GRUNDLAGEN: ALPHABETE, WÖRTER, SPRACHEN

- **Alphabet**: Σ endliche, nichtleere Menge. Elemente: Zeichen/Symbole.
- $|\Sigma|$: Anzahl Elemente in Σ.
- **Wort**: Folge von Symbolen aus Σ.
- ε: Leeres Wort.
- Σ^*: Menge aller Wörter über Σ.
- Σ^+: $\Sigma^* \setminus \{\varepsilon\}$.
- **Konkatenation**: $u \circ v \equiv uv$.
- $w^n = w \circ \cdots \circ w$ (n-mal). $w^0 = \varepsilon$.
- **Länge**: $|w|$, Anzahl Symbole in w. $|\varepsilon| = 0$. $|w^n| = n|w|$.
- **Sprache**: $L \subseteq \Sigma^*$.

GRAMMATIKEN

- **Grammatik**: $G = (V, \Sigma, P, S)$, wobei:
  - V: Endliche Menge der Variablen.
  - Σ: Terminalalphabet. $V \cap \Sigma = \emptyset$.
  - P: Endliche Menge der Produktionen. $P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$.
  - $S \in V$: Startvariable.
- **Unmittelbare Ableitung**: $u \Rightarrow_G v$, falls $u = xyz$, $v = xy'z$, $(y \to y') \in P$, und $x, z \in (V \cup \Sigma)^*$.
- **Ableitung**: $u \Rightarrow_G^* v$, falls $u = v$ oder $u = w_1 \Rightarrow_G w_2 \Rightarrow_G \cdots \Rightarrow_G w_k = v$.
- **Erzeugte Sprache**: $L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$.
- **Ableitung in k Schritten**: $(w_0, \ldots, w_k)$ mit $w_0 = S, w_k = w, w_i \Rightarrow_G w_{i+1}$ für $i = 0, \ldots, k-1$.

DIE CHOMSKY-HIERARCHIE

- **Typ 0 (uneingeschränkt)**: Jede Grammatik.
- **Typ 1 (kontextsensitiv)**: Grammatik G mit $(u \to v) \in P \Rightarrow |u| \leq |v|$.
  - Ausnahme: $S \to \varepsilon$ erlaubt, falls S nicht auf rechter Seite einer Produktion vorkommt.
- **Typ 2 (kontextfrei)**: Typ-1-Grammatik mit $(u \to v) \in P \Rightarrow u \in V$ (d.h., u ist einzelne Variable).
- **Typ 3 (regulär)**: Typ-2-Grammatik mit $(u \to v) \in P \Rightarrow v \in \Sigma$ oder $v \in \Sigma V$.
- **Sprache vom Typ k**: $L \subseteq \Sigma^*$ ist vom Typ $k \leftrightarrow \exists$ Typ-k-Grammatik G mit $L = L(G)$.
- **Wortproblem für Typ-1-Sprachen**: Entscheidbar.

ENDLICHE AUTOMATEN & REGULÄRE SPRACHEN (TYP 3)

- **Deterministischer Endlicher Automat (DEA)**: $M = (Z, \Sigma, \delta, z_0, E)$, wobei:
  - Z: Endliche Zustandsmenge.
  - Σ: Eingabealphabet. $Z \cap \Sigma = \emptyset$.
  - $\delta : Z \times \Sigma \to Z$: Überführungsfunktion.
  - $z_0 \in Z$: Startzustand.
  - $E \subseteq Z$: Menge der Endzustände.
- **Erweiterte Überführungsfunktion** $\hat{\delta} : Z \times \Sigma^* \to Z$:
  - $\hat{\delta}(z, \varepsilon) = z$.
  - $\hat{\delta}(z, ax) = \hat{\delta}(\delta(z, a), x)$ für $a \in \Sigma, x \in \Sigma^*$.
- **Von DEA M akzeptierte Sprache**: $L(M) = \{x \in \Sigma^* \mid \hat{\delta}(z_0, x) \in E\}$.
- **Nichtdeterministischer Endlicher Automat (NEA)**: $M = (Z, \Sigma, \delta, z_0, E)$, wobei $\delta : Z \times \Sigma \to P(Z)$.
- **Erweiterte Überführungsfunktion** $\hat{\delta} : P(Z) \times \Sigma^* \to P(Z)$:
  - $\hat{\delta}(Z', \varepsilon) = Z'$.
  - $\hat{\delta}(Z', ax) = \hat{\delta}(\bigcup_{z \in Z'} \delta(z, a), x)$.
- **Von NEA M akzeptierte Sprache**: $L(M) = \{x \in \Sigma^* \mid \hat{\delta}(\{z_0\}, x) \cap E \neq \emptyset\}$.
- **Satz**: $\forall$ NEA M, $\exists$ DEA M' s.t. $L(M) = L(M')$.
- **Satz**: $L \subseteq \Sigma^*$ ist regulär $\leftrightarrow \exists$ DEA M mit $L = L(M) \leftrightarrow \exists$ Typ-3-Grammatik G mit $L = L(G)$.
- **Pumping Lemma für reguläre Sprachen**: L regulär $\Rightarrow \exists n \in \mathbb{N}$ s.t. $\forall x \in L, |x| \geq n \Rightarrow \exists u, v, w \in \Sigma^*$ s.t. $x = uvw \wedge |v| \geq 1 \wedge |uv| \leq n \wedge (\forall i \geq 0 : uv^i w \in L)$.

KELLERAUTOMATEN & KONTEXTFREIE SPRACHEN (TYP 2)

- **Nichtdeterministischer Kellerautomat (NKA/PDA)**: $M = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$, wobei:
  - Z: Endliche Zustandsmenge.
  - Σ: Eingabealphabet.
  - Γ: Kelleralphabet.
  - $\delta : Z \times \Sigma \times \Gamma \to P(Z \times \Gamma^*)$: Überführungsfunktion (endliche Menge).
  - $z_0 \in Z$: Startzustand.
  - $\# \in \Gamma$: Unterstes Kellersymbol.
  - $E \subseteq Z$: Menge der Endzustände.
- **Arbeitsweise**: Bei (z, a, A) kann M in $(z', B_1 \ldots B_k)$ übergehen: A wird aus Keller entfernt, $B_1 \ldots B_k$ auf Keller geschrieben (B_1 unterster). Eingabekopf bewegt sich rechts.
- **Akzeptierte Sprache**: $L(M) = \{w \in \Sigma^* \mid M$ akzeptiert w (durch Erreichen eines Endzustands nach vollständiger Eingabe)$\}$.
- **Satz**: $L \subseteq \Sigma^*$ ist kontextfrei $\leftrightarrow \exists$ NKA M mit $L = L(M) \leftrightarrow \exists$ Typ-2-Grammatik G mit $L = L(G)$.
- **Pumping Lemma für kontextfreie Sprachen (uvwxy-Theorem)**: L kontextfrei $\Rightarrow \exists n \in \mathbb{N}$ s.t. $\forall z \in L, |z| \geq n \Rightarrow \exists u, v, w, x, y \in \Sigma^*$ s.t. $z = uvwxy \wedge |vx| \geq 1 \wedge |vwx| \leq n \wedge (\forall i \geq 0 : uv^i wx^i y \in L)$.

TURINGMASCHINEN & TYP-0/1 SPRACHEN

- **Turingmaschine (TM)**: $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$, wobei:
  - Z: Zustandsmenge.
  - Σ: Eingabealphabet.
  - $\Gamma \supseteq \Sigma$: Arbeitsalphabet.
  - $z_0 \in Z$: Startzustand.
  - $\square \in \Gamma \setminus \Sigma$: Leerzeichen/Blank.
  - $E \subseteq Z$: Endzustände.
  - δ: Übergangsfunktion.
- **DTM (Deterministische TM)**: $\delta : Z \times \Gamma \to Z \times \Gamma \times \{L, N, R\}$.
- **NTM (Nichtdeterministische TM)**: $\delta : Z \times \Gamma \to P(Z \times \Gamma \times \{L, N, R\})$.
- **Konfiguration**: $k = uzv$ mit $u, v \in \Gamma^*, z \in Z$.
- **Startkonfiguration**: $z_0 w$ für Eingabe w.
- **Übergangsrelation** $\vdash_M$: $ua_m z b_1 \ldots b_n \vdash_M ua_m c z' b_2 \ldots b_n$ (N), $ua_m c b_2 z' \ldots b_n$ (R), $uz' a_m c b_2 \ldots b_n$ (L) gemäß $\delta(z, b_1) = (z', c, X)$. Spezialfälle bei Bandanfang/-ende durch $\square$.
- **Akzeptierte Sprache**: $L(M) = \{w \in \Sigma^* \mid z_0 w \vdash_M^* uzv$ für ein $z \in E, u, v \in \Gamma^*\}$.
- **Linear beschränkter Automat (LBA)**: NTM mit Bandendemarkierungen $\lhd, \rhd \in \Gamma \setminus \Sigma$ und Kopfbewegungsbeschränkungen:
  - Kein Kopfbewegung nach links bei $\lhd$, kein nach rechts bei $\rhd$.
  - $\lhd, \rhd$ nicht überschreibbar.
  - Akzeptanz: $z_0 \lhd w \rhd \vdash_M^* uzv$ für $z \in E$.
- **Satz**: L ist kontextsensitiv (Typ 1) $\leftrightarrow \exists$ LBA M mit $L = L(M)$.
- **Satz**: L ist vom Typ 0 $\leftrightarrow \exists$ TM M mit $L = L(M) \leftrightarrow \exists$ NTM M mit $L = L(M)$.

BERECHENBARKEIT & TURING-BERECHENBARKEIT

- **Intuitive Berechenbarkeit**: $f : \mathbb{N}^k \to \mathbb{N}$ berechenbar $\leftrightarrow \exists$ Algorithmus, der für Eingabe $(n_1, \ldots, n_k)$ nach endlicher Zeit $f(n_1, \ldots, n_k)$ ausgibt. Wenn f undefiniert, Algorithmus hält nicht.
- **Turing-Berechenbarkeit ($f : \mathbb{N}^k \to \mathbb{N}$)**: $\exists$ DTM M s.t.:
  - $f(n_1, \ldots, n_k) = m \Rightarrow M(\text{bin}(n_1)\# \ldots \#\text{bin}(n_k))$ hält mit $\square \ldots \square\text{bin}(m)\square \ldots \square$.
  - $f(n_1, \ldots, n_k)$ undefiniert $\Rightarrow M(\text{bin}(n_1)\# \ldots \#\text{bin}(n_k))$ hält nicht.
- **Turing-Berechenbarkeit ($f : \Sigma^* \to \Delta^*$)**: $\exists$ DTM M s.t.:
  - $f(x) = y \Rightarrow M(x)$ hält mit $\square \ldots \square y \square \ldots \square$.
  - $f(x)$ undefiniert $\Rightarrow M(x)$ hält nicht.
- **k-Band-DTM**: $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ mit $\delta : Z \times \Gamma^k \to Z \times \Gamma^k \times \{L, R, N\}^k$.
- **Satz**: $\forall k > 1$, $\forall$ k-Band-DTM M, $\exists$ 1-Band-DTM M' mit $L(M) = L(M')$ (oder sie berechnen dieselbe Funktion).
- **Hintereinanderschaltung von TM**: $M_1; M_2$ (sequenzielle Ausführung).
- **Bedingte Verzweigungen**: IF M_1 THEN M_2 ELSE M_3 (basierend auf Endzuständen von M_1).
- **Test auf Null**: "Band = 0?" (TM zur Prüfung, ob Bandinhalt "0" ist).
- **Schleifen**: WHILE Band $i \neq 0$ DO M (standard Schleifenkonstrukt).

DIE PROGRAMMIERSPRACHE LOOP

- **Syntaktische Komponenten**: Variablen $(x_0, x_1, \ldots)$, Konstanten $(0, 1, 2, \ldots)$, Operatoren $(+, -)$, Trennsymbole $(;:=)$, Schlüsselwörter (LOOP, DO, END).
- **Syntax von LOOP-Programmen**:
  - $x_i := x_j + c$
  - $x_i := x_j - c$
  - $P_1; P_2$
  - LOOP x_i DO P END
- **Semantik von LOOP**: Berechnet $f : \mathbb{N}^k \to \mathbb{N}$. Eingabewerte $n_1, \ldots, n_k$ in $x_1, \ldots, x_k$; andere Variablen 0.
  - $x_i := x_j + c$: $x_i \leftarrow \text{val}(x_j) + c$.
  - $x_i := x_j - c$: $x_i \leftarrow \max(0, \text{val}(x_j) - c)$.
  - $P_1; P_2$: Erst P_1, dann P_2.
  - LOOP x_i DO P END: P wird val(x_i)-mal ausgeführt (Wert von x_i bei Loop-Eintritt).