX\}, X)$

$X \to XX \to X(X) \to ()(X) \to ()(())$ kontextfrei

Ableitbarkeit: x' ist aus x in einem Schritt ableitbar gdw. es eine Produktion gibt mit der man von x zu x' gelangt. x' ist aus x ableitbar gdw. es Produktionen gibt, so dass man aus x irgendwann x' erhält.

NFA zu Grammatik

$\mathcal{A} = (\Sigma, Q, q_0, \Delta, A)$

$G_A = (\Sigma, V, P, X_0)$

$V = \{X_q : q \in Q\}$

$P = \{X_q \to aX_{q'}$ für $(q,a,q') \in \Delta$

$X_q \to \varepsilon$ für $q \in A\}$

Backus-Naur-Form: Mehrere Produktionen zusammenfassen

$\left.\begin{array}{l} X \to a \\ X \to b \end{array}\right\} X \to a \mid b$

Erweiterte Backus-Naur-Form: Optionale Elemente

$\left.\begin{array}{l} X \to uvw \\ X \to uw \end{array}\right\} X \to u[v]w$

Chomsky-Hierarchie Typ | Name | Grammatik

$X \to uvw$
$X \to uw$ $\Big\}$ $X \to u[v]w$

Chomsky-Hierarchie

Typ	Name	Grammatik	
3	regulär	Produktionen rechtslinear	Das Nicht-Terminal immer weiter nach rechts schieben
		$X \to \varepsilon \mid a \mid aY$	DFA / NFA
2	kontextfrei	Variable zu irgendwas	
		$X \to v$	PDA → Startsymbol $\to \varepsilon$ Startsymbol nur am Anfang erreichbar
1	Kontextsensitiv	Nicht verkürzend. Nur harmlose ε-Prod.	
		$v \to v' \quad \lvert v\rvert \le \lvert v'\rvert$	DTM $\Rightarrow$ Wortproblem mit endl. Anz. Schritte lösen.
0	allgemein	Alles erlaubt	

Kontextfrei

Jede kontextfreie Grammatik ist äquivalent zu einer kontextfreien Grammatik mit maximal harmlosen ε-Produktionen.

Chomsky Normalform: Kontextfreie Grammatik der Form $X \to YZ$ und $X \to a$. Jede kontextfreie Grammatik ohne ε-Produktionen kann in die Chomsky Normalform gebracht werden.

→ Vorteil: Beim Ableiten entsteht ein Binärbaum

Abschlusseigenschaften:

Abgeschl. unter	$\cup$	$\cap$	$\overline{}$	$\cdot$	$*$	
reg. Sprachen	✓	✓	✓	✓	✓	DFA NFA
Kontextfreie Spr.	✓	✗	✗	✓	✓	PDA
Kontextsensitive Spr.	✓	✓	✓	✓	✓	DTM
allgem. Sprachen	✓	✓	✗	✓	✓	DTM (nur rek. aufz.) · Siehe TM: Halteproblem: $H \in$ Typ 0, $\overline{H} \notin$ Typ 0

Für $\cup$: Neues Startsymbol $X_0 \longrightarrow$ Startsymbol Gram. 1 | Startsymbol Gram. 2
Für $\cdot$: Neues Startsymbol $X_0 \longrightarrow$ Startsymbol Gram. 1 Startsymbol Gram. 2
Für $*$: Neues Startsymbol $X_0 \longrightarrow X_0X_0$ | Altes Startsymbol | ε

Kontextfrei

Pumping Lemma (kontextfrei)

Nachweis, dass eine Sprache nicht kontextfrei ist.

$\exists n\ \forall x \in L$ mit $\lvert x\rvert \ge n$ und $x = yuvwz$ wobei $uw \ne \varepsilon$ $\forall m$: $yu^m v w^m z \in L$

Zeige, dass $L = \{tt : t \in \Sigma^*\}$ nicht kontextfrei ist.

Wir nehmen an, L sei kontextfrei und wählen $n > 0$ beliebig.

Dazu betrachten wir $x = a^n b^n a^n b^n$.

Es sei $x = yuvwz$ eine beliebige Zerlegung mit $\lvert uvw\rvert \le n$ und $u, w \ne \varepsilon$

Dann muss für die Zerlegung gelten: $uvw = a^\alpha b^\beta$ oder $uvw = b^\alpha a^\beta$

mit $0 < \alpha + \beta \le n$ und $\alpha, \beta \ge 0$

Wir wählen $m = 2$. Dann können folgende Fälle eintreten:

$yu^m v w^m z = a^{n+\gamma} b^{n+\delta} a^n b^n$, oder

$yu^m v w^m z = a^n b^{n+\gamma} a^{n+\delta} b^n$, oder

$yu^m v w^m z = a^n b^n a^{n+\gamma} b^{n+\delta}$

mit $\gamma, \delta \ge 0$ und $\gamma + \delta \ge 2$ da $u, w \ne \varepsilon$

Keines dieser Wörter liegt in L. Also wird das Pumping-Lemma nicht erfüllt und L kann nicht kontextfrei sein.

CYK-Algorithmus

Ziel: Lösen des Wortproblems für kontextfreie Sprachen: $w \in L(G)$?

Voraussetzung: G als Grammatik in Chomsky Normalform

Idee: Welche Wörter kann ich aus welchen Variablen ableiten?

Beispiel: Ist "bacb" in folgender Sprache?

$X \to XY \mid ZZ \mid a$

Länge \ Zeichen	b	a	c	b

Beispiel: Ist "bacb" in folgender Sprache?

$X \to XY \mid ZZ \mid a$
$Y \to WZ \mid a$
$Z \to XY \mid b$
$W \to YW \mid c$

Länge \ Zeichen	b	a	c	b
1	Z	X,Y	W	Z
2	Teilwort ba /	Teilwort ac W ($\to$YW)	Teilwort cb Y ($\hookrightarrow$WZ)	
3	Teilwort bac /	Teilwort acb Y,X,Z		
4	Teilwort bacb X,...			

ac · b — aus W, aus Z
Kann ich WZ bekommen?
↳ Ja: Aus $Y \to WZ$

a · cb — aus X,Y, aus Y
Kann ich XY oder YY bekommen?
↳ Ja aus X oder Z

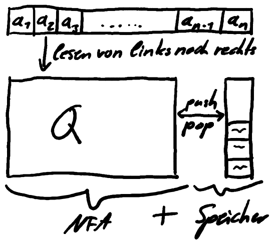
Da X das Startsymbol ist und ich aus X das Wort ableiten kann, ist das Wortproblem gelöst:

$X \to ZZ \to bZ \to bXY \to baY \to baWZ \to bacZ \to bacb$

Kellerautomat

Erkennt Typ-2 (kontextfreie) Sprachen

Idee: NFA + Kellerspeicher = PDA



Eingabe

$P = (\Sigma, Q, q_0, \Delta, A, \Gamma, \#)$
Σ: Eingabealphabet
Q: Endliche Zustandsmenge
$q_0 \in Q$: Startzustand
$\Delta \subseteq Q \times \Gamma \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \times Q$: Endl. Übergangsrelation
$A \subseteq Q$: Akzeptierende Zustände
Γ: Kelleralphabet
$\# \in \Gamma$: Anfangs-Kellersymbol

Definition

Konfiguration $C = (q, v, \alpha)$ q: Akt. Zust. v: Restwort α: Kellerinhalt

Der Automat akzeptiert in der Konfiguration $(q, \varepsilon, \varepsilon)$: $q \in A$

Die Startkonfiguration bei der Eingabe w ist: $C_0[w] = (q_0, w, \#)$

Bsp. Kellerautomat für die Klammersprache:

$P = (\{(,)\}, Q, q_0, \Delta, A, \Gamma, \#)$

$Q = \{q\}$ $q_0 = q$ $A = \{q\}$ $\Gamma = \{\mathrm{I}, \#\}$

$\Delta = \{ (q, \#, (, \mathrm{I}\#, q),$ Erste öffnende Klammer
$(q, \mathrm{I}, (, \mathrm{II}, q),$ Weitere öffnende Klammern
$(q, \mathrm{I},), \varepsilon, q),$ Schließende Klammern
$(q, \#, \varepsilon, \varepsilon, q) \}$ Am Ende Startsymbol entfernen

Jede kontextfreie Sprache wird von einem PDA erkannt.

↳ Geg: Kontextfreie Grammatik

↳ Lösung: PDA mit $Q = A = \{q\}$ $q_0 = q$ $\Gamma = V \cup \Sigma$ $\# = X_0$

Erstelle folgende Transitionen:

$(q, X, \varepsilon, \alpha, q)$ für jede Produktion $X \to \alpha$

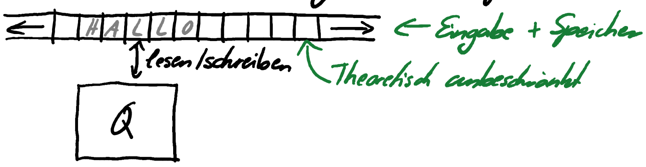
$(q, a, a, \varepsilon, q)$ für jedes $a \in \Sigma$

$(q, X, \varepsilon, \alpha, q)$ für jede Produktion $X \to \alpha$

$(q, a, a, \varepsilon, q)$ für jedes $a \in \Sigma$

Turingmaschine

Alles was berechenbar ist geht mit Turing



$M = (\Sigma, Q, q_0, \delta, q^+, q^-)$

Σ: Zeichenalphabet

Q: Zustandsmenge

$q_0 \in Q$: Startzustand

$\delta : Q \times (\Sigma \cup \{\square\}) \to (\Sigma \cup \{\square\}) \times \{<, \circ, >\} \times Q$

Akt. Zust. Akt. Bandsymb. Neues Symb. Kopfbewegt. Folgezustand

Übergangsfunktion: TM sind deterministisch

$q^+ \in Q$: Akzept. Endzustand

$q^- \in Q$: Verwerfender Endzustand

Konfiguration $C = (\alpha, q, x, \beta)$

α: Bandinhalt links vom Kopf

q: Akt. Zustand

x: Bandinhalt beim Kopf

β: Bandinhalt rechts vom Kopf

Startkonfiguration: $C_0[w] = (\varepsilon, q_0, \square, w)$

Die Turingmaschine startet links von der Eingabe.

Endkonfiguration $q \in \{q^+, q^-\}$

Akzeptierte Sprache: $\{w \in \Sigma^* : w \xrightarrow{M} q^+\}$ ← weniger als, da ohne q^-

Entscheidung des Wortproblems: $w \xrightarrow{M} \begin{cases} q^+ & \text{für } w \in L \\ q^- & \text{für } w \notin L \end{cases}$

Eine Sprache L ist aufzählbar gdw. sie von einer DTM akzeptiert wird.

↳ Wenn ich ein Wort der Sprache bekomme kann ich bestätigen, dass es in L liegt.

↳ " " " " nicht aus d. Sprache bekomme, kann ich nicht zwingend ausschließen, dass es nicht doch in L ist.

Eine Sprache L ist entscheidbar gdw. L von einer DTM entscheidbar ist.

↳ Ich kann zu jedem Wort immer klar sagen, ob es in L liegt, oder nicht.

↳ Die Sprachen von NFA und PDA sind immer entscheidbar

DFA aufstellen CYK-Algorithmus

Algorithmische Erzeugbarkeit $\Leftrightarrow$ Aufzählbarkeit einer DTM

Algorithmische Entscheidbarkeit $\Leftrightarrow$ Entscheidbarkeit einer DTM

Halteproblem: Wird eine DTM bei einer Eingabe w irgendwann nach q^+ od. q^- kommen?

$H = \{\langle M \rangle \in \Sigma^* : \langle M \rangle \xrightarrow{M} STOP\}$ $\langle M \rangle$ ist die Kodierung der DTM M.

Wende die Kodierte Maschine als Eingabe auf sich selbst an.

H ist nicht entscheidbar, aber aufzählbar

$\bar{H}$ ist " " und nicht "

Beweis: Wir nehmen an M_0 entscheidet H für alle M

$\langle M \rangle \xrightarrow{M_0} \begin{cases} q^+ & \text{falls } \langle M \rangle \xrightarrow{M} STOP \\ q^- & \text{falls } \langle M \rangle \xrightarrow{M} \infty \end{cases}$

Idee: Wir bauen M_1, die statt in q^+ zu gehen in eine Endlosschleife gelangt.

$\langle M \rangle \xrightarrow{M_1} \begin{cases} \infty & \text{falls } \langle M \rangle \xrightarrow{M} STOP \\ q^- & \text{falls } \langle M \rangle \xrightarrow{M} \infty \end{cases}$

q falls $\langle M \rangle \rightarrow \infty$

Idee: Wir bauen M_1, die statt in q^+ zu gehen in eine Endlosschleife gelangt.

$$\langle M \rangle \xrightarrow{M_1} \begin{cases} \infty & \text{falls } \langle M \rangle \xrightarrow{M} STOP \\ q^- & \text{falls } \langle M \rangle \xrightarrow{M} \infty \end{cases}$$

Daraus folgt für alle M: $\langle M \rangle \xrightarrow{M_1} \infty \iff \langle M \rangle \xrightarrow{M} STOP$

Für den Spezialfall $M := M_1$ $\langle M_1 \rangle \xrightarrow{M_1} \infty \iff \langle M_1 \rangle \xrightarrow{M_1} STOP$ ↯ Widerspruch

FGdI 2 - Aussagenlogik

Sonntag, 2. August 2015 14:25

Syntax und Semantik

Geg: Menge an aussagenlogischen Variablen V

AL(V) induktiv definiert über:

- $0, 1, p$ für $p \in V$ — Atome
- $\varphi \in AL(V) \Rightarrow \neg\varphi \in AL(V)$ — Negation
- $\varphi, \psi \in AL(V) \Rightarrow (\varphi \wedge \psi), (\varphi \vee \psi) \in AL(V)$ — Konjunktion, Disjunktion

Weitere Junktoren, wie $\rightarrow, \leftrightarrow, \ldots$ lassen sich immer auf obige Definition zurückführen.

Eine **V-Interpretation (Belegung)** ist eine Funktion $\mathfrak{J}: V \rightarrow \mathbb{B} : p \rightarrow \mathfrak{J}(p)$

Die Interpretation ist induktiv definiert.

- $0^{\mathfrak{J}} := 0 \quad 1^{\mathfrak{J}} := 1 \quad p^{\mathfrak{J}} := \mathfrak{J}(p)$ — Atome
- $(\neg\varphi)^{\mathfrak{J}} := 1 - \varphi^{\mathfrak{J}}$ — Negation
- $(\varphi \wedge \psi)^{\mathfrak{J}} := \min(\varphi^{\mathfrak{J}}, \psi^{\mathfrak{J}})$ — Konjunktion
- $(\varphi \vee \psi)^{\mathfrak{J}} := \max(\varphi^{\mathfrak{J}}, \psi^{\mathfrak{J}})$ — Disjunktion

Semantik: V-Interpret. $\mathfrak{J}$ und $\varphi \in AL(V)$

$\mathfrak{J}$ erfüllt $\varphi \Leftrightarrow \varphi^{\mathfrak{J}} = 1$ Man schreibt $\mathfrak{J} \models \varphi$

Für Formelmengen: $\mathfrak{J}$ muss <u>alle</u> Formeln erfüllen.

Semantische Begriffe

Folgerungsbeziehung: $\varphi \models \psi \Leftrightarrow$ aus φ kann ich ψ folgern $\Leftrightarrow \forall \mathfrak{J}: \mathfrak{J} \models \varphi \Rightarrow \mathfrak{J} \models \psi$

Allgemeingültigkeit: $\models \varphi \Leftrightarrow \varphi$ allgemeingültig $\Leftrightarrow \forall \mathfrak{J}: \mathfrak{J} \models \varphi$

Logische Äquivalenz: $\varphi \equiv \psi \Leftrightarrow \varphi$ und ψ sind log. äquiv. $\Leftrightarrow \forall \mathfrak{J}: \mathfrak{J} \models \varphi \Leftrightarrow \mathfrak{J} \models \psi$ $\quad p \equiv \neg\neg p$

Erfüllbarkeit: φ erfüllbar $\Leftrightarrow \exists \mathfrak{J}: \mathfrak{J} \models \varphi$ (Äquivalent für Formelmengen)

Funktionale Vollständigkeit: Zu jeder Funktion $f \in \mathbb{B}_n$ gibt es eine aussagenlog. Formel die äquiv. ist.

DNF: Disjunktion ($\vee$) von Konjunktionen ($\wedge$)

KNF: Konjunktion ($\wedge$) von Disjunktionen ($\vee$)

Zu jeder $\varphi \in AL_n$ gibt es eine äquiv. Formel φ_1 in DNF und φ_2 in KNF

Allerdings sind diese nicht immer gleich lang. (DNF länger?)

Vollständiges Junktorensystem: Alle Formeln können nur mit diesen Junktoren ausgedrückt werden.

Bsp.: $\{NAND\}, \{\wedge, \neg\}, \{\vee, \neg\}, \{\rightarrow, 0\}, \{\rightarrow, \neg\}$

Nachweis der nicht-vollständigkeit: Es ist nicht möglich ODER, UND oder NICHT darzustellen

Kompaktheitssatz: Für jede Formelmenge $\Phi \subseteq AL$ gilt: Φ erfüllbar $\Leftrightarrow$ jede endl. Teilmenge $\Phi_0 \subseteq \Phi$ erfüllbar.

Für jede Formelmenge $\Phi \subseteq AL$ und $\psi \in AL$ gilt: $\Phi \models \psi \Leftrightarrow$ Es gibt eine endl. Teilmenge Φ_0, so dass $\Phi_0 \models \psi$

$\Rightarrow$ Unerfüllbarkeit lässt sich bereits mit endl. Teilmenge nachweisen.

Lemma von König: Ein unendlicher, aber endlich verzweigter Baum muss einen unendlichen Pfad haben.

(Folgerung aus Kompaktheitssatz)

K-Färbbarkeit: Ein Graph ist k-färbbar gdw. jeder endliche Teil

(Folgerung aus Kompaktheitssatz)

AL-Resolution

(für Unerfüllbarkeit)

Logikkalkül: rein syntaktisches Format für Beweise

Kalkül: Regelsystem der Beweisführung

Ableitung: Anwendung der Regeln

Korrektheit: Die Ergebnisse des Kalküls sind immer richtig

Vollständigkeit: Jeder semantisch korrekte Sachverhalt ist formal beweisbar

Klausel: Menge von Literalen $\quad \square$ steht für die leere Klausel $\quad \square \equiv \bigvee\emptyset = 0$

Klauselmenge: Menge von Klauseln $\quad \bigwedge\emptyset = 1$

Klausel: Menge von Literalen $\square$ steht für die leere Klausel $\square \equiv \bigvee\emptyset = 0$

Klauselmenge: Menge von Klauseln $\bigwedge\emptyset = 1$

KNF: $(p \vee r) \wedge (q \vee r) \wedge (p \vee q)$
$K = \{ \{p,r\}, \{q,r\}, \{p,q\} \}$ } „Mind. zwei von p,q,r"

Resolutionskalkül: Je zwei Klauseln, in denen einmal ein Literal und in der anderen das negierte Literal vorkommt können zusammengefasst werden. Dabei werden Literal und die Negation entfernt. Es entsteht eine neue Klausel

$\{ \{p\}, \{q,r\}, \{\neg p, \neg q\}, \{\neg p, \neg r\} \}$

$\{\neg q\}$ $\{r, \neg p\}$ $\{\neg r\}$ $\{q, \neg p\}$

$\{\neg p\}$

$\square$

Es lässt sich die leere Klausel ableiten.
Also ist die Formel unerfüllbar

Das Resolutionskalkül ist korrekt, da sich die leere Klausel nur abl. lässt wenn d. Formel unerfüllbar ist.
Das Resolutionskalkül ist vollständig, da wenn die Formel unerfüllbar ist sich die leere Klausel abl. lässt.

⚠ Es dürfen nicht „2 Schritte auf einmal" gemacht werden.

$\{ \{p,q,r\}, \{\neg p, \neg q, r\} \}$ $\{ \{p,q,r\}, \{\neg p, \neg q, r\} \}$

$\{p, \neg p, r\}$ $\{q, \neg q, r\}$ ← Richtig $\{r\}$ ← FALSCH
$\equiv 1$ $\equiv 1$

Hornklauseln: Klauseln mit max einem pos. Literal

Positive Hornklausel: C besteht nur aus einem pos. Literal $C = \{q\}$

Negative Hornklausel: C besteht nur aus neg. Literalen $C = \{\neg p_1, \neg p_2, \dots, \neg p_r\}$

Vorteil: $C = \{\neg p_1, \neg p_2, \dots, \neg p_r, q\} \iff (p_1 \wedge p_2 \wedge \dots \wedge p_r) \rightarrow q$

Polynomieller Erfüllbarkeitstest

AL Sequenzenkalkül
(für Allgemeingültigkeit)

Sequenz: $\Gamma \vdash \Delta$ Gamma leitet Delta ab.
Wenn alle Formeln in Γ wahr sind, dann muss mind. eine

Eine Sequenz heißt **allgemeingültig** gdw. $\bigwedge\Gamma \models \bigvee\Delta$

$\varphi \vdash \psi$ allgemeingültig gdw. $\varphi \models \psi$
$\emptyset \vdash \psi$ allgemeingültig gdw. ψ allgemeingültig
$\varphi \vdash \emptyset$ allgemeingültig gdw. φ unerfüllbar

Sequenzenkalkül: Regeln zur Erzeugung allgemeingültiger Sequenzen

Prämissensequenz ← Voraussetzung: Diese müssen gelten
Konklusionssequenz ← Wenn Voraussetzung gilt, gilt auch das

$$\frac{\;\_\_ \vdash \_\_\;}{\_\_ \vdash \_\_}$$
UND $\bigwedge$ ODER $\bigvee$

(Ax)	$\dfrac{}{\Gamma, \varphi \vdash \Delta, \varphi}$		Prämissenfrei
(0-Ax)	$\dfrac{}{\Gamma, 0 \vdash \Delta}$	(1-Ax)	$\dfrac{}{\Gamma \vdash \Delta, 1}$
$(\neg L)$	$\dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma, \neg\varphi \vdash \Delta}$	$(\neg R)$	$\dfrac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi}$
$(\vee L)$	$\dfrac{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}$	$(\vee R)$	$\dfrac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}$
$(\wedge L)$	$\dfrac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}$	$(\wedge R)$	$\dfrac{\Gamma \vdash \Delta, \varphi \quad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \wedge \psi}$

$$\frac{\Gamma \vdash \varphi \qquad \Gamma', \varphi \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}$$
(modus ponens)

$$\dfrac{\dfrac{}{p \vdash p, \neg q}\,(Ax) \qquad \dfrac{\dfrac{}{p, q \vdash q}\,(Ax)}{p \vdash q, \neg q}\,(\neg R)}{p \vdash (p \wedge q), \neg q}\,(\wedge R)$$

Problem

$$\dfrac{\dfrac{}{p \vdash p}\,(Ax) \quad p \vdash q}{p \vdash p \wedge q}\,(\wedge R) \qquad \dfrac{q \vdash p \quad \dfrac{}{q \vdash q}\,(Ax)}{q \vdash p \wedge q}\,(\wedge R)$$

$$\dfrac{\dfrac{\dfrac{}{p \vdash p, \neg q}\,(Ax) \qquad \dfrac{\cdots}{p \vdash q, \neg q}\,(\neg R)}{p \vdash (p \wedge q), \neg q}\,(\wedge R)}{p \vdash (p \wedge q) \vee \neg q}\,(\vee R)$$

$\Rightarrow$ allgemeingültig

$$\dfrac{\dfrac{\dfrac{}{p \vdash p}\,(Ax) \quad p \vdash q}{p \vdash p \wedge q}\,(\wedge R) \qquad \dfrac{q \vdash p \quad \dfrac{}{q \vdash q}\,(Ax)}{q \vdash p \wedge q}\,(\wedge R)}{p \vee q \vdash p \wedge q}\,(\vee L)$$

$\Rightarrow$ nicht allgemeingültig

Korrektheit von Regeln: $\dfrac{\ldots\ (A)}{\ldots\ (B)}$

Zu zeigen: (A) allgemeingültig $\Rightarrow$ (B) allgemeingültig

Sei J eine Interpretation, die ..... erfüllt, dann wahr ...

Sei J eine Interpretation, die .... nicht erfüllt, dann wahr ...

Im AL-Sequenzenkalkül ist die Reihenfolge der Regelanwendung irrelevant

FGdI 2 - Logik erster Stufe

Donnerstag, 23. Juli 2015 13:04

Strukturen und Belegungen

Gegenstand: S-Strukturen mit Belegung für Variablen

Ausdrucksmögl.: Atomare Aussagen über Terme, Funktionen, Konstanten, Variablen

$\wedge, \vee, \neg$

Quantifizierung $\forall, \exists$

Signatur: Auswahl an Konstanten, Fnkt. und Relationssymb. mit def. Stelligkeit

Struktur: $\mathcal{A} = (A, c^{\mathcal{A}}, \dots, f^{\mathcal{A}}, \dots, R^{\mathcal{A}}, \dots)$ zur Signatur S

A: Trägermenge $(A \neq \emptyset)$

$c^{\mathcal{A}}$: Ausgezeichnetes Element für $c \in S$

$f^{\mathcal{A}}: A^n \to A$ für n-stelliges Funktionssymbol $f \in S$

$R^{\mathcal{A}} \subseteq A^n$ für n-stelliges Relationssymbol $R \in S$

Beispiel: Signatur $S = (+, \cdot, \leq, 0, 1)$ — Nur ein Symbol

Struktur $\mathcal{N} = (\mathbb{N}, +^{\mathcal{N}}, \cdot^{\mathcal{N}}, \leq^{\mathcal{N}}, 0^{\mathcal{N}}, 1^{\mathcal{N}})$ — Konkrete Definition als Addition d. nat. Zahlen

Terme: Sei S Signatur und $S_F \subseteq S$ der funktionale Anteil (nur Funktions- und Konstantensymbole)

Terme sind geschachtelte Funktionsausdrücke aus Variablen, Konstanten und Funktionen

Induktive Definition:

Es kommt immer ein Term raus

$x \in T(S)$ für jede Variable $x \in V$

$c \in T(S)$ für jedes Konstantensymbol $c \in S$

Wenn $t_1, \dots, t_n \in T(S)$, dann $f t_1 \dots t_n \in T(S)$ für f n-stellig fc fcc

$T_n(S)$: Menge der Terme in denen nur Variablen $x_1, \dots, x_n$ vorkommen.

$T_0(S)$: Variablenfreie Terme

Belegung: Wenn ich in S auch Variablen habe, dann muss ich sie belegen können.

Die Funktion $\beta: V \to A$ heißt Belegung für die $x \in V$ in der Struktur $\mathcal{A} = (A, \dots)$

Eine S-Struktur und Belegung β zusammen bilden eine S-Interpretation $\mathfrak{J} = (\mathcal{A}, \beta)$

Induktive Definition der Interpretation für Terme

- Für $t = x$, $x \in V$: $t^{\mathfrak{J}} := \beta(x)$
- Für $t = c$, $c \in S$ Konst.symb.: $t^{\mathfrak{J}} := c^{\mathcal{A}}$
- Für $t = f t_1 \dots t_n$, $f \in S$: $t^{\mathfrak{J}} := f t_1^{\mathfrak{J}} \dots t_n^{\mathfrak{J}}$

Die Semantik von Termen ist über die Belegung d. Variablen definiert.

Herbrandstruktur: Die Menge aller S-Terme $\{c, fc, ffc, fffc, \dots\}$

Syntax u. Semantik von FO

Induktive Definition der Menge der FO

- $t_1 = t_2 \in FO(S)$ — Gleichheit
- $R t_1 t_2 \dots t_n \in FO(S)$ — n-st. Relationen
- $\varphi \in FO(S) \Rightarrow \neg\varphi \in FO(S)$ — Negation
- $\varphi, \psi \in FO(S) \Rightarrow (\varphi \vee \psi), (\varphi \wedge \psi) \in FO(S)$ — Konjunktion, Disjunktion
- $\varphi \in FO(S), x \in V \Rightarrow \exists x \varphi \in FO(S)$ — Existenzquantor
  $\Rightarrow \forall x \varphi \in FO(S)$ — Allquantor

φ sind „Elemente" KEINE Mengen (das wäre 2. Stufe)

Freie Variablen sind nicht durch Quantoren gebunden

$frei(\varphi) \subseteq V$ für $\varphi \in FO(S)$

Eine Formel ohne freie Variablen heißt Satz

Quantorenrang $qr(\varphi)$: Wie viele Quantoren in der FO vorkommen

⚠ $qr(\underbrace{\forall x}_{1} \underline{\exists y}: fx = y) = 2$

Quantorenrang $qr(\varphi)$: Wie viele Quantoren in der FO vorkommen

⚠ $qr(\underbrace{\forall x}_{1}\underbrace{\exists y}_{2}: fx = y) = 2$

$qr(\underbrace{\forall x}_{1}(\underbrace{\exists y}_{2}: \underbrace{fx = y \lor}_{\text{gl. Level}} \underbrace{\exists z}_{2}: gx = z)) = 2$ und NICHT 3

Eine Formel vom Quantorenrang 0 heißt quantorenfrei

Die Semantik einer FO basiert auf S-Interpretationen. Es ist ein boolescher Wert

Sei φ eine FO(S) und $\mathfrak{J} = (\mathcal{A}, \beta)$ eine S-Interpretation.

Man sagt: $\mathfrak{J}$ ist ein Modell von φ gdw. $\varphi^{\mathfrak{J}} = 1$ $\quad (\mathfrak{J} \models \varphi)$

Mit dieser Interpretation kann ich φ wahr machen. Also ist es ein Modell

Man sagt: $\mathcal{A} \models \varphi[a_1, \dots, a_n]$ gdw. $\left[\begin{array}{l} (\mathcal{A}, \beta) \models \varphi \text{ für ein/alle } \beta \text{ mit} \\ \beta(x_i) = a_i \text{ für } i = 1, \dots, n \end{array}\right]$

Folgerungsbeziehung: $\varphi \models \psi$ gdw. $\forall \mathfrak{J}: \mathfrak{J} \models \varphi \Rightarrow \mathfrak{J} \models \psi$ $\qquad$ $\mathfrak{J}$ ist immer eine S-Interpretation

Allgemeingültigkeit: $\varphi \in FO(S)$ allgemeingültig gdw. $\forall \mathfrak{J}: \mathfrak{J} \models \varphi$

Logische Äquivalenz: $\varphi, \psi \in FO(S)$ log. äquivalent gdw. $\forall \mathfrak{J}: \mathfrak{J} \models \varphi \Leftrightarrow \mathfrak{J} \models \psi$

Semantik Spiel: Spieler: Verifizierer, Falsifizierer

$\qquad\qquad\qquad\qquad\downarrow\qquad\qquad\downarrow$

$\qquad\qquad\qquad\qquad\exists\qquad\qquad\forall$

Gleichheit ist egal! $FO^{\neq}(S) \leq FO(S)$

Man stellt eine Relation $\sim$ für die Gleichheit auf. $\sim$ ist eine Äquivalenzrelation

Pränexe Normalform

Pränexe Normalform: Alle Quantoren stehen vorne

Zu jeder FO gibt es eine FO in pränexer Normalform

$\exists y \forall x\, Exy \lor \neg\exists y Exy \equiv \exists y_1 \forall y_2 \forall y_3 (E y_2 y_1 \lor E x y_3)$

Regeln für die Bewegung:

$\forall x\, \varphi \land \forall y\, \psi \equiv \forall x \forall y (\varphi \land \psi)$ $\quad$ wenn $x \notin frei(\psi)$ und $y \notin frei(\varphi)$, sonst umbenennen.

$\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$

$\forall x\, \varphi(x) \to \forall y\, \psi(y) \equiv \exists x \forall y (\varphi(x) \to \psi(y))$

Substitution

$\underbrace{\mathfrak{J} \models \varphi(t/x)}_{\text{Neue Formel}} \Leftrightarrow \underbrace{\mathfrak{J}[x \mapsto t^{\mathfrak{J}}] \models \varphi}_{\text{Alte Formel}}$

Man kann x nicht einfach durch t ersetzen (woanders vorkommen; frei-gebunden)

(t/x) t als Substitution für x

Skolemisierung

Skolemnormalform: Keine $\exists$; Nur $\forall$ ganz vorne

Vorgehen: Für jedes $\exists$ eine Funktion einführen. Parameter: Alle anderen Werte

Die Skolemnormalform ist erfüllbarkeitsäquivalent. NICHT logisch äquivalent

$\psi = \forall x \exists y\, \varphi(x, y)$ erfüllbar $\Leftrightarrow$ $\psi' = \forall x\, \varphi(x, fx)$ erfüllbar

Herbrand

Ziel: Erfüllbarkeit von universellen, gleichheitsfreien Sätzen

Für jede $FO_0^{\neq}$ Formelmenge Φ gilt: Φ erfüllbar $\Leftrightarrow$ Es gibt Herbrand Modell $H = (\underbrace{T_0(S)}_{\text{Gleichheitsfrei}}, \underbrace{(R^H)_{R\in S}}_{\text{„Richtige" Relationen}}) \models \Phi$

$T_0(S)$: Menge aller mögl. Terme $\{c, fc, ffc, fffc, \dots\}$

Herbrand Modell: Sei $\mathcal{A} = T_0(S)$

$\left.\begin{array}{l} c^{\mathcal{A}} = c \quad \text{Konstante } c \text{ wird als } c \text{ interpretiert} \\ f^{\mathcal{A}}(t_1, t_2) \mapsto f(t_1, t_2) \end{array}\right\}$ Herbrand Modell

Aufstellen eines Herbrand Modells:

- Geg: Formelmenge mit Formeln in Skolemnormalform
- Ziel: Finden eines Herbrandmodells, so dass alle erfüllt werden.
- Zu bestimmen: Eine Trägermenge mit allen benutzten Elementen

$\quad\hookrightarrow T = \{c, fc, ffc, \dots\}$

· Zu bestimmen: eine Trägermenge mit allen benötigten Elementen
↳ $T = \{c, fc, ffc, \ldots\}$
Für jede Relation in den Formeln welche Elemente das sind
↳ Für 2-stellige Relat. R: $R^H = T \times T$ oder $R^H = \{(c,c), (c,fc), (c,ffc), \ldots\}$
↳ Für 1-stellige Relat. P: $P^H = T$ oder $P^H = \{c\}$
· Definieren des Herbrandmodells mit Trägermenge und Relationsdefinitionen
↳ $H = (T, R^H, P^H)$

Erfüllbarkeit
↳ unentscheidbar

Ziel: Ist $\Phi \subseteq FO(S)$ erfüllbar?

$\Phi \subseteq FO(S)$ bel. Formelmenge
↓ freie Var. → Konst.
$\Phi' \subseteq FO_0(S')$ Satzmenge: freie Variablen durch Konstanten
↓ Gleichheit als Relation
$\Phi'' \subseteq FO_0^{\neq}(S'')$ Gleichheitsfreie Satzmenge
↓ Skolemisierung
$\Phi''' \subseteq FO_0^{\forall}(S''')$ Universell-pränexe Formelmenge
↓ Herbrandmodell suchen
AL

Für Erfüllbarkeitsfragen kann FO auf AL reduziert werden

Kompaktheitssatz: Formelmenge Φ erfüllbar $\Leftrightarrow$ Jede endl. Teilmenge $\Phi_0 \subseteq \Phi$ erfüllbar (= Endlichkeitssatz)

Jedes endl. $\Phi_0 \subseteq \Phi$ erfüllbar $\Longrightarrow$ Φ erfüllbar	FO
⇓ ⇑	
Für jedes endl. Φ_0 ist Φ_0^{AL} erfüllbar $\Longrightarrow$ Jede endl. Teilmenge $\Psi_0 \subseteq \Phi^{AL}$ erfüllbar $\xrightarrow[\text{satz}]{\text{AL-Komp.-}}$ Φ^{AL} erfüllbar	AL

Unerfüllbarkeit

Eine Formelmenge ist unerfüllbar gdw. eine endl. Teilmenge unerfüllbar ist.

Resolutionskalkül: Grundinstanzresolution

Gegenstand: $FO^{\neq}$ Klauselmenge K (universell skolemisierte $FO^{\neq}$ Sätze $\forall x_1 \forall x_2 \ldots \forall x_n$ {)

Ziel: Ableiten der leeren Klausel □

Korrektheit: □ ableitbar aus K ⇒ K unerfüllbar

Vollständigkeit: K unerfüllbar ⇒ □ aus K ableitbar

Vorgehen:

1. Skolemnormalform aufstellen — $\varphi_1 = \forall x \forall y\, (Rxy \to (Qx \Leftrightarrow \neg Qy))$ $\quad \varphi_2 = \forall x \forall y\, Rxy$
2. Relevante Atome bestimmen und ersetzen — $\alpha = Rxy \quad \beta_1 = Qx \quad \beta_2 = Qy$ $\quad \gamma = Rxy$
3. Kern in KNF — $\underline{(\neg\alpha \vee \beta_1 \vee \neg\beta_2)} \wedge (\neg\alpha \vee \beta_1 \vee \beta_2) \wedge (\neg\alpha \vee \neg\beta_1 \vee \neg\beta_2) \wedge \underline{(\neg\alpha \vee \neg\beta_1 \vee \beta_2)}$ (erste und letzte Klausel: $\equiv 1$)
4. Klauseln aufstellen — $K(\varphi) = \{\{\neg\alpha, \beta_1, \beta_2\}, \{\neg\alpha, \neg\beta_1, \neg\beta_2\}\}$ $\quad K(\varphi_2) = \{\{\gamma\}\}$
5. Resolution wie bei AL nur denen — $= \{\{\neg Rxy, Qx, Qy\}, \{\neg Rxy, \neg Qx, \neg Qy\}\}$ $\quad = \{\{Rxy\}\}$

Idee: Es gibt ein Problem bei $x = y = c$

$\{\neg Rcc, Qc\}$, $\{\neg Rcc, \neg Qc\}$ $\quad \{Rcc\}$

$\{\neg Rcc\}$

□

K unerfüllbar $\Leftrightarrow$ $GI(K)$ unerfüllbar $\Leftrightarrow$ □ $\in Res^*(GI(K))$

Resolutionssatz: Für $FO^{\neq}(S)$ Klauselmengen K gilt: K unerfüllbar $\Leftrightarrow$ □ $\in Res^*(K)$

Allgemeingültigkeit
↳ rek. aufzählbar
↳ semi-entscheidbar

Sequenzenkalkül: Ziel: Allgemeingültige FO Formeln

Eine Sequenz ist ein Paar von endl. FO-Satzmengen $\Gamma \vdash \Delta$

Eine Sequenz ist allgemeingültig gdw. $\bigwedge \Gamma \models \bigvee \Delta$

$\emptyset \vdash \{\varphi\}$ gdw. φ allgemeingültig

↳ semi-entscheidbar

Eine Sequenz ist allgemeingültig gdw. $\bigwedge\Gamma \models \bigvee\Delta$

$\emptyset \vdash \{\varphi\}$ gdw. φ allgemeingültig

$\Gamma \vdash \emptyset$ gdw. Γ unerfüllbar

AL

(Ax) $\dfrac{}{\Gamma, \varphi \vdash \Delta, \varphi}$

(¬L) $\dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma, \neg\varphi \vdash \Delta}$ (¬R) $\dfrac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi}$

(∨L) $\dfrac{\Gamma, \varphi \vdash \Delta \qquad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}$ (∨R) $\dfrac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}$

(∧L) $\dfrac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}$ (∧R) $\dfrac{\Gamma \vdash \Delta, \varphi \qquad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \wedge \psi}$

∀/∃

Ein Term t eingesetzt für die Variable x

Bereits das konkrete t macht wahr

Also erst recht für alle

(∀L) $\dfrac{\Gamma, \varphi(t/x) \vdash \Delta}{\Gamma, \forall x \varphi(x) \vdash \Delta}$ (∀R) $\dfrac{\Gamma \vdash \Delta, \varphi(c/x)}{\Gamma \vdash \Delta, \forall x \varphi(x)}$ falls c nicht in $\Gamma, \Delta, \varphi(x)$

(∃L) $\dfrac{\Gamma, \varphi(c/x) \vdash \Delta}{\Gamma, \exists x \varphi(x) \vdash \Delta}$ falls c nicht in $\Gamma, \Delta, \varphi(x)$ (∃R) $\dfrac{\Gamma \vdash \Delta, \varphi(t/x)}{\Gamma \vdash \Delta, \exists x \varphi(x)}$

=/Sub

(=) $\dfrac{\Gamma, t = t \vdash \Delta}{\Gamma \vdash \Delta}$

(Sub-L) $\dfrac{\Gamma, \varphi(t/x) \vdash \Delta}{\Gamma, t = t', \varphi(t'/x) \vdash \Delta}$ (Sub-R) $\dfrac{\Gamma \vdash \Delta, \varphi(t/x)}{\Gamma, t = t' \vdash \Delta, \varphi(t'/x)}$

und analoge Regeln mit $t' = t$ statt $t = t'$

Benötigte Änderung, damit in beide Richtungen

(∀L)' $\dfrac{\Gamma', \underline{\forall x \varphi(x)}, \varphi(t/x) \vdash \Delta}{\Gamma', \forall x \varphi(x) \vdash \Delta}$

(∃R)' $\dfrac{\Gamma \vdash \Delta, \underline{\exists x \varphi(x)}, \varphi(t/x)}{\Gamma \vdash \Delta, \exists x \varphi(x)}$

Modus ponens: $\dfrac{\Gamma \vdash \varphi \qquad \Gamma', \varphi \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}$

Kontr. $\dfrac{\Gamma \vdash \varphi \qquad \Gamma' \vdash \neg\varphi}{\Gamma, \Gamma' \vdash \emptyset}$

Bei dem Sequenzenkalkül muss auf die Reihenfolge geachtet werden.
Wenn man dumm ist bekommt man nichts raus.

$$\begin{array}{ll}
\exists y\, fy = c \vdash \exists y\, ffy = c & \text{offensichtlich nicht allgemeingültig} \\ \hline
\forall x \exists y\, fy = x \vdash \exists y\, ffy = c & \forall L \\ \hline
\forall x \exists y\, fy = x \vdash \forall x \exists y\, ffy = x & \forall R \quad \text{erlaubt, da } c \text{ noch nicht vorhanden}
\end{array}$$

Kein Ergebnis

$$\begin{array}{ll}
\hline
fb = c \vdash fb = c & Ax \\ \hline
fa = b,\ fb = c \vdash ffa = c & SubR \\ \hline
fa = b,\ fb = c \vdash \exists y\, ffy = c & \exists R \\ \hline
\exists y\, fy = b,\ fb = c \vdash \exists y\, ffy = c & \exists L \\ \hline
\forall x \exists y\, fy = x,\ fb = c \vdash \exists y\, ffy = c & \forall L : x \text{ zu } b \\ \hline
\exists y\, fy = c,\ \forall x \exists y\, fy = x \vdash \exists y\, ffy = c & \exists L : y \text{ zu } b \\ \hline
\forall x \exists y\, fx = y \vdash \exists y\, ffy = c & \\ \hline
\forall x \exists y\, fx = y \vdash \forall x \exists y\, ffy = x & \forall R
\end{array}$$

Idee:

$\underbrace{\forall x \exists y\, fy = x}^{f \text{ surjektiv}} \vdash \underbrace{\forall x \exists y\, ffy = x}^{\text{2-fache Anwendung: Immer noch surjektiv}}$

$a \xrightarrow{f} b \xrightarrow{f} c$

Ableitbar: Sei $\Phi \subseteq FO_0(S)$ eine Satzmenge. $\varphi \in FO_0(S)$ heißt ableitbar aus Φ gdw.

$\exists \Gamma_0 \in \Phi : \Gamma_0 \vdash \varphi$

Man schreibt auch $\Phi \vdash \varphi$

Gödelscher Vollständigkeitssatz: $\Phi \vdash \varphi \Leftrightarrow \Phi \models \varphi$

Φ erfüllt φ ⇔ Aus Φ folgt φ

Konsistenz ⇔ Erfüllbarkeit

⇒ Da man alle ableitbaren Sequenzen rek. aufzählen kann, sind

Konsistenz ⟺ Erfüllbarkeit

⟹ Da man alle ableitbaren Sequenzen rek. aufzählen kann, sind auch die allgemeingültigen Sequenzen rek. aufzählbar.

Konsistenz: Eine Formelmenge Φ heißt konsistent wenn für kein $\Gamma \subseteq \Phi$ die Sequenz $\Gamma \vdash \emptyset$ ableitbar ist.

Hintikka Konstruktion: Zu zeigen: aus Φ lässt sich nicht $\Gamma \vdash \Delta$ ableiten.

Idee: Zeigen, dass $\Phi \cup \Gamma \cup \Delta^{\neg}$ erfüllbar. $\Delta^{\neg} := \{\neg\psi : \psi \in \Delta\}$ Es gilt nicht: $\Phi \models \bigwedge\Gamma \to \bigvee\Delta$

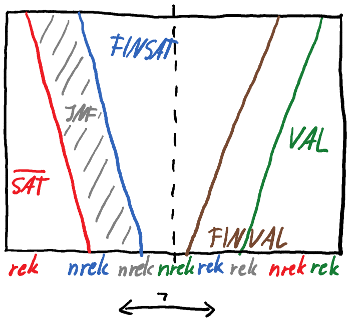
Man findet Modelle einer induktiv geeignet gewählten Obermenge von $\Phi \cup \Gamma \cup \Delta^{\neg}$ (→Hintikkamenge)

Henkin Konstruktion: Zeige: Φ konsistent ⟹ Φ erfüllbar

Idee: $\Phi \cup \{\varphi\}$ konsistent gdw. nicht $\Phi \vdash \neg\varphi$, da $\Phi \vdash \neg\varphi$ bedeutet, dass für ein $\Gamma \subseteq \Phi$ die Sequenz $\Gamma \vdash \neg\varphi$ ableitbar ist. Also ergibt sich $\Gamma, \varphi \vdash \emptyset$; also $\Phi \cup \{\varphi\}$ nicht konsistent.

$$\dfrac{\dfrac{\Gamma \vdash \neg\varphi \quad \overline{\Gamma, \neg\varphi, \varphi \vdash \neg\varphi}\;Ax}{\Gamma, \varphi \vdash \neg\varphi}\;\text{mod. pon.} \qquad \overline{\Gamma, \varphi \vdash \varphi}\;Ax}{\Gamma, \varphi \vdash \emptyset}\;\text{Kontr.}$$

Unentscheidbarkeit



SAT: Erfüllbar in FO

FINSAT: Erfüllbar mit endl. Modellen in FO (es gibt welche, nicht nur)

INF: Erfüllbar und nur endl. Modelle

FINVAL: Allgemeingültig mit endl. Modellen in FO

VAL: Allgemeingültig in FO

Beispiel: Unentscheidbarkeit von SAT(FO(S)) durch Reduktion des Halteproblems:

$M, w \mapsto \varphi_{M,w} \in FO(S_M)$ — $\varphi_{M,w}$ erfüllbar gdw. $w \xrightarrow{M} \infty$

TM Eingabe — beschreibt Lauf von M auf w — M terminiert nicht bei w

$M = (\Sigma, Q, q_0, \delta, q^+, q^-)$

$S_M = (succ, pred, 0, R_a, Z_q, K)$

- succ, pred: Schritt der TM auf Band, 1-stellig
- 0: Startpos Konst.
- R_a: a steht zum Zeitpunkt $t \in$ an Stelle i auf d. Band (1 pro Zeichen) 2-stellig
- Z_q: Ob zum Zeitpunkt t in Zustand q (1 pro Zust.) 1-stellig
- K: Zum Zeitpunkt t steht Kopf an Zelle i des Bands 2-stellig

Satz von Church Turing: Wenn SAT(FO(S)) entscheidbar wäre müsste man das Halteproblem entscheiden können

Satz von Trakhtenbrot: FINSAT(FO(S)) nicht entscheidbar → $\overline{\text{FINSAT(FO(S))}}$ nicht rek. aufzbar.

Satz von Tarski: Sei $\mathcal{N} = (\mathbb{N}, +, \cdot, 0, 1, <)$ Dann ist $Th(\mathcal{N}) := \{\varphi \in FO_0 : \mathcal{N} \models \varphi\}$ unentscheidbar (noch nicht mal rek. aufz.)

$Th(\mathcal{N})$: Theorie der Arithmetik

Entscheidbare Teilklassen: FO allgemein unentscheidbare Erfüllbarkeit; Wenn Formel in pränexer Normalform und als Quantoren nur $\forall^*$, $\exists^*$, oder $\exists^*\forall\forall\exists^*_{\neq}$ hat, dann entscheidbar.

FO-Theorie: $Th(\mathcal{A}) := \{\varphi \in FO_0(S) : \mathcal{A} \models \varphi\}$ wobei $\mathcal{A}$ eine S-Struktur ist.

Entscheidbarkeit von Theorien

entscheidbar	unentscheidbar
MSO-Theorie von Bäumen (Rabin)	Graphentheorie FO
FO-Th $(\mathbb{R}, +, \cdot, 0, 1, <)$ (Tarski)	FO-Th $(\mathbb{N}, +, \cdot, 0, 1, <)$
FO-Th $(\mathbb{N}, +, 0, 1, <)$ (Presburger)	
FO-Theorien abelscher Gruppen	Gruppentheorie FO

Problem: Bei $\mathbb{N}$ kann man nicht sagen, ob es evtl. noch eine Zahl ganz weit weg gibt. Also gilt Induktionsbeweis nicht.

Ausdrucksstärke:

[illegible]

FO-Theorien abelscher Gruppen | Gruppentheorie FO

Ausdrucksstärke:

In FO kann man nicht ausdrücken:

- Endlichkeit der Trägermenge
- Zusammenhang von (endlichen) Graphen / Erreichbarkeit in Graphen
- gerade Länge endlicher linearer Ordnungen

(- für die Definition einer lin. Ordnung braucht man zwei Variablen)

Ehrenfeucht-Fraïssé-Spiel: Unentscheidbarkeit von Strukturen

↳ Spieler eins: Strukturen unterschiedlich

↳ Spieler zwei: Strukturen ähnlich

Geg: Feste endl. relationale Signatur S

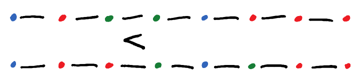
Idee: 1 markiert Element in W oder W'

2 antwortet in jeweils anderer Struktur, dabei muss stets lokale Isomorphie ($\equiv_0$) gewährleistet sein.

2 verliert wenn $W,m \not\equiv_0 W'm'$

2 Gewinnstrategie in $G^q(W,m,W',m')$ gdw. $W,m \equiv_q W'm'$ Spiel d. Länge q

Bsp.:



Idee für Unterschied:

- 2 grüne hinter einander: $\exists x \exists y (x<y \wedge \forall z(z<x \vee z>y) \wedge Gx \wedge Gy)$
  Quantorenrang 3: Also gewinnt eins in drei Runden
- Es gibt einen grünen, so dass alle blauen kleiner sind: $\exists x(Gx \wedge \forall y(By \rightarrow y<x))$
  Quantorenrang 2: Also gewinnt eins in zwei Runden
- Es gibt nichts mit Quantorenrang eins.

Andere Logiken

Monadische Logik zweiter Stufe: Quantifizierung über Mengen erlaubt

↳ Mengen mit Großbuchstabe benannt

↳ Xy heißt: Die Menge X enthält das Element y

Bsp.: Linear Ordnung ungerader Länge

$\varphi = \exists X(\exists x(\psi_{min}(x) \wedge Xx) \wedge \exists x(\psi_{max}(x) \wedge Xx) \wedge \forall x \forall y(\psi_{next}(x,y) \rightarrow (Xx \leftrightarrow \neg Xy)))$

ψ_{min}: erstes Element der Ordnung ψ_{max}: letztes Element ψ_{next}: auf x folgt direkt y

Modallogik (ML): Teillogik von FO über Transitionssysteme.

Transitionssystem: Struktur $\mathcal{Q} = (Q, (E_a)_{a\in\Sigma}, P_1, \dots, P_s)$ Q ist Zustandsmenge $E \subseteq Q \times Q$: Kanten P: Atomare Eigenschaften