# Turingmaschine - Motivation und Definition

## Church'sche These

Eine Funktion ist im intuitiven Sinne berechenbar gdw. sie Turing-berechenbar ist.

sehr eingeschränktes Modell

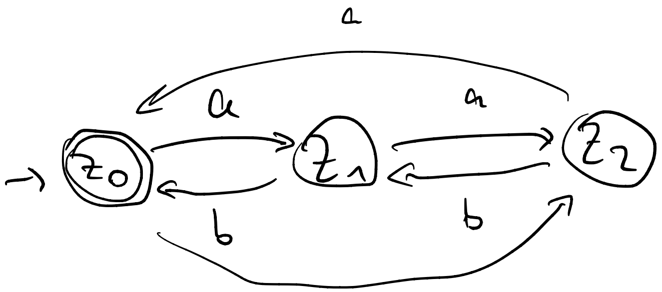
~> leichter zu analysieren,
formale Beweise dafür, dass gewisse Funktionen nicht berechenbar sind

stärkere Aussagen als im Abschnitt zu NP!
werden beweisen (!): gewisse Problem sind nicht algorithmisch lösbar

# Wie arbeiten Turing-Maschinen?

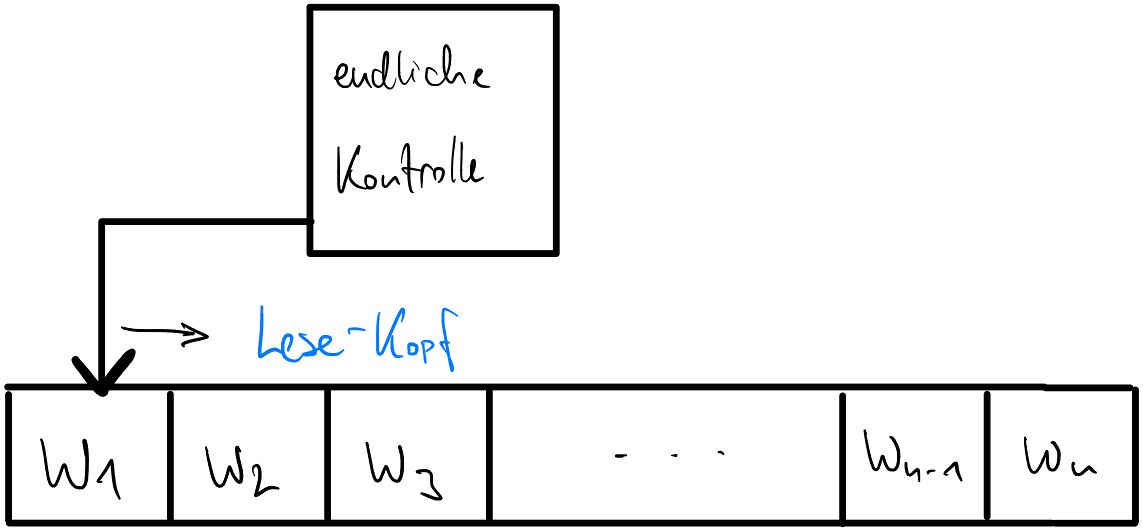
## endlicher Automat

M:



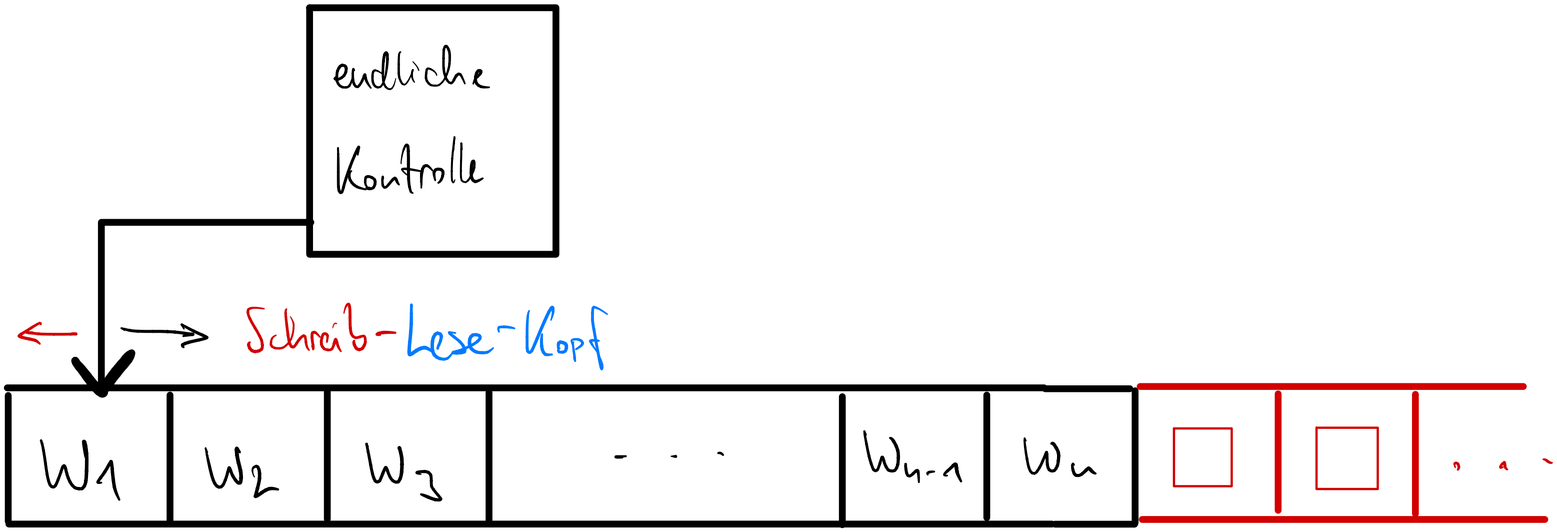
$$L(M) = \{ w \in \{a,b\}^* \mid |w|_a \equiv |w|_b \bmod 3 \}$$

schematische Darstellung der Funktionsweise:



# Turing-Maschine

schematische Darstellung der Funktionsweise:



# Formale Umsetzung

zur Erinnerung: DEA $A=(\Sigma, Q, \delta, q_0, F)$

| | |
|---|---|
| $\Sigma$ | Eingabealphabet (endl., nicht-leere Menge) |
| $Q$ | Menge von Zuständen (endl., nicht-leere Menge) |
| $\delta : Q \times \Sigma \to Q$ | Überführungsfunktion |
| $q_0 \in Q$ | Startzustand |
| $F \subseteq Q$ | Menge von Endzuständen |

- Arbeitsalphabet
- Blank-Symbol
- bei Übergängen: Angabe der Kopfbewegung und des geschriebenen Symbols

Eine Turing-Maschine (TM) ist ein Tupel

$(Q, \Sigma, \Gamma, \delta, q_0, acc, rej, \square)$, wobei

| | |
|---|---|
| $Q$ | Menge von Zuständen (endl., nicht-leere Menge) |
| $\Sigma$ | Eingabealphabet (endl., nicht-leere Menge) |
| $\Gamma \supseteq \Sigma$ | Arbeitsalphabet |
| $\delta : Q \times \Gamma \to Q \times \Gamma \times \{+1, -1\}$ | Überführungsfunktion |
| $q_0 \in Q$ | Startzustand |
| $acc, rej \in Q$ | akzeptierender und ablehnender Zustand |
| $\square \in \Gamma$ | Blank-Symbol |

# Arbeitsweise von Turing-Maschinen

**Konfiguration:** $(q, x, i) \in Q \times \Gamma^* \times \mathbb{N}$

$$\delta(q, a) = (q', b, \pm 1)$$

$$(q, xay, i) \Rightarrow_M (q', xby, i \pm 1)$$

$$\Rightarrow_M^*$$

Sei $w \in \Sigma^*$

Startkonf. von $M$ auf Eingabe $w$: $(q_0, w, 0)$

$M$ akzeptiert $w$, falls $(q_0, w, 0) \Rightarrow^* (acc, x, i)$ für gewisse $x, i$.

$M$ lehnt $w$ ab, falls $(q_0, w, 0) \Rightarrow^* (rej, x, i)$ für gewisse $x, i$.

# Entscheidbarkeit

Turing-Maschine $M$ entscheidet Sprache $L \subseteq \Sigma^*$, falls $M$ alle Wörter in $L$ akzeptiert und alle Wörter in $\overline{L} = \Sigma^* \setminus L$ ablehnt.

Sprache $L$ ist entscheidbar, wenn es eine TM $M$ gibt, die sie entscheidet.

# Beispiel zu Entscheidbarkeit

$L = \{ w\#w \mid w \in \{0,1\}^* \}$

Konstruiere TM $M = (Q, \Sigma, \Gamma, \delta, q_0, acc, rej, \square)$, die $L$ entscheidet.

$\cancel{0}\cancel{0}\cancel{1}\cancel{0}\cancel{1}\cancel{0}\#\cancel{0}\cancel{0}\cancel{1}\cancel{0}\cancel{1}\cancel{0}$

$Q = \{q_0, seek0_a, seek0_b, seek1_a, seek1_b, back_a, back_b, check, acc, rej\}$

$\Sigma = \{0, 1, \#\}$

$\Gamma = \Sigma \cup \{\square, \$\}$

$\delta$:

$(q_0, 0) \rightarrow (seek0_a, \$, +1)$

$(q_0, 1) \rightarrow (seek1_a, \$, +1)$

$(q_0, \#) \rightarrow (check, \#, +1)$

$(seek0_a, x) \rightarrow (seek0_a, x, +1)$ für $x \in \{0,1\}$

$(seek0_a, \#) \rightarrow (seek0_b, \#, +1)$

$(seek0_b, \$) \rightarrow (seek0_b, \$, +1)$

$(seek0_b, 0) \rightarrow (back_a, \$, -1)$

$(\text{seek1}_a, x) \to (\text{seek1}_a, x, +1)$ für $x \in \{0,1\}$
$(\text{seek1}_a, \#) \to (\text{seek1}_b, \#, +1)$

$(\text{seek1}_b, \$) \to (\text{seek1}_b, \$, +1)$
$(\text{seek1}_b, 1) \to (\text{back}_a, \$, -1)$

$(\text{back}_a, \$) \to (\text{back}_a, \$, -1)$
$(\text{back}_a, \#) \to (\text{back}_b, \#, -1)$
$(\text{back}_b, x) \to (\text{back}_b, x, -1)$ für $x \in \{0,1\}$
$(\text{back}_b, \$) \to (q_0, \$, +1)$

$(\text{check}, \$) \to (\text{check}, \$, +1)$
$(\text{check}, \square) \to (\text{acc}, \square, -1)$
zusätzlich: $\delta(q, a) = (\text{rej}, a, +1)$ f.a. $(q,a)$, für die $\delta$ noch nicht definiert ist.

$\delta$: $(q_0, 0) \to (\text{seek}0_a, \$, +1)$
$(q_0, 1) \to (\text{seek}1_a, \$, +1)$
$(q_0, \#) \to (\text{check}, \#, +1)$

$(\text{seek}0_a, x) \to (\text{seek}0_a, x, +1)$ für $x \in \{0,1\}$
$(\text{seek}0_a, \#) \to (\text{seek}0_b, \#, +1)$

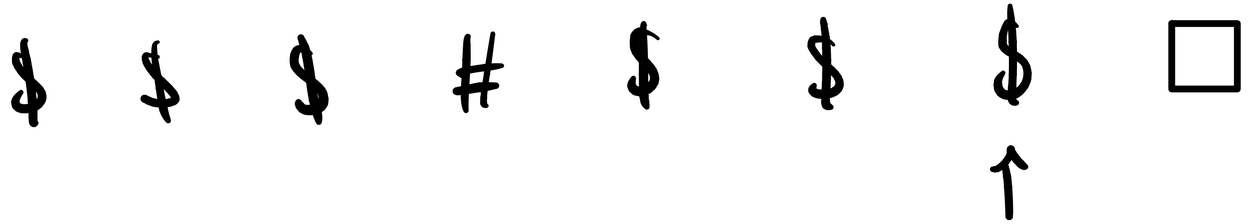
$(\text{seek}0_b, \$) \to (\text{seek}0_b, \$, +1)$
$(\text{seek}0_b, 0) \to (\text{back}_a, \$, -1)$

$(\text{seek}1_a, x) \to (\text{seek}1_a, x, +1)$ für $x \in \{0,1\}$
$(\text{seek}1_a, \#) \to (\text{seek}1_b, \#, +1)$

$(\text{seek}1_b, \$) \to (\text{seek}1_b, \$, +1)$
$(\text{seek}1_b, 1) \to (\text{back}_a, \$, -1)$

$(\text{back}_a, \$) \to (\text{back}_a, \$, -1)$
$(\text{back}_a, \#) \to (\text{back}_b, \#, -1)$
$(\text{back}_b, x) \to (\text{back}_b, x, -1)$ für $x \in \{0,1\}$
$(\text{back}_b, \$) \to (q_0, \$, +1)$

$(\text{check}, \$) \to (\text{check}, \$, +1)$
$(\text{check}, \square) \to (\text{acc}, \square, -1)$



acc

✓

# Berechenbarkeit

Ersetze akzeptierenden und ablehnenden Zustand durch einen Zustand halt.

Wird dieser erreicht, stoppt die Maschine.

Ausgabe ist der Bandinhalt (ohne □ am Ende).

$f: \Sigma^* \to \Delta^*$

f ist berechenbar, falls es eine TM M gibt, sodass f.a. $x \in \Sigma^*$ M bei Eingabe $x$ die Ausgabe $f(x)$ erzeugt.

# Beispiel

$f(\langle n\rangle_2) = \langle n+1\rangle_2$

1 1 1 0 1 0 1

$M = (Q, \Sigma, \Gamma, \delta, q_0, halt, \square)$

$Q = \{q_0, halt\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \square\}$

$\delta:$ $(q_0\ 0) \to (halt, 1, +1)$

$(q_0, 1) \to (q_0, 0, +1)$

$(q_0, \square) \to (halt, 1, +1)$

# Semi-Entscheidbarkeit

M akzeptiert eine Sprache $L \subseteq \Sigma^*$, falls M genau die Wörter in L akzeptiert.

Sprache L heißt semi-entscheidbar, wenn es TM M gibt, die L akzeptiert.

# Beispiel

Collatz-Folge für Zahl $n \in \mathbb{N}$:

$$(a_i)_{i \in \mathbb{N}} \text{ mit } a_i = \begin{cases} n, & \text{falls } i=0 \\ a_{i-1}/2, & \text{falls } a_{i-1} \text{ gerade} \\ 3a_{i-1}+1, & \text{sonst} \end{cases}$$

unbekannt, ob die Folge f.a. $n$ irgendwann die Zahl 1 erreicht

$$L_{Collatz} = \{ w \in \{0,1\}^* \mid \text{Collatzfolge für } val(w) \text{ erreicht irgendwann } 1 \}$$

$L_{Collatz} = \{ w \in \{0,1\}^* \mid$ Collatzfolge für val(w) erreicht irgendwann 1 $\}$

gegeben: Folgenglied $a_{i-1}$

berechne $a_i$

Fall 1: $a_{i-1}$ endet auf 0 → Lösche LSB

Fall 2: $a_{i-1}$ endet nicht auf 0 ⇝ berechne Bit für Bit $3a_{i-1}$

anschließend: inkrementieren

---

## **Semi-Entscheidungs-Algorithmus für $L_{Collatz}$**

Eingabe $w \in \{0,1\}^*$

Berechne sukzessive Folgenglieder $a_0, a_1, a_2, \ldots$ für $n = val(w)$

Wird 1 gefunden: Akzeptiere

(sonst implizit Endlosschleife)

# Variationen der Turingmaschine

Modell der TM ist sehr robust: kleinere Änderungen an der Definition ändern nichts an der Mächtigkeit.

verschiedene gängige Definitionen, führen zu gleichen Begriffen der Entscheidbarkeit und Berechenbarkeit

**Variation: $\delta$ darf partiell sein.**

$\delta(q, a)$ nicht definiert $\rightsquigarrow$ wird a in Zustand q gelesen, stürzt die Maschine ab

Klar: Jede Sprache, die mit ursprünglicher Def. entscheidbar ist, ist auch mit dieser Variante entscheidbar

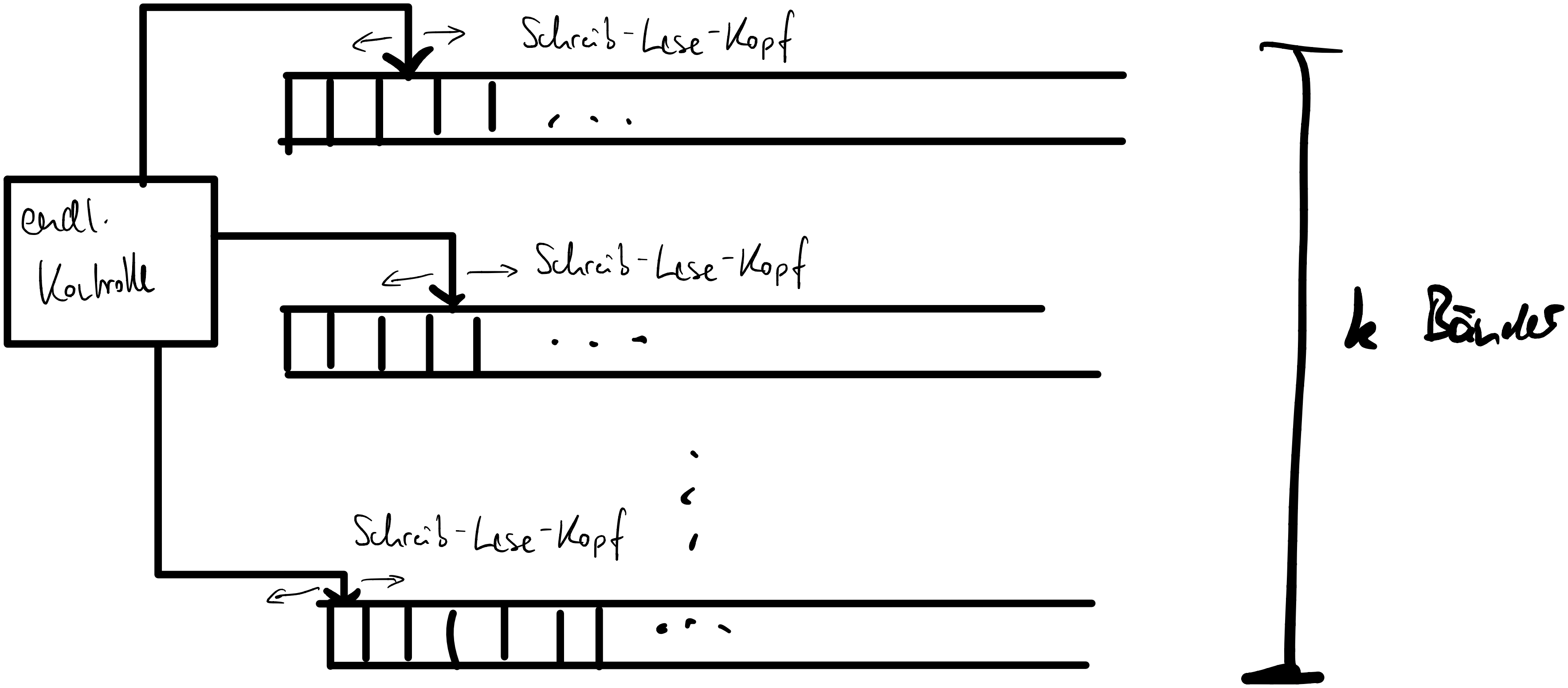
Umgekehrt: Sei L entscheidbar nach dieser Def. mit Maschine M.

wissen: M stürzt auf keiner Eingabe ab.

Modifiziere M, indem $\delta(q, a) = (q, a, +1)$ gesetzt wird, sofern vorher nicht definiert.

Modifizierte Maschine entscheidet L nach ursprünglicher Def.

# Mehrband - Turing - Maschine



$$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{+1, -1\}^k$$

Klar: entscheidbar mit gewöhnlicher TM f. a. k

$\Rightarrow$ entscheidbar mit k-Band-TM

## **Umgekehrt**

entscheidbar mit k-Band-TM ⇒ entscheidbar mit gewöhnlicher TM

f.a. k

3-Band-TM:

Band 1: $\underset{\uparrow}{u_1}$ $u_2$ $u_3$ ...

Band 2: $v_1$ $v_2$ $\underset{\uparrow}{v_3}$ ...

Band 3: $w_1$ $\underset{\uparrow}{w_2}$ $w_3$ ...

⇝ gewöhnliche TM: $(\underline{u_1}, v_1, w_1)$ $(u_2, v_2, \underline{w_2})$ $(u_3, \underline{v_3}, w_3)$ ...

ein Schritt der 3-Band-TM:

- Laufe zu Kopfposition auf Band 1, 2, 3, speichere Symbole im Zustand
- Laufe zu Kopfpositionen auf Band 1, 2, 3, modifiziere gemäß $\delta$
- wechsle Zustand gemäß $\delta$, Laufe zu erster Bandzelle

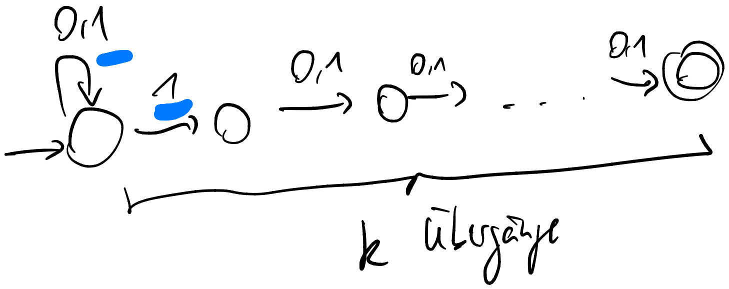
# Komplexität

Laufzeit bei Turing-Maschine M: Anzahl von Übergängen $\Rightarrow_M$ bis die Maschine hält

P: Klasse der Sprachen, die von einer TM in poly. Zeit entschieden werden können.

NP nichtdeterministische Polynomialzeit

NEA für "k-letztes Symbol ist 1" (über $\{0,1\}$)



deterministisch: $\delta: Q \times \Sigma \to Q$

nichtdeterministisch: $\delta: Q \times \Sigma \to P(Q)$

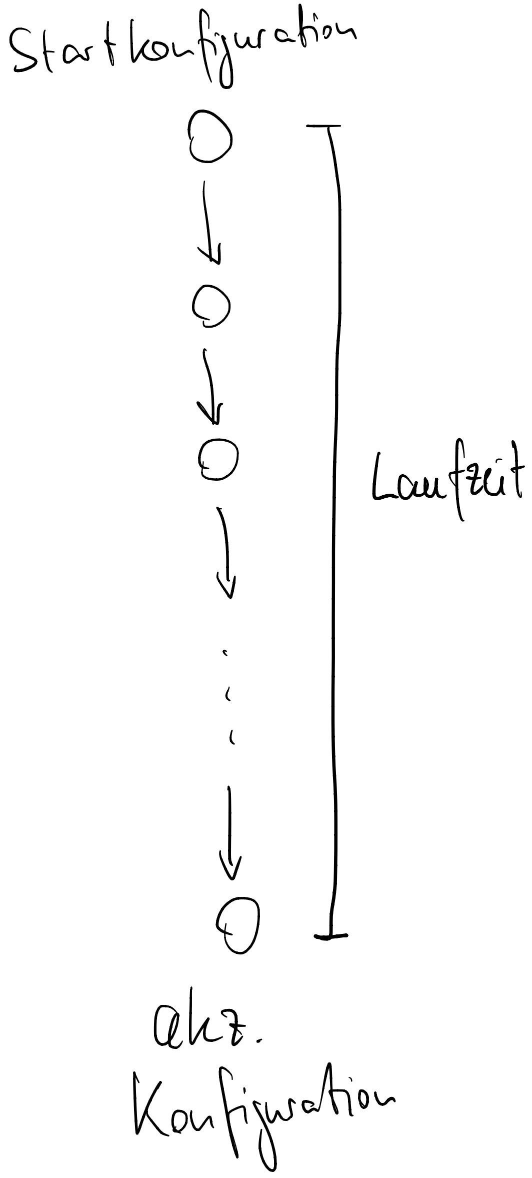
Eine nichtdeterministische Turing-Maschine (NTM) ist ein Tupel

$(Q, \Sigma, \Gamma, \delta, q_0, acc, rej, \square)$, wobei

| | |
|---|---|
| $Q$ | Menge von Zuständen (endl., nicht-leere Menge) |
| $\Sigma$ | Eingabealphabet (endl., nicht-leere Menge) |
| $\Gamma \supseteq \Sigma$ | Arbeitsalphabet |
| $\delta : Q \times \Gamma \to P(Q \times \Gamma \times \{+1, -1\})$ | Überführungsfunktion |
| $q_0 \in Q$ | Startzustand |
| $acc, rej \in Q$ | akzeptierender und ablehnender Zustand |
| $\square \in \Gamma$ | Blank-Symbol |

## deterministisch

Berechnung ist ein Pfad



## nichtdeterministisch

Berechnungsbaum