# Einführung in die Logik

## Wahrheitstabellen:

| ¬ | |
|---|---|
| T | F |
| F | T |

| ∧ | T | F |
|---|---|---|
| T | T | F |
| F | F | F |

| ∨ | T | F |
|---|---|---|
| T | T | T |
| F | T | F |

| → | T | F |
|---|---|---|
| T | T | F |
| F | T | T |

## Präzedenzen:

1. ¬
2. ∧, ∨
3. →, <->

## Konsequenzrelation:

Aus der Wahrheit der Prämisse $A_1$ … $A_n$ , folgt die Wahrheit der Konklusion B: $p \land \neg q \models q$

| p | q | p ∧ ¬ q | |
|---|---|---|---|
| 0 | 0 | 0 | F -> F = T |
| 0 | 1 | 0 | F -> T = T |
| 1 | 0 | 0 | F -> F = T |
| 1 | 1 | 0 | T -> T = T |

## Tautologie und Erfüllbarkeit:

| | |
|---|---|
| **Tautologie** | **Eine aussagenlogische Formel ist dann eine Tautologie, wenn ALLE Interpretationen auf Wahr (True) enden.** |
| **Erfüllbar** | **Eine aussagenlogische Formel ist dann erfüllbar, wenn mindestens EINE Interpretation auf Wahr (True) endet.** |
| **Unerfüllbar** | **Eine aussagenlogische Formel ist dann unerfüllbar, wenn KEINE Interpretation auf Wahr (True) endet.** |

## Äquivalenz von aussagenlogischen Formeln:

Zwei aussagenlogische Formeln sind dann Äquivalent, wenn A |= B & B |= A gilt:

$$p \wedge (q \wedge r) \models (p \wedge q) \wedge r$$

| p | q | r | $p \wedge (q \wedge r)$ | $(p \wedge q) \wedge r$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

## Konjunktion (∧) und Disjunktion (∨):

1. DNF (Disjunktive Normalform): A ist in DNF, wenn A eine Disjunktion von Konjunktionen ist. Die DNF wird von Wahren Endergebnissen gebildet (siehe Beispiel unten).

2. KNF (Konjunktive Normalform): A ist in KNF, wenn A eine Konjunktion von Disjunktionen ist. Die KNF wird von Falschen Endergebnissen gebildet (siehe Beispiel unten).

| A | B | C | A ∧ B | B ∨ C | (A ∧ B) ∧ (B ∨ C) | MIN | MAX |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | | A ∨ B ∨ C |
| 0 | 0 | 1 | 0 | 1 | 0 | | A ∨ B ∨ ¬C |
| 0 | 1 | 0 | 0 | 1 | 0 | | A ∨ ¬B ∨ C |
| 0 | 1 | 1 | 0 | 1 | 0 | | A ∨ ¬B ∨ ¬C |
| 1 | 0 | 0 | 0 | 0 | 0 | | ¬A ∨ B ∨ C |
| 1 | 0 | 1 | 0 | 1 | 0 | | ¬A ∨ B ∨ ¬C |
| 1 | 1 | 0 | 1 | 1 | 1 | A ∧ B ∧ ¬C | |
| 1 | 1 | 1 | 1 | 1 | 1 | A ∧ B ∧ C | |

**KNF: (A ∨ B ∨ C) ∧ (A ∨ B ∨ ¬C) ∧ (A ∨ ¬B ∨ C) …**

**DNF: (A ∧ B ∧ ¬C) ∨ (A ∧ B ∧ C)**

# Einführung in die Algebra

## Wahrheitstabellen binäre Algebra:

| ¬ | |
|---|---|
| T | F |
| F | T |

->

| ~ | |
|---|---|
| T | F |
| F | T |

| ∧ | T | F |
|---|---|---|
| T | T | F |
| F | F | F |

->

| * | T | F |
|---|---|---|
| T | T | F |
| F | F | F |

| ∨ | T | F |
|---|---|---|
| T | T | T |
| F | T | F |

->

| + | T | F |
|---|---|---|
| T | T | T |
| F | T | F |

# Theorie der formalen Spachen

Allgemein gilt, dass ein Wort der Länge 0 das Leerwort ist. Endliche, nicht leere Menge von atomaren Symbolen wird das Alphabet ∑ (Sigma) genannt.

## Formale Sprachen:

Seien L und M formale Sprachen über dem Alphabet ∑, dann wird der Durchschnitt als L ∩ M geschrieben und die Vereinigung als L ∪ M.

Konkatenation mit dem Leerwort:

Ob das Leerwort am Anfang hin konkateniert wird, oder am Schluss, macht keinen Unterschied auf die Gleichung siehe Beispiel: $\varepsilon \circ w = w = w \circ \varepsilon$

Wenn wir aber zwei Wörter konkatenieren, die nicht ident sind, macht die Konkatenationsreihenfolge einen unterschied!

Der Kleene-Stern (auch Abschluss genannt) über einem beliebigen Alphabet L ist wie folgt definiert:

$$L^* = \bigcup_{k \geqslant 0} L^k = \{x_1 \cdots x_k \mid x_1, \ldots, x_k \in L \text{ und } k \geqslant 0\}$$

# Grammatiken und formale Sprachen

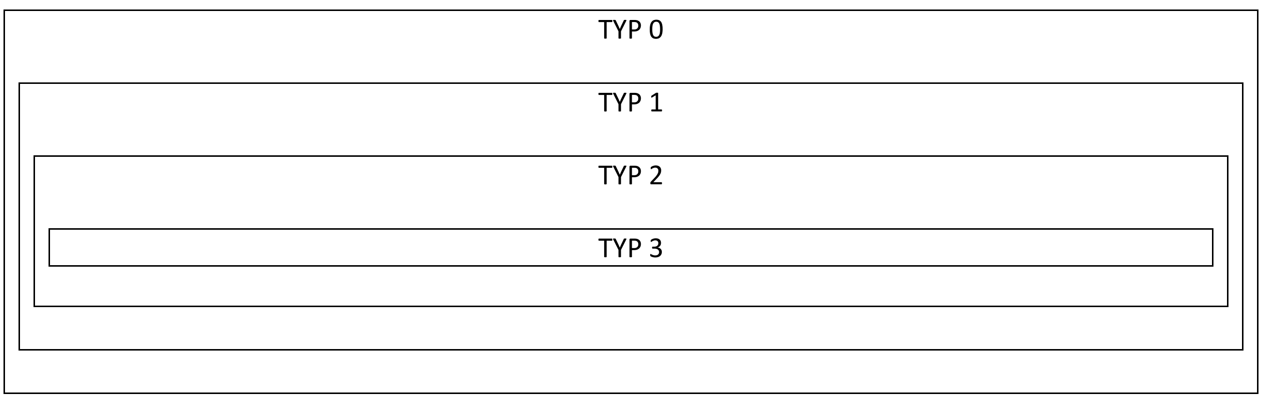
Eine Grammatik G ist ein Quadrupel G = (V, ∑, R, S) , wobei die Variablen für folgende Ausdrücke stehen:

| | |
|---|---|
| **V** | **Eine endliche Menge an Variablen (Terminale).** |
| **∑** | **Ein Alphabet, dass terminiert.** |
| **R** | **Eine endliche Menge an Regeln.** |
| **S** | **S ∈ V -> das Startsymbol von G** |

## Chomsky Hierarchie:

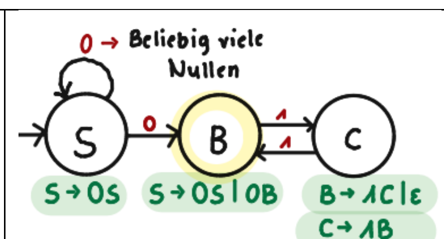
Die Chomsky Hierarchie beschäftigt sich mit der Einschränkung von Grammatiken in die folgenden 4 Typen:

| |
|---|
| **TYP 0:** keine Einschränkung |
| **TYP 1 (kontextsensitiv):** Typ0 inkl. Längenbeschränkung w1 -> w2 muss gelten \|w1\| <= \|w2\| mit der Ausnahme, dass S das Leerwort (ε) impliziert. |
| **TYP 2 (kontextfrei):** 1x V -> Kombination aus V & ∑, „A -> BC“ ist ein gültiger Ausdruck, „B -> abCab“ ist auch gültig aber „aB -> abCabA“ ist KEIN gültiger Ausdruck! |
| **TYP 3 (regulär):** 1x V -> Kombination aus V & ∑<br>• linksregulär: A -> Ba<br>• rechtsregulär: A -> aB |




**TYP 3 ⊂ TYP 2 ⊂ TYP 1 ⊂ TYP 0**

## Reguläre Sprachen:

| | | | |
|---|---|---|---|
| G = (V, ∑, R, S)<br>V = {B, C, D, S}<br>∑ = {a, b, c}<br>R … Produktionsregeln<br>S … Startsymbol | **LINKSREGULÄR**<br>1. A -> Ba<br>2. A -> a<br>3. A -> ε | **RECHTSREULÄR**<br>1. A -> aB<br>2. A -> a<br>3. A -> ε |  |

# Berechenbarkeitstheorie

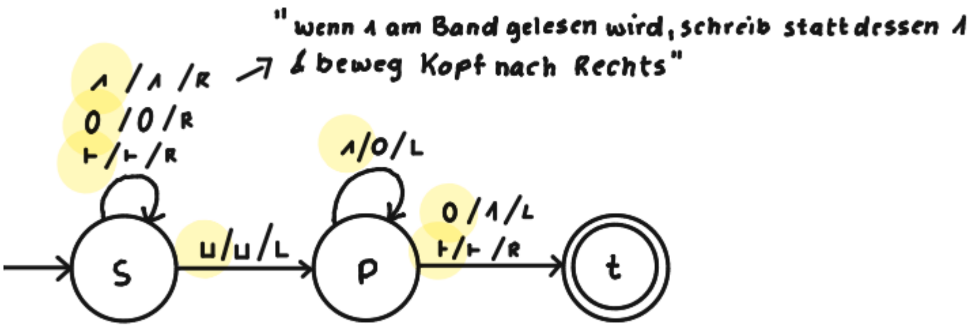
## Turingmaschine (M):

M = ({s, p, t, r}, {0, 1}, { |-, |_|, 0, 1}, δ,  s, t, r)

Zustandstabelle:

| | \|- | 0 | 1 | \|_\| |
|---|---|---|---|---|
| s | (s, \|-, R) | (s, 0, R) | (s, 1, R) | (p, \|_\|, L) |
| p | (t, \|-, R) | (t, 1, L) | (p, 0, L) | * |

Zustandsdiagramm:




# Verifikation nach HOARE

$$[z] \quad \frac{}{\{Q\{x \mapsto t\}\}\ x := t\ \{Q\}} \qquad [a] \quad \frac{\{Q'\}\ P\ \{R'\}}{\{Q\}\ P\ \{R\}}\ Q \models Q', R' \models R$$

$$[s] \quad \frac{\{Q\}\ P_1\ \{R\} \quad \{R\}\ P_2\ \{S\}}{\{Q\}\ P_1; P_2\ \{S\}} \qquad [w] \quad \frac{\{I \wedge B\}\ P\ \{I\}}{\{I\}\ \text{while } B \text{ do } P \text{ end } \{I \wedge \neg B\}}$$