GRUNDLAGEN: SPRACHEN UND GRAMMATIKEN

- **Alphabet**: Endliche, nichtleere Menge Σ. Elemente heißen Zeichen/Symbole.
- **Wort**: Folge von Symbolen aus Σ.
- **Leeres Wort**: ε. $|\varepsilon| = 0$.
- Σ^*: Menge aller Wörter über Σ.
- Σ^+: Menge aller nicht-leeren Wörter über Σ, d.h. $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$.
- **Sprache**: Teilmenge von Σ^*.
- **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.
- **Ableitungsschritt**: $u \Rightarrow_G v \iff u = xyz, v = xy'z$ mit $(y, y') \in P$.
- **Ableitung**: $u \Rightarrow_G^* v \iff u = v$ oder $u \Rightarrow_G w_1 \Rightarrow_G \cdots \Rightarrow_G w_k \Rightarrow_G v$.
- **Von G erzeugte Sprache**: $L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$.

CHOMSKY-HIERARCHIE

- **Typ 0 (Phrase-Struktur-Grammatik)**: Keine Einschränkungen an P.
- **Typ 1 (Kontextsensitiv)**: Für alle $(u \to v) \in P$ gilt $|u| \leq |v|$.
- **Typ 2 (Kontextfrei)**: Für alle $(u \to v) \in P$ gilt $u \in V$ (d.h., u ist eine einzelne Variable).
- **Typ 3 (Regulär)**: Für alle $(u \to v) \in P$ gilt:
  - $u \in V$.
  - $v \in \Sigma$ oder $v \in \Sigma V$.
- **Sonderfall ε-Regel**: Für Typ 1, 2, 3 ist $S \to \varepsilon$ erlaubt. Falls $S \to \varepsilon \in P$, darf S auf der rechten Seite keiner Produktion vorkommen.
- **Sprache vom Typ i**: $L \subseteq \Sigma^*$ ist vom Typ $i \iff \exists$ Grammatik G vom Typ i mit $L = L(G)$.
- **Wortproblem für Typ-1-Sprachen**: Entscheidbar.

REGULÄRE (TYP-3-) SPRACHEN

- **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 $\hat{\delta} : Z \times \Sigma^* \to Z$ (für DEA)**:
  - $\hat{\delta}(z, \varepsilon) = z$
  - $\hat{\delta}(z, ax) = \hat{\delta}(\delta(z, a), x)$ 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 $\delta : Z \times \Sigma \to \mathcal{P}(Z)$.
- **Erweiterte Überführungsfunktion $\hat{\delta} : \mathcal{P}(Z) \times \Sigma^* \to \mathcal{P}(Z)$ (für NEA)**:
  - $\hat{\delta}(Z', \varepsilon) = Z'$
  - $\hat{\delta}(Z', ax) = \hat{\delta}(\bigcup_{z \in Z'} \delta(z, a), x)$ für $a \in \Sigma, x \in \Sigma^*$
- **Von NEA M akzeptierte Sprache**: $L(M) = \{x \in \Sigma^* \mid \hat{\delta}(\{z_0\}, x) \cap E \neq \emptyset\}$.
- **Äquivalenz NEA $\leftrightarrow$ DEA**: Zu jedem NEA M existiert ein DEA M' mit $L(M) = L(M')$.
- **Äquivalenz Typ-3-Grammatik $\leftrightarrow$ DEA**: Sprache $L \subseteq \Sigma^*$ ist Typ 3 $\iff \exists$ DEA M mit $L = L(M)$.
- **Pumping-Lemma (uvw-Theorem) für reguläre Sprachen**:
  - (L regulär) $\implies (\exists n \in \mathbb{N})(\forall x \in L, |x| \geq n)(\exists u, v, w \in \Sigma^*)[x = uvw \wedge |v| \geq 1 \wedge |uv| \leq n \wedge (\forall i \geq 0 : uv^i w \in L)]$.
  - **Anwendung**: Contraposition: Ist die rechte Seite falsch, ist L nicht regulär.

KONTEXTFREIE (TYP-2-) SPRACHEN

- **Nichtdeterministischer Kellerautomat (NKA/PDA)**: $M = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$, wobei:
  - Z: Endliche Zustandsmenge.
  - Σ: Eingabealphabet.
  - Γ: Kelleralphabet.
  - $\delta : Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Z \times \Gamma^*)$: Überführungsfunktion (endlich).
  - $z_0 \in Z$: Startzustand.
  - $\# \in \Gamma$: Unterstes Kellersymbol.
  - $E \subseteq Z$: Menge der Endzustände.
- **Arbeitsweise NKA**:
  - Start: z_0, Eingabeanfang, Keller enthält nur $\#$.
  - Übergang $(z, a, A) \to (z', B_1 \ldots B_k)$: Ist M in z, liest a (oder ε), und A oberstes Kellersymbol $\implies$ M geht nach z', A wird durch $B_1 \ldots B_k$ ersetzt (B_1 wird oberstes Symbol), Eingabekopf nach rechts.
- **Von NKA M akzeptierte Sprache**: $L(M) = \{w \in \Sigma^* \mid M$ akzeptiert w (d.h. erreicht Endzustand nach vollst. Eingabe)$\}$.
- **Äquivalenz NKA $\leftrightarrow$ Kontextfreie Sprache**: Eine Sprache L ist kontextfrei $\iff \exists$ NKA M mit $L = L(M)$.
- **Pumping-Lemma (uvwxy-Theorem) für kontextfreie Sprachen**:
  - (L kontextfrei) $\implies (\exists n \in \mathbb{N})(\forall z \in L, |z| \geq n)(\exists u, v, w, x, y \in \Sigma^*)[z = uvwxy \wedge |vx| \geq 1 \wedge |vwx| \leq n \wedge (\forall i \geq 0 : uv^i wx^i y \in L)]$.
  - **Anwendung**: Contraposition: Ist die rechte Seite falsch, ist L nicht kontextfrei.

TURING-MASCHINEN (TYP-0-SPRACHEN)

- **Turingmaschine (TM)**: $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$, wobei:
  - Z: Zustandsmenge.
  - Σ: Eingabealphabet.
  - $\Gamma \supset \Sigma$: Arbeitsalphabet (Bandalphabet), mit $\square \notin \Sigma$.
  - δ: Übergangsfunktion.
  - $z_0 \in Z$: Startzustand.
  - $\square \in \Gamma \setminus \Sigma$: Leerzeichen (Blank).
  - $E \subseteq Z$: Endzustände.
- **Deterministische TM (DTM)**: $\delta : Z \times \Gamma \to Z \times \Gamma \times \{L, N, R\}$ (Bewegungsrichtungen: Links, Neutral, Rechts).
- **Nichtdeterministische TM (NTM)**: $\delta : Z \times \Gamma \to \mathcal{P}(Z \times \Gamma \times \{L, N, R\})$.
- **Arbeitsweise TM**:
  - Start: z_0, Kopf auf 1. Zeichen der Eingabe w, Band sonst mit $\square$.
  - Übergang $\delta(z, a) = (z', b, X)$: In Zustand z, liest $a \implies$ nach z', schreibt b, bewegt Kopf X.
  - Ende: M hält, sobald ein Zustand aus E erreicht wird.
- **Konfiguration**: Wort uzv, wobei $u, v \in \Gamma^*$ (Bandinhalt, z ist Zustand, Kopf auf 1. Zeichen von v).
- **Startkonfiguration bei Eingabe** w: $z_0 w$.
- **Übergangsrelation $\vdash_M$**: Beschreibt einen Schritt zwischen Konfigurationen.
- **Von TM M akzeptierte Sprache**: $L(M) = \{w \in \Sigma^* \mid z_0 w \vdash_M^* uzv$ für ein $z \in E$ und $u, v \in \Gamma^*\}$.
- **Äquivalenzen für Typ-0-Sprachen ($A \subseteq \Sigma^*$)**: Die folgenden Aussagen sind äquivalent:
  - A ist vom Typ 0.
  - $A = L(M)$ für eine Turingmaschine M.
  - A ist semi-entscheidbar.
  - A ist rekursiv-aufzählbar.
  - A ist Wertebereich einer totalen berechenbaren Funktion oder $A = \emptyset$.
  - A ist Wertebereich einer (eventuell partiellen) berechenbaren Funktion.
  - A ist Definitionsbereich einer berechenbaren Funktion.
  - $\exists$ entscheidbare Sprache B sodass $A = \{x \in \Sigma^* \mid \exists y : \langle x, y \rangle \in B\}$.
- **Korollar**: Typ-1-Sprachen sind echte Teilmenge der Typ-0-Sprachen.

UNENTSCHEIDBARE PROBLEME

- **Gödelisierung**: Kodierung von Turing-Maschinen durch Binärwörter $w \in \{0, 1\}^*$.
- M_w: Die Turingmaschine, die durch w kodiert wird. Falls w keine gültige Gödelisierung ist, sei M_w eine festgehaltene TM.
- **Spezielles Halteproblem (K)**: $K = \{w \in \{0, 1\}^* \mid M_w$ hält bei Eingabe $w\}$.
- **Allgemeines Halteproblem (H)**: $H = \{w\#x \mid M_w$ hält bei Eingabe $x\}$.
- **Eigenschaften von K, H**:
  - K und H sind rekursiv-aufzählbar.
  - K ist nicht entscheidbar.
  - H ist nicht entscheidbar.
- **Halteproblem auf leerem Band (H_0)**: $H_0 = \{w \mid M_w$ angesetzt auf leerem Band hält$\}$.
- **Eigenschaft von H_0**: H_0 ist nicht entscheidbar ($H \leq H_0$).

SATZ VON RICE

- Sei R die Klasse aller berechenbaren Funktionen.
- Sei $S \subseteq R$ mit $S \neq \emptyset$ und $S \neq R$.
- Sei $C(S) = \{w \mid$ die von M_w berechnete Funktion ist aus $S\}$.
- **Satz von Rice**: $C(S)$ ist nicht entscheidbar.
- **Korollare (Anwendungen)**: Die folgenden Sprachen sind nicht entscheidbar:
  - $\{w \mid M_w$ berechnet eine totale Funktion$\}$.
  - $\{w \mid M_w$ berechnet eine monotone Funktion$\}$.
  - $\{w \mid M_w$ berechnet eine konstante Funktion$\}$.
  - $\{w \mid M_w$ berechnet die Funktion $f(x) = x + 1\}$.

DIAGONALISIERUNG UND BERECHENBARKEIT

- **Diagonalisierung (Konzept)**: Konstruktion eines Elementes, das sich von jedem Element einer (abzählbaren) Aufzählung unterscheidet (mindestens an einer Stelle).
- **Anwendung** $\mathbb{R}$: Reelle Zahlen in $[0, 1]$ sind überabzählbar (Cantor's Diagonalargument).
- **Anwendung Berechenbare Funktionen**: Es gibt Funktionen $q : \mathbb{N} \to \mathbb{N}$, die nicht berechenbar sind. (Konstruktion einer Diagonalfunktion $q_{diag}(n)$, die sich von jeder n-ten berechenbaren Funktion q_n an Stelle n unterscheidet).
- **Anwendung Sprachen**: Es gibt Sprachen, die nicht rekursiv-aufzählbar sind (L_{diag}).
- **LOOP-Programme / WHILE-Programme**:
  - Es gibt eine totale, WHILE-berechenbare Funktion, die nicht LOOP-berechenbar ist (z.B. Ackermann-Funktion).
  - Beweisidee: Diagonalisierung über alle LOOP-Programme.