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 ($f: A \dashrightarrow B$)

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

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

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

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

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

Produkt: $L_1 \cdot L_2 = \{w_1 w_2 | w_1 \in L_1 \wedge w_2 \in L_2\}$
Potenzen: $L^0 = \{\varepsilon\} \quad \forall k \in \mathbb{N}_0: L^{k+1} = L \cdot L^k$

Algorithmus: (Vollständige Induktion über Schleifeninvariante)

- Endliche Beschreibung
- Elementare Anweisungen
- Determinismus
- Endliche Eingabe -> endliche Ausgabe
- Endlich viele Schritte
- Beliebig 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 $\Longrightarrow$ bzw. $\Longrightarrow^*$ wobei $R_{\Rightarrow} \subseteq V^* \times V^*$)
Von einer Grammatik erzeugte formale Sprache: $L(G) = \{w \in T^* | S \Longrightarrow^* w\}$

Relationen:

Produkt: $R_1 \subseteq M_1 \times M_2, R_2 \subseteq M_2 \times M_3$ $, dann: R_2 \circ R_1 = \{(x, z) \in M_1 \times M_3 | \exists y \in M_2: (x, y) \in R_1 \wedge (y, z) \in R_2\}$
Identische Abbildung: $I_M = \{(x, x) | x \in M\}$
Potenzen: $R^0 = I_M \quad \forall\ i \in \mathbb{N}_0: R^{i+1} = R^i \circ R$
transitiv-reflexive-Hülle: $R^* = \bigcup_{i=0}^{\infty} R^i$

Ein Byte $\widehat{=}$ 8 Bit

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

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

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

von Holger Ebhart

- 1 Megabyte ≙ 10^6 Bytes; 1 Mebibyte ≙ 1024^2 Bytes = 2^{20} Bytes

Codierung: $num_n(x)$ ≙ Bedeutung von x, $Num_n(w)$ ≙ Bedeutung des Wort w ($n \in \mathbb{N}$), $Repr_n(w)$ ≙ $Num_n(w)$ aber ohne führende Nullen, $Trans_{n,m} = Repr_n \circ Num_m$, Sem ≙ Menge von Bedeutungen

Bsp.: $Z_2 = \{0,1\}$ $\quad num_2(0) = 0, num_2(1) = 1, Num_2(\varepsilon) = 0$
$\forall w \in Z_2^* \forall x \in Z_2: Num_2(wx) = 2 \cdot Num_2(w) + num_2(x)$
$f: L_1 \longrightarrow L_2$ heißt Übersetzung, wenn: $\forall w \in L_a: sem_A(w) = sem_B\big(f(w)\big)$ und f injektiv ist.

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

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

Huffman-Codierung ist eine Abbildung $h: A^* \longrightarrow Z_2^*$ 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: $p=(v_0,\ldots,v_n)$ 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 $\forall (x,y) \in V^2: \exists\ ein\ Pfad\ p\ von\ x\ nach\ y$ ($\Longleftarrow E^* = V \times V$). 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: $d^-(y) = |\{x | (x,y) \in E\}|$, $\quad$ Ausgangsgrad: $d^+(x) = |\{y | (x,y) \in E\}|$

G_1 ist Isomorph zu G_2 ⇔ es existiert eine bijektive Abbildung $f: V_1 \longrightarrow V_2$ 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: $d(x) = |\{y | y \neq x \wedge \{x,y\} \in E\}| + \begin{cases} 2 & falls\ \{x,x\} \in E \\ 0 & sonst \end{cases}$

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

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

(zur Berechnung siehe Skript und Algorithmus von Warshall)

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

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

Rechenregeln:

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

- $O(f_1) + O(f_2) = O(f_1 + f_2)$
- $g_1 \preceq f_1 \wedge g_1 \asymp g_2 \wedge f_1 \asymp f_2 \Longrightarrow g_2 \preceq f_2$
- $g \preceq f \Longrightarrow O(g) \subseteq O(f)\ und\ O(g+f) = O(f)$

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

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

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

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

- Zustandsfunktionen:
  $f^*: Z \times X^* \longrightarrow Z \qquad f^*(z,\varepsilon) = z, \quad \forall w \in X^*: \forall x \in X: f^*(z, wx) = f(f^*(z,w), x)$
  $f^{**}: Z \times X^* \longrightarrow Z^* \qquad f^{**}(z,\varepsilon) = z, \quad \forall w \in X^*: \forall x \in X: f^{**}(z, wx) = f^{**}(z,w) \cdot f(f^*(z,w), x)$
- Ausgabefunktionen:
  $g^*: Z \times X^* \longrightarrow Y^* \qquad g^*(z,\varepsilon) = \varepsilon, \quad \forall w \in X^*: \forall x \in X: g^*(z, wx) = g(f^*(z,w), x)$
  $g^{**}: Z \times X^* \longrightarrow Y^* \qquad g^{**}(z,\varepsilon) = \varepsilon, \quad \forall w \in X^*: \forall x \in X: g^{**}(z, wx) = g^{**}(z,w) \cdot g^*(z, wx)$
  Moore-Automaten: $g^* = h \circ f^*\ \left[\Leftrightarrow g^*(z,w) = h\big(f^*(z,w)\big)\right]$ 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 $P = \{R \longrightarrow \phi | (R|R) | (RR) | (R*)\} \cup \{R \longrightarrow x | x \in A\}$, 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: $h(T) = \begin{cases} 0 & falls\ die\ Wurzel\ Blatt\ ist \\ 1 + max_i h(U_i) & falls\ die\ U_i\ alle\ Unterbäume\ von\ T\ sind \end{cases}$

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 Zustandsüberführungsfunktion $f: Z \times X \dashrightarrow Z$, partielle Ausgabefunktion $g: Z \times X \dashrightarrow X$, partielle Bewegungsfunktion $m: Z \times X \dashrightarrow \{-1, 0, 1\}$!!!Achtung: Eingabealphabet A einer Turingmaschine: $A \subseteq X \backslash \{\Box\}$

Konfiguration: (Gesamtzustand zu einem Zeitpunkt) $c_i = (z, b, p)$ $z \in Z$ Zustand der Steuereinheit, $b: \mathbb{Z} \longrightarrow X$ aktuelle Bandbeschriftung, $p \in \mathbb{Z}$ Position des Kopfs
Konfiguration nach t Schritten: $\Delta_0 = I \quad \forall t \in \mathbb{N}_+: \Delta_{t+1} = \Delta_1 \circ \Delta_t \qquad \Delta_*: C_T \dashrightarrow C_T$

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

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

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 $X = \{\square, 1\}$, startet auf leerem Band und hält nach endlich vielen Schritten an. $bb: \mathbb{N}_+ \longrightarrow \mathbb{N}_+ \quad bb(n) = maximale\ Anzahl\ an\ Einsen\ die\ eine\ n - Biebermaschine\ auf\ dem\ Band\ hinterlässt$
Es gilt: $f: \mathbb{N}_+ \longrightarrow \mathbb{N}_+\ berechenbar \implies \exists\, n_0 \in \mathbb{N}: \forall n \geq n_0: bb(n) > f(n)$ und bb(n) ist nicht berechenbar

Eine Relation R heißt Äquivalenzrelation ⇔ R ist reflexiv, symmetrisch und transitiv
Eine Relation R heißt Kongruenzrelation ⇔ R ist Äquivalenzrelation und mit allen gerade interessierenden Funktionen f verträglich bzw. mit allen gerade interessierenden binären Relationen ⋄ verträglich.
f ist verträglich mit R ⇔ Es gilt für $\sim\ Äqivalenzrelation, M\ Menge, f: M \longrightarrow M\ Abbildung: \forall x_1, x_2 \in M: x_1 \sim x_2 \implies f(x_1) \sim f(x_2)$
⋄ ist verträglich mit R ⇔ Es gilt für $\diamond\ binäre\ Relation, M\ Menge: \forall x_1, x_2 \in M \forall y_1, y_2 \in M: x_1 \sim x_2 \land y_1 \sim y_2 \implies x_1 \diamond y_1 \sim x_2 \diamond y_2$

$R \subseteq M \times M$ heißt Halbordnung ⇔ R ist reflexiv, antisymmetrisch ($\forall x, y \in M: xRy \land yRx \implies x = y$) 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 ($H_R = (R \backslash I) \backslash (R \backslash I)^2$ $und\ H_R^* = R$).

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

- $x \in T$ heißt maximales Element von T ⇔ $\nexists y \in T: x \sqsubseteq y \land x \neq y$
- $x \in T$ heißt minimales Element von T ⇔ $\nexists y \in T: y \sqsubseteq x \land y \neq x$
- $x \in T$ heißt größtes Element von T ⇔ $\forall y \in T: y \sqsubseteq x$ $\quad$ $x \in T$ heißt kleinstes Element von T ⇔ $\forall y \in T: x \sqsubseteq y$
- $x \in M$ heißt obere Schranke von T ⇔ $\forall y \in T: y \sqsubseteq x$ $\quad$ $x \in M$ heißt untere Schranke von T ⇔ $\forall y \in T: x \sqsubseteq y$
- 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 $(x_0, x_1, x_2, \dots), x_i \in halbgeordneter\ Menge$ und es gilt: $\forall i \in \mathbb{N}_0: x_i \sqsubseteq x_{i+1}$.
In einer vollständigen Halbordnung besitzt jede aufsteigende Kette ($x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \cdots$) ein kleinstes Element ($\perp$) und ein Supremum ($\sqcup_i x_i$).

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

Sei $f: D \longrightarrow D$ eine monotone, stetige Abbildung auf einer vollständigen Halbordnung $(D, \sqsubseteq)$ mit $x_0 = \perp\ und\ \forall i \in \mathbb{N}_0: x_{i+1} = f(x_i) \implies x_i$ 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: $\forall x, y \in M: xRy \lor yRx$