TIL 4 PG1

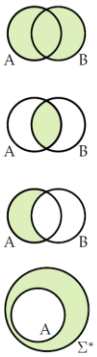
- **Alphabet**: endliche, nicht-leere Menge atomarer Symbole (Σ, T)
- **Wort über Σ**: endliche Folge von Symbolen aus Σ
- **Konkatenation**: Hintereinanderschreiben von Wörtern
- **Potenzbildung**: Verkettung eines Wortes w mit sich selbst $w^0 = \epsilon, w^n = ww^{n-1}$
- $\mathbf{\Sigma^+}$: Menger aller (nicht-leeren) Wörter übe Σ
- $\mathbf{\Sigma^*}$: $Menge\ aller\ Wörter\ (inklusive\ \epsilon)\ über\ \Sigma$
- **Formale Sprache**: beliebige Teilmenge L von $\Sigma^*, L \subseteq \Sigma^*$ Bsp.: $\{\}, \{\epsilon\}, \{(ab)^n \mid n \geq 0\}$
- $\boldsymbol{P(\Sigma^*)}$**:** Menge aller Sprachen $L \subseteq \Sigma^*$
- Operationen auf Sprachen
  - **Konkatenation**: $L_1 * L_2 = \{ w_1 * w_2 \mid w_1 \in L_1, w_2 \in L_2\}$
    - Bsp.: $\{ab, b\}\{a, bb\} = \{aba, abbb, ba, bbb\}$
  - **Potenzbildung:** $L^n = \{w_1..w_n \mid w_1, \dots, w_n \in L\},\ L^0 = \{\epsilon\}$
    - Bsp.: $\{ab, ba\}^2 = \{abab, abba, baab, baba\}$
  - **Kleene-Stern:** $L^* = \cup_{n \geq 0} L^n$
    - Bsp.: $\{01\}^* = \{\epsilon, 01, 0101, 010101, ..\} \neq \{0,1\}^*$
- Mengenoperationen & Rechenreglen für Sprachen

Vereinigung:
$A \cup B = \{x \in \Sigma^* \mid x \in A \text{ oder } x \in B\}$

Durchschnitt:
$A \cap B = \{x \in \Sigma^* \mid x \in A \text{ und } x \in B\}$
(A und B sind **disjunkt** wenn $A \cap B = \{\}$)

Differenz:
$A - B = \{x \in \Sigma^* \mid x \in A \text{ und } x \notin B\}$

Komplement:
$\overline{A} = \Sigma^* - A = \{x \in \Sigma^* \mid x \notin A\}$



$$A \cdot (B \cdot C) = (A \cdot B) \cdot C \quad \text{Assoziativität von} \cdot$$
$$A \cdot (B \cup C) = A \cdot B \cup A \cdot C \quad \text{Distributivität von} \cdot$$
$$(B \cup C) \cdot A = B \cdot A \cup C \cdot A \quad \text{Distributivität von} \cdot$$

$$(A \cup \{\varepsilon\})^* = A^* \qquad \{\varepsilon\} \cdot A = A$$
$$(A^*)^* = A^* \qquad A \cdot \{\varepsilon\} = A$$
$$A \cdot A^* = A^+ \qquad \{\} \cdot A = \{\}$$
$$A^* \cdot A = A^+ \qquad A \cdot \{\} = \{\}$$
$$A^+ \cup \{\varepsilon\} = A^*$$

- Anzahl der Elemente einer Menge M = |M| = **Kardinalität**
- Potenzmenge |P(A)| = $2^{|A|}$ → eine Menge mit n Elementen hat genau 2^n Teilmengen
- **gleichmächtig:** es existiert eine bijektive Abbildung zwischen 2 Mengen
- **abzählbar (unendlich)** = gleichmächtig mit ℕ
  - $\Sigma^*\ ist\ abzählbar\ falls\ \Sigma\ endllich$
- **überabzählbar =** unendliche, nicht abzählbare Menge

Unentscheidbarkeit

- Turingmaschinen
  - Band mit Eingabe, Lese-Schreib-Kopf auf Band → 3 Operationen: Lesen, Schreiben, Halten
  - Maschine kann verschiedene Zustände annehmen

Eine **Turingmaschine** ist ein Siebentupel

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F),$$

- Q endliche Menge von Zuständen,
- Σ endliche Menge der Eingabesymbole,
- Γ endliche Menge der Bandsymbole, $\Sigma \subset \Gamma$,
- $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R, S\}$ Übergangsfunktion,
- $q_0 \in Q$ Startzustand,
- $B \in \Gamma - \Sigma$ Blank-Symbol,
- $F \subseteq Q$ Menge von Endzuständen.

Eine *Turingmaschine* ist ein Achttupel

$$M = (Q, T, \Gamma, \delta, q_0, \{Z_0, Z_1, Z_2\}, B, F),$$

- Q endliche Menge von Zuständen,
- T Alphabet des Eingabebandes,
- Γ Alphabet des Arbeitsbandes,
- δ Übergangsfunktion,
- $q_0 \in Q$ Startzustand,
- $Z_0 \in \Gamma$ linkes Begrenzungssymbol auf Arbeitsband, Z_1, Z_2 Begrenzungssymbole des Eingabewortes,
- $B \in \Gamma$ Blanksymbol,
- $F \subseteq Q$ Menge von Endzuständen.

- **rekursiv aufzählbar**e Sprache
  - wird von einer (deterministischen) Turingmaschine akzeptiert
  - ist nicht überabzählbar
- **rekursiv/entscheidbar**e Sprache
  - TM **entscheidet** die Sprache: wird von einer (D)TM akzeptiert, die **immer hält**
  - ist $L\ und\ \bar{L}$ rekursiv aufzählbar → $L\ und\ \bar{L}$ ist rekursiv/entscheidbar
    - ist einer der beiden Sprachen nicht rekursiv aufzählbar, ist keine rekursiv/entscheidbar sein

deterministische Turingmaschinen (DTM) vs. **nichtdeterministische** Turingmaschinen (NTM)

- DTM: $\forall\ (q, X) \in Q\ x\ \Gamma$ existiert höchstens 1 Element $(q, X; p, Y, D) \in \delta\ (Übergangsfunktion)$
- NTM: $\delta(q, X) = \{(q_1, Y_1, D_1), \dots, (q_k, Y_k, D_k)\}$ → es gibt mehrere mögliche Übergänge
- zu jeder NTM gibt es eine DTM
  - die DTM hat eine quadratisch größere Speicherkomplexitätsschranke als NTM

Church-Turing-These

- kann man eine Sprache in endlichen Schritten exakt spezifizieren, gibt es eine TM, die diese akzeptiert

Berechenbarkeit

- Funktion f: $\Sigma^* \to \Sigma^*$, definiert für $\forall\ x \in \Sigma^*$ heißt **(total) berechenbar**, wenn
  - es eine TM gibt, die $\forall\ x$ mit der Eingabe x auf dem Band f(x) liefert/schreibt

Reduktion

Eine Sprache $A \subseteq \Sigma^*$ heißt **reduzierbar** auf die Sprache $B \subseteq \Gamma^*$, (bezeichnet mit $A \leq B$), wenn es eine (total) berechenbare Funktion $f : \Sigma^* \longrightarrow \Gamma^*$ so gibt, dass

für alle $w \in \Sigma^*$: $w \in A$ genau dann, wenn $f(w) \in B$.

In diesem Fall nennt man f eine **Reduktion** von A auf B.

Reduktion A auf B: $A \leq B$

- A nicht rekursiv aufzählbar → B nicht rekursiv aufzählbar
- B rekursiv aufzählbar → A rekursiv aufzählbar
- A nicht entscheidbar → B nicht entscheidbar
- B entscheidbar → A entscheidbar

Eigenschaften von Sprache

- … Teilmenge von rekursiv aufzählbaren Sprachen mit bestimmter Eigenschaft/Bedingung
- **triviale Eigenschaft**: Eigenschaft die für alle oder keine rekursiv aufzählbaren Sprachen besteht

Satz von Rice

- Jede nicht triviale Eigenschaft der rekursiv aufzählbaren Sprachen ist unentscheidbar

eine Turingmaschine ist **Normalform**, wenn

- nur 1 Endzustand → letzter Übergang von Gestalt $\delta(s, Z_2, Z_0) = (f, Z_0, S, R)..f\ einziger\ Endzustand$
- das Arbeitsband am Ende eines akzeptierenden Laufes der TM leer ist

TM Varianten

- **linear beschränkter Automat** (LBA)
  - TM, die auf dem Arbeitsband höchsten um einen linearen Faktor (* c + d) mehr Platz benötigt, als das Eingabewort lang ist; Speicherkomplexität liegt in O(n)
- **Kellerautomaten (PDA- Pushdown Automaten)**
  - TM, die auf dem Eingabeband nie nach links geht

    - und am Arbeitsband → wie Stack LIFO
        - links vom Schreibkopf keine Blanksymbole stehen
        - rechts vom Schreibkopf nur Blanksymbole stehen
    - zu jeder kontextfreien Sprache gibt es einen Kellerautomaten, der diese entscheidet
    - **Normalform des PDA**
        - Eingabewort wird durch Endzustand & leeren Keller akzeptiert
- **Endlicher Automat**
    - TM ohne Arbeitsband; Lesekopf geht nicht nach links zurück
    - Deterministische endliche Automaten (**DEA**)
    - Nichtdeterministischer endliche Automaten (**NEA)**
        - in einem Zustand von einem Eingabesymbol gibt es mehrere mögliche Folgezustände
        - Verfeinerung $\epsilon - NEA$
    - Determinisierung
        - zu jedem NEA gibt es einen äquivalenten DEA
    - Minimierung
        - zu jeder regulären Sprache kann man einen DEA mit minimaler Anzahl von Zuständen finden

Eine formale Sprache heißt

- 0 **rekursiv aufzählbar,** wenn Sie von
    - einer *Turingmaschine* akzeptiert wird
    - einer Typ-0-Grammatik erzeugt wird
- **monoton,** wenn Sie von
    - einer monotonen-Grammatik erzeugt wird
- 1 **kontextsensitiv**, wenn sie von
    - einem *linear beschränkten Automaten (LBA)* akzeptiert wird
    - einer Typ-1-Grammatik erzeugt wird
- 2 **kontextfrei**, wenn sie von
    - einem *Kellerautomaten* akzeptiert wird
    - einer Typ-2-Grammatik erzeugt wird
- 3 **regulär**, wenn sie von
    - einem *endlichen Automaten* akzeptiert wird
    - einer Typ-3-Grammatik erzeugt wird

Homomorphismen

- Abbildung von einem Alphabet auf ein anderes; **Homomorphismus: h: $\Sigma \rightarrow \Gamma^*$**
- induktive Erweiterung auf Wörter über Σ
    - $h(\epsilon) = \epsilon$
    - $h(wa) = h(w)h(a)\ \forall\ w \in \Sigma^*, a \in \Sigma$

Abschlusseigenschaften von Sprachen

- **reguläre Sprachen**
    - abgeschlossen
        - Vereinigung ∪, Konkatenation ·, Stern-Operator *, Plus-Operator, Komplement bzgl. Σ^*, Durchschnitt, Differenz, Spiegelung, Homomorphismen
- **kontextfreie Sprachen**
    - abgeschlossen
        - Vereinigung ∪, Konkatenation ·, SternOperator *, Homomorphismen, Schnitt mit regulären Sprachen
    - nicht abgeschlossen
        - Durschnitt und Komplement ($L_1 \cap\ L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$)

Grammatiken

Grammatik

Grammatik $G = \langle N, T, P, S \rangle$, wobei

N ... endliche Menge von Nonterminalen (Variablen)
T ... endliche Menge von Terminalsymbolen ($N \cap T = \{\}$)
$P \subseteq (N \cup T)^+ \times (N \cup T)^*$ ... Produktionen
$S \in N$ ... Startsymbol

- Gilt für alle Produktionen $\alpha \rightarrow \beta$

- **Typ-0**
  - unbeschränkt
- **monoton**
  - $|\alpha| \leq |\beta|$ → keine Produktion verkleinert ein Wort
- **Typ-1 (kontextsensitiv)**
  - $\alpha = uAv$ und $\beta = uwv$ für ein $A \in N, w \in (N \cup T)^{+} und\ u, v \in (N \cup T)^{*}$
  - d.h. w ist ein Wort aus Nonterminalen & Terminalen darf aber nicht leer (ϵ) sein
    u,v ebenfalls ein Wort aus Nonterminalen & Terminalen kann leer sein
- **Typ-2 (kontextfrei)**
  - A → β … $A \in N$
  - nur Nonterminale werden übersetzt; die Übersetzung ist unabhängig vom umliegenden Kontext
- **Typ-3 (regulär)**
  - A → aB oder A → ϵ für A, B $\in N$ und $a \in T$
  - jedes Nonterminal wird entweder ins Leerwort übersetzt, oder in ein Terminal gefolgt von einem Nonterminal; → Ableitung „baut" immer rechts weiter

gsm-Abbildungen

- **gsm (generalized sequential machine)**: endlicher Automat mit Ausgabe
- **gsm-Abbildung**: $f_M: \Sigma^* \to \mathrm{P}(\Gamma^*)$
  - $f_M(w)$ = alle Ausgabewörter v, die sich bei Analyse des Eingabewortes w auf einem Pfad vom Startknoten zu einem Endknoten aus F ergeben
  - Erweiterung auf Sprachen $L \subseteq \Sigma^*$: $f_M(L) := \{v \in \Gamma^* \mid v \in f_M(w)\ für\ ein\ w \in L\}$
  - $\boldsymbol{\epsilon - frei}$: M & f_M → $\delta: Q\ x\ \Sigma \to Q\ x\ \Gamma^+$
  - **deterministisch:** M & f_M → für jedes Paar (q, a) gibt es höchstens ein $(q, a; p, v) \in \delta$
  - jeder Homoorphismus ist eine einfache gsm-Abbildung, die nur einen Zustand benötigt

Sprachfamilien

- $\mathcal{L}_i(\Sigma)$= Menger aller formalen Sprachen $L \subseteq \Sigma^*$ die von einer Grammatik vom Typ *i* erzeugt werden können
- $\mathcal{L}_i$= Familie der formalen Sprachen, die von einer Typ-iGrammatik erzeugt werden können

Chomsky-Hierarchie & Abschlusseigenschaft

Sprachfamilie	Grammatiktyp	Maschinenmodell
$\mathcal{L}_0$ (rekursiv aufzählb.)	unbeschränkt	Turingmaschine (= EA + RAM)
$\mathcal{L}_1$ (kontextsensitiv)	kontextsensitiv monoton	Linear beschränkter Aut. (= EA + beschr.RAM)
$\mathcal{L}_2$ (kontextfrei)	kontextfrei	Kellerautomat (= EA + Stack)
$\mathcal{L}_3$ (regulär)	regulär	endlicher Automat (EA)

	$\mathcal{L}_3$	$\mathcal{L}_2$	$\mathcal{L}_1$	$\mathcal{L}_0$
Vereinigung	ja	ja	ja	ja
Konkatenation	ja	ja	ja	ja
Kleenescher Stern	ja	ja	ja	ja
Komplement	ja	nein	ja	nein
Durchschnitt	ja	nein	ja	ja
Durchschnitt mit reg. Mengen	ja	ja	ja	ja
Homomorphismen	ja	ja	nein	ja
ε-freie Homomorphismen	ja	ja	ja	ja
inverse Homomorphismen	ja	ja	ja	ja
gsm-Abbildungen	ja	ja	nein	ja
ε-freie *gsm*-Abbildungen	ja	ja	ja	ja

kontextfreie Grammatik

- **eindeutig**
  - zu jedem ableitbaren Terminalwort gibt es genau eine Linksableitung → ansonsten **mehrdeutig**
- **inhärent mehrdeutig**
  - heißt eine Sprache, wenn sie nur von mehrdeutigen Grammatiken erzeugt werden kann
- **ϵ-Produktionen**: Produktionen der Form $A \to \epsilon \quad A \in Nonterminale$
- **Einheitsproduktionen:** Produktionen der Form $A \to B \quad A, B \in Nonterminale$
- **Nutzlose Produktionen:** Produktionen, die vom Startsymbol nicht erreichbar sind oder kein $w \in T^*$ ableitbar ist
- **reduzierte Grammatik:** kontextfreie Grammatik ohne nutzlos-, ϵ- und Einheitsproduktionen
- **Chomsky Normalform**
  - alle Produktionen der Form: $A \to BC$ oder $A \to a \qquad (a \in T; A, B, C \in N)$
  - sofern S nicht auf der rechten Seite einer Produktion ist $S \to \epsilon$ erlaubt
  - zu jeder k.f. Grammatik kann man eine äquivalente Grammatik in Chomsky Normalform konstruieren
- **Greibach Normalform**
  - alle Produktionen der Form: $A \to aw \quad (a \in T, A \in N, W \in N^*)$
  - **erweiterte Greibach Normalform**
    - alle Produktionen der Form: $A \to aw \quad (a \in T, A \in N, w \in (N \cup T)^*)$
  - sofern S nicht auf der rechten Seite einer Produktion ist $S \to \epsilon$ erlaubt
  - zu jeder k.f. Grammatik kann man eine äquivalente Grammatik in (erweiterter) Greibach NF konstruieren

- **Dyck-Sprache** (Klammersprache)
  - Sprache aller korrekt geformten Klammerausdrücken
  - Definition: D_n $über$ $\Sigma_n = \{ x_1, \overline{x_1}, x_2, \overline{x_2}, \dots, x_n, \overline{x_n}\}$ ein Alphabet mit 2n Symbolen, n … Anzahl Klammerpaare
  - $\epsilon \in D_n$
  - Falls $w \in D_n$, so ist auch $x_i w \overline{x_i} \in d_n$
  - Falls u, v ∈ D_n so ist auch uv ∈ D_n
  - wird x_i als öffnende & $\overline{x_i}$ als schließende Klammer interpretiert, beschreibt die Dycksprache die **Menge aller korrekten Klammerausdrücke**

Satz von **Chomsky-Schützenberger**

- jede kontextfreie Sprache ist homomorphes Bild des Durchschnitts einer regulären Sprache mit einer Dyck-Sprache
- d.h. ist eine Sprache L kontextfrei, kann man eine Dycksprache D_n & einen Regulären Ausdruck R angeben, so dass $\boldsymbol{h(R \cap D_n) = L}$
- dabei bestimmt der RA nur die Reihenfolge der Zeichen, nicht ihre Anzahl oder evtl. Abhängigkeiten
- mit der Dycksprache können, dann Schachtelungen & Klammeregeln erkannt & definiert werden
- durch den Homomorphismus kann nun die Schnittmenge auf das Zielalphabet abgebildet werden
  - dabei könne auch Elemente in der Zielsprache nicht gebraucht werden $h(a_1) = \epsilon$

Komplexitätstheorie

- A auf B polynomiell reduzierbar $A \leq_p B)$ → f: $\Sigma^* \to \Sigma^*$ $so\ dass: x \in A \Leftrightarrow F(x) \in B$
- Problemklassen
  - **P**
    - polynomiell lösbar
    - Komplement abgeschlossen (Akzeptieren/Verwerfen vertauschbar)
  - **NP**
    - Lösung polynomiell überprüfbar (mit polynomiellem Zertifizierer)
  - **NP-Schwer**
    - jedes NP-Problem lässt sich in Polynomialzeit auf ein NP-Schweres Problem reduzieren
  - **NP-vollständig**
    - NP-Probleme können darauf reduziert werden & es ist polynomiell überprüfbar
    - → liegt in NP-Schwer & NP
    - (nur möglich für Ja/Nein-Probleme)
  - **Co-NP**
    - enthält die Sprachen/Probleme, deren Komplement in NP ist
  - **PSPACE**
    - enthält die Sprachen, die von einer polynomiell platzbeschränkten DTM entscheidbar sind
  - **NPSPACE**
    - enthält die Sprachen, die von einer polynomiell platzbeschränkten NTM entscheidbar sind

Satz von Savitch

- ein auf einer NTM unter bestimmten Platzbeschränkung gelöstes Problem, kann auf einer DTM mit einer quadratisch höheren Platzschranke gelöst werden → **PSPACE = NPSAPCE**