

GBI Definitionen

Inschrift $\xrightarrow{\text{speichert}}$ Nachricht $\xrightarrow{\text{Bedeutung}}$ Information (erfordert Interpretation)

Alphabet = endliche Menge von Symbolen

Eigenschaften von Relationen: linkstotal, rechtstotal, linkseindeutig, rechtseindeutig

Funktionen sind linkstotal und rechtseindeutig; Partielle Funktionen nur rechtseindeutig ($f: A \rightarrow B$)

Wörter sind surjektive Abbildungen: $w: \mathbb{G}_n \rightarrow A'$ $w \in A^*$ $A' \subseteq A^*$

Vorkommen eines Zeichens: $N_x(\varepsilon) = 0$ $\forall y \in A: \forall w \in A^*: N_x(yw) = \begin{cases} 1 + N_x(w), & \text{falls } y = x \\ N_x(w), & \text{falls } y \neq x \end{cases}$

Konkatenationsabschluss: $A^* = \bigcup_{i=0}^{\infty} A^i$ ε -freier Konkatenationsabschluss: $A^+ = \bigcup_{i=1}^{\infty} A^i$

Binäre Operation: $\diamond: M \times M \rightarrow M$ (Kommutativität, Assoziativität)

Formale Sprache L: $L \subseteq A^*$ (Konkatenationsabschluss wie bei Wörtern)

Produkt: $L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$

Potenzen: $L^0 = \{\varepsilon\}$ $\forall k \in \mathbb{N}_0: L^{k+1} = L \cdot L^k$

Algorithmus: (Vollständige Induktion über Schleifeninvariante)

- Endliche Beschreibung
- Elementare Anweisungen
- Determinismus
- Endliche Eingabe \rightarrow endliche Ausgabe
- Endlich viele Schritte
- Beliebige große Eingaben möglich
- Verständlich / Nachvollziehbar

Dokument besteht aus: Inhalt, Struktur und Erscheinungsbild

Kontextfreie Grammatiken (Typ-2-Grammatiken), rechtslineare Grammatiken (Typ-3-Grammatiken)

$G = (N, T, S, P)$ (Wichtig: $N \cap T = \emptyset$; $S \in N$; $P \subseteq N \times V^*$ mit $V = N \cup T$)

Ableitung eines Wortes nach einer Grammatik (als Baum oder mit \Rightarrow bzw. \Rightarrow^* wobei $R_{\Rightarrow} \subseteq V^* \times V^*$)

Von einer Grammatik erzeugte formale Sprache: $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$

Relationen:

Produkt: $R_1 \subseteq M_1 \times M_2, R_2 \subseteq M_2 \times M_3$, dann: $R_2 \circ R_1 = \{(x, z) \in M_1 \times M_3 \mid \exists y \in M_2: (x, y) \in R_1 \wedge (y, z) \in R_2\}$

Identische Abbildung: $I_M = \{(x, x) \mid x \in M\}$

Potenzen: $R^0 = I_M$ $\forall i \in \mathbb{N}_0: R^{i+1} = R^i \circ R$

transitiv-reflexive-Hülle: $R^* = \bigcup_{i=0}^{\infty} R^i$

Ein Byte $\hat{=}$ 8 Bit

Speicher als Abbildungen: $(a, a' \in \text{Adr}; v \in \text{Val}; m, m' \in \text{Mem})$

- Gesamtzustand des Speichers: $m: \text{Adr} \rightarrow \text{Val}$
- Lesen aus dem Speicher: $\text{memread}: \text{Val}^{\text{Adr}} (= \text{Mem}) \times \text{Adr} \rightarrow \text{Val}, (m, a) \mapsto m(a)$
- In den Speicher schreiben: $\text{memwrite}: \text{Val}^{\text{Adr}} \times \text{Adr} \times \text{Val} \rightarrow \text{Val}^{\text{Adr}}, (m, a, v) \mapsto m'$

$$m'(a') = \begin{cases} v & \text{falls } a' = a \\ m(a') & \text{falls } a' \neq a \end{cases}$$

- 1 Megabyte $\hat{=}$ 10^6 Bytes; 1 Mebibyte $\hat{=}$ 1024^2 Bytes = 2^{20} Bytes

Codierung: $\text{num}_n(x) \hat{=}$ Bedeutung von x , $\text{Num}_n(w) \hat{=}$ Bedeutung des Wort w ($n \in \mathbb{N}$), $\text{Repr}_n(w) \hat{=}$ $\text{Num}_n(w)$ aber ohne führende Nullen, $\text{Trans}_{n,m} = \text{Repr}_n \circ \text{Num}_m$, $\text{Sem} \hat{=}$ Menge von Bedeutungen

Bsp.: $Z_2 = \{0,1\}$ $\text{num}_2(0) = 0, \text{num}_2(1) = 1, \text{Num}_2(\varepsilon) = 0$
 $\forall w \in Z_2^* \forall x \in Z_2: \text{Num}_2(wx) = 2 \cdot \text{Num}_2(w) + \text{num}_2(x)$
 $f: L_1 \rightarrow L_2$ heißt Übersetzung, wenn: $\forall w \in L_a: \text{sem}_A(w) = \text{sem}_B(f(w))$ und f injektiv ist.

Homomorphismus: A, B Alphabete $h^{**}(\varepsilon) = \varepsilon \quad \forall w \in A^*: \forall x \in A: h^{**}(wx) = h^{**}(w)h(x)$

Präfixfreie Decodierung: $u(w) = \begin{cases} \varepsilon & \text{falls } w = \varepsilon \\ xu(w') & \text{falls } w = h(x)w' \text{ für ein } x \in A \\ \perp & \text{sonst} \end{cases}$

Huffman-Codierung ist eine Abbildung $h: A^* \rightarrow Z_2^*$ die ein ε -freier Homomorphismus ist

Graphen: $G = (V, E)$ mit $E \subseteq V \times V$ und $|V|$ endlich
 Teilgraph $G' = (V', E')$ mit $V' \subseteq V$ und $E' \subseteq E \cap V' \times V'$

Pfade: $p = (v_0, \dots, v_n)$ ist Pfad wenn $\forall i \in \mathbb{G}_n: (v_i, v_{i+1}) \in E, n = |p| - 1$ heißt Länge des Pfads, ist $v_0 = v_n$ so heißt der Pfad geschlossen bzw. ist ein Zyklus (ist er wiederholungsfrei, so nennt man ihn einfachen Zyklus). Ein Graph heißt streng zusammenhängend, wenn $\forall (x, y) \in V^2: \exists \text{ ein Pfad } p \text{ von } x \text{ nach } y$ ($\Leftarrow E^* = V \times V$). Ein Baum ist ein Graph mit einem Knoten $r \in V: \forall x \in V$ gibt es genau einen Pfad von r nach x .

Eingangsgrad: $d^-(y) = |\{x | (x, y) \in E\}|$, Ausgangsgrad: $d^+(x) = |\{y | (x, y) \in E\}|$

G_1 ist Isomorph zu $G_2 \Leftrightarrow$ es existiert eine bijektive Abbildung $f: V_1 \rightarrow V_2$ mit $\forall x \in V_1: \forall y \in V_1: (x, y) \in E_1 \Leftrightarrow (f(x), f(y)) \in E_2$

Grad eines Knoten in ungerichteten Graphen: $d(x) = |\{y | y \neq x \wedge \{x, y\} \in E\}| + \begin{cases} 2 & \text{falls } \{x, x\} \in E \\ 0 & \text{sonst} \end{cases}$

Adjazenzmatrix: $A \in K^{n \times n} \quad A_{ij} = \begin{cases} 1 & \text{falls } (i, j) \in E \\ 0 & \text{falls } (i, j) \notin E \end{cases}$

Erreichbarkeitsrelation $E^* = \bigcup_{i=0}^{n-1} E^i$ Wegematrix: $W \in K^{n \times n} \quad W_{ij} = \begin{cases} 1 & \text{falls } (i, j) \in E^* \\ 0 & \text{falls } (i, j) \notin E^* \end{cases}$

(zur Berechnung siehe Skript und Algorithmus von Warshall)

Im Folgenden seien g und f Funktionen mit $f, g: \mathbb{N}_0 \rightarrow \mathbb{R}_0^+$

- $g(n) \in \Theta(f(n)) \Leftrightarrow \exists c, c' \in \mathbb{R}_+: \exists n_0 \in \mathbb{N}_0: \forall n \geq n_0: cf(n) \leq g(n) \leq c'f(n) \Leftrightarrow f \asymp g$
 $\Theta(f) = \{g | g \asymp f\}$ g wächst größenordnungsmaßig genau so schnell wie f
- $g(n) \in O(f(n)) \Leftrightarrow \exists c \in \mathbb{R}_+: \exists n_0 \in \mathbb{N}_0: \forall n \geq n_0: g(n) \leq cf(n) \Leftrightarrow f \geq g$
 $O(f) = \{g | g \leq f\}$ g wächst asymptotisch höchstens so schnell wie f
- $g(n) \in \Omega(f(n)) \Leftrightarrow \exists c \in \mathbb{R}_+: \exists n_0 \in \mathbb{N}_0: \forall n \geq n_0: g(n) \geq cf(n) \Leftrightarrow f \leq g$
 $\Omega(f) = \{g | g \geq f\}$ g wächst asymptotisch mindestens so schnell wie f

Rechenregeln:

- $\forall f: \mathbb{N}_0 \rightarrow \mathbb{R}_0^+ \forall a, b \in \mathbb{R}_+: a \cdot f(n) \asymp b \cdot f(n) \Leftrightarrow \Theta(af(n)) = \Theta(bf(n))$
- $g(n) \in O(f(n)) \Leftrightarrow f(n) \in \Omega(g(n))$
- $\Theta(f(n)) = O(f(n)) \cap \Omega(f(n)) \Leftrightarrow g \asymp f \Leftrightarrow g \leq f \wedge g \geq f$

- $O(f_1) + O(f_2) = O(f_1 + f_2)$
- $g_1 \leq f_1 \wedge g_1 \asymp g_2 \wedge f_1 \asymp f_2 \Rightarrow g_2 \leq f_2$
- $g \leq f \Rightarrow O(g) \subseteq O(f) \text{ und } O(g + f) = O(f)$

Mastertheorem: $T(n) = aT\left(\frac{n}{b}\right) + f(n)$

1. $f(n) \in O(n^{\log_b a - \varepsilon}), \varepsilon > 0 \Rightarrow T(n) \in \Theta(n^{\log_b a})$
2. $f(n) \in \Theta(n^{\log_b a}) \Rightarrow T(n) \in \Theta(n^{\log_b a} \cdot \log(n))$
3. $f(n) \in \Omega(n^{\log_b a + \varepsilon}), \varepsilon > 0 \wedge \exists 0 < d < 1 \exists n_0 \in \mathbb{N}: \forall n \geq n_0: af\left(\frac{n}{b}\right) \leq df(n) \Rightarrow T(n) \in \Theta(f(n))$

Mealy-Automat (endlich): $A = (Z, z_0, X, f, Y, g)$ mit endlicher Zustandsmenge Z , Anfangszustand $z_0 \in Z$, Eingabealphabet X , Ausgabealphabet Y , Zustandsüberföhrungsfunktion $f: Z \times X \rightarrow Z$, Ausgabefunktion $g: Z \times X \rightarrow Y^*$

Moore-Automat: $A = (Z, z_0, X, f, Y, h)$ mit Ausgabefunktion $h: Z \rightarrow Y^*$, Rest wie bei Mealy-Automat

- Zustandsfunktionen:
 $f^*: Z \times X^* \rightarrow Z \quad f^*(z, \varepsilon) = z, \quad \forall w \in X^*: \forall x \in X: f^*(z, wx) = f(f^*(z, w), x)$
 $f^{**}: Z \times X^* \rightarrow Z^* \quad f^{**}(z, \varepsilon) = z, \quad \forall w \in X^*: \forall x \in X: f^{**}(z, wx) = f^{**}(z, w) \cdot f(f^*(z, w), x)$
 - Ausgabefunktionen:
 $g^*: Z \times X^* \rightarrow Y^* \quad g^*(z, \varepsilon) = \varepsilon, \quad \forall w \in X^*: \forall x \in X: g^*(z, wx) = g(f^*(z, w), x)$
 $g^{**}: Z \times X^* \rightarrow Y^* \quad g^{**}(z, \varepsilon) = \varepsilon, \quad \forall w \in X^*: \forall x \in X: g^{**}(z, wx) = g^{**}(z, w) \cdot g(f^*(z, w), x)$
- Moore-Automaten: $g^* = h \circ f^* [\Leftrightarrow g^*(z, w) = h(f^*(z, w))]$ und $g^{**} = h^{**} \circ f^{**}$

Endlicher Akzeptor: $A = (Z, z_0, X, f, F)$ mit Menge $F \subseteq Z$ akzeptierender Zustände, Rest wie bei Moore-Automaten

Von einem Akzeptor akzeptierte formale Sprache: $L(A) = \{w \in X^* | f^*(z_0, w) \in F\}$

Reguläre Ausdrücke: $G = (\{R\}, \{ |, (,), *, \phi \} \cup A, R, P)$ mit $P = \{R \rightarrow \phi | (R|R) | (RR) | (R^*)\} \cup \{R \rightarrow x | x \in A\}$, A Alphabet

Beschriebene formale Sprache: $\langle \phi \rangle = \{ \}$, $\langle x \rangle = \{x\} (x \in A)$, $\langle R_1 | R_2 \rangle = \langle R_1 \rangle \cup \langle R_2 \rangle$, $\langle R_1 R_2 \rangle = \langle R_1 \rangle \cdot \langle R_2 \rangle$, $\langle R^* \rangle = \langle R \rangle^*$

Für jede formale Sprache L sind die folgenden drei Aussagen äquivalent:

- L kann von einem endlichen Akzeptor erkannt werden.
- L kann durch einen regulären Ausdruck beschrieben werden.
- L kann von einer rechtslinearen Grammatik erzeugt werden.

Kantorowitsch- (/Regex-) Bäume: $h(T) = \begin{cases} 0 & \text{falls die Wurzel Blatt ist} \\ 1 + \max_i h(U_i) & \text{falls die } U_i \text{ alle Unterbäume von } T \text{ sind} \end{cases}$

Strukturelle Induktion: siehe Skript S.154

Turingmaschinen: $T = (Z, z_0, X, f, g, m)$ mit Zustandsmenge Z , Anfangszustand $z_0 \in Z$, Bandalphabet X , partielle Zustandsüberföhrungsfunktion $f: Z \times X \rightarrow Z$, partielle Ausgabefunktion $g: Z \times X \rightarrow X$, partielle Bewegungsfunktion $m: Z \times X \rightarrow \{-1, 0, 1\}$!!!Achtung: Eingabealphabet A einer Turingmaschine: $A \subseteq X \setminus \{\square\}$

Konfiguration: (Gesamtzustand zu einem Zeitpunkt) $c_i = (z, b, p)$ $z \in Z$ Zustand der Steuereinheit, $b: \mathbb{Z} \rightarrow X$ aktuelle Bandbeschriftung, $p \in \mathbb{Z}$ Position des Kopfs

Konfiguration nach t Schritten: $\Delta_0 = I \quad \forall t \in \mathbb{N}_+: \Delta_{t+1} = \Delta_1 \circ \Delta_t \quad \Delta_*: C_T \rightarrow C_T$

Von Turingmaschinen erkennbare Sprachen heißen aufzählbare Sprachen, von TM akzeptierte Sprachen heißen entscheidbare Sprachen, wenn die TM für jede Eingabe hält. Eine TM kann man codieren.

Zeitkomplexität: $time_T: A^+ \rightarrow \mathbb{N}_+$ mit $time_T(w) = t$, sodass $\Delta_t(c_0(w)) = \Delta_*(c_0(w))$
 $Time_T: \mathbb{N}_+ \rightarrow \mathbb{N}_+$ mit $Time_T(n) = \max\{time_T(w) | w \in A^n\}$

Raumkomplexität: $space_T: A^+ \rightarrow \mathbb{N}_+$ mit $space_T(w) = \text{Anzahl der "besuchten" Felder}$
 $Space_T: \mathbb{N}_+ \rightarrow \mathbb{N}_+$ mit $Space_T(n) = \max\{space_T(w) | w \in A^n\}$

Es gilt: $space(w) \leq \max(|w|, 1 + time(w))$

Komplexitätsklassen: Es gilt: $P \subseteq PSPACE$

- P : Menge der Entscheidungsprobleme die eine TM in polynomieller **Zeit**komplexität entscheiden kann
- $PSPACE$: Menge aller Entscheidungsprobleme die eine TM in polynomieller **Raum**komplexität entscheiden kann

Es gibt Probleme die eine TM nicht entscheiden kann (z.B. Halteproblem siehe S.172).

Busy-Beaver-Funktion: TM mit $n+1$ Zuständen, Bandalphabet $X = \{0, 1\}$, startet auf leerem Band und hält nach endlich vielen Schritten an. $bb: \mathbb{N}_+ \rightarrow \mathbb{N}_+$ $bb(n) = \text{maximale Anzahl an Einsen die eine } n - \text{Biebermaschine auf dem Band hinterlässt}$

Es gilt: $f: \mathbb{N}_+ \rightarrow \mathbb{N}_+$ berechenbar $\Rightarrow \exists n_0 \in \mathbb{N}: \forall n \geq n_0: bb(n) > f(n)$ und $bb(n)$ ist nicht berechenbar

Eine Relation R heißt Äquivalenzrelation $\Leftrightarrow R$ ist reflexiv, symmetrisch und transitiv

Eine Relation R heißt Kongruenzrelation $\Leftrightarrow R$ ist Äquivalenzrelation und mit allen gerade interessierenden Funktionen f verträglich bzw. mit allen gerade interessierenden binären Relationen \diamond verträglich.

f ist verträglich mit $R \Leftrightarrow$ Es gilt für \sim Äquivalenzrelation, M Menge, $f: M \rightarrow M$ Abbildung: $\forall x_1, x_2 \in M: x_1 \sim x_2 \Rightarrow f(x_1) \sim f(x_2)$

\diamond ist verträglich mit $R \Leftrightarrow$ Es gilt für \diamond binäre Relation, M Menge: $\forall x_1, x_2 \in M \forall y_1, y_2 \in M: x_1 \sim x_2 \wedge y_1 \sim y_2 \Rightarrow x_1 \diamond y_1 \sim x_2 \diamond y_2$

$R \subseteq M \times M$ heißt Halbordnung $\Leftrightarrow R$ ist reflexiv, antisymmetrisch ($\forall x, y \in M: xRy \wedge yRx \Rightarrow x = y$) und transitiv

Halbordnungen lassen sich in einem Hasse-Diagramm H_R darstellen (gerichteter azyklischer Graph), sodass H_R mit der reflexiv-transitiven Hülle wieder die ursprüngliche Halbordnung ist ($H_R = (R \setminus I) \setminus (R \setminus I)^2$ und $H_R^* = R$).

Es sei (M, \sqsubseteq) eine halbgeordnete Menge und $T \subseteq M$:

- $x \in T$ heißt maximales Element von $T \Leftrightarrow \nexists y \in T: x \sqsubset y \wedge x \neq y$
- $x \in T$ heißt minimales Element von $T \Leftrightarrow \nexists y \in T: y \sqsubset x \wedge y \neq x$
- $x \in T$ heißt größtes Element von $T \Leftrightarrow \forall y \in T: y \sqsubseteq x$ $x \in T$ heißt kleinstes Element von $T \Leftrightarrow \forall y \in T: x \sqsubseteq y$
- $x \in M$ heißt obere Schranke von $T \Leftrightarrow \forall y \in T: y \sqsubseteq x$ $x \in M$ heißt untere Schranke von $T \Leftrightarrow \forall y \in T: x \sqsubseteq y$
- Das kleinste Element der Menge oberer Schranken heißt Supremum. Bezeichnung: $\sqcup T$ oder $\sup(T)$
- Das größte Element der Menge unterer Schranken heißt Infimum.

Eine aufsteigende Kette ist eine abzählbare unendliche Folge $(x_0, x_1, x_2, \dots), x_i \in \text{halbgeordneter Menge}$ und es gilt: $\forall i \in \mathbb{N}_0: x_i \sqsubseteq x_{i+1}$.

In einer vollständigen Halbordnung besitzt jede aufsteigende Kette $(x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots)$ ein kleinstes Element (\perp) und ein Supremum ($\sqcup_i x_i$).

\sqsubseteq sei eine Halbordnung auf M : $f: M \rightarrow M$ heißt monotone Abbildung $\Leftrightarrow \forall x, y \in M: x \sqsubseteq y \Rightarrow f(x) \sqsubseteq f(y)$. $f: M \rightarrow M$ heißt stetige Abbildung \Leftrightarrow für jede aufsteigende Kette gilt: $f(\sqcup_i x_i) = \sqcup_i f(x_i)$

Sei $f: D \rightarrow D$ eine monotone, stetige Abbildung auf einer vollständigen Halbordnung (D, \sqsubseteq) mit $x_0 = \perp$ und $\forall i \in \mathbb{N}_0: x_{i+1} = f(x_i) \Rightarrow x_i$ bilden eine Kette, Supremum $x_f = \sqcup_i x_i$ dieser Kette ist Fixpunkt ($f(x_f) = x_f$) und x_f ist kleinster Fixpunkt von f : Wenn $f(y_f) = y_f$, dann $x_f \sqsubseteq y_f$.

Eine Halbordnung $R \subseteq M \times M$ ist eine (totale) Ordnung, wenn gilt: $\forall x, y \in M: xRy \vee yRx$