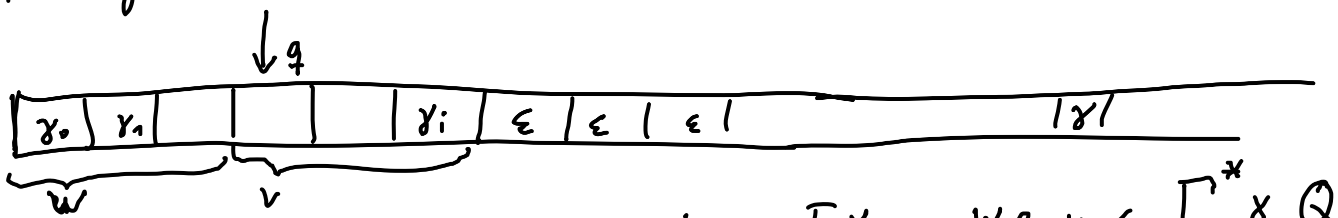
30. 11. 2011

Jede Boolesche Algebra lässt sich in eine Potenzmengenalgebra einbetten
$(B, 0, 1, \sqcap, \sqcup, ')$ $\qquad (\mathcal{P}(X), \emptyset, X, \cap, \cup, x \to X \setminus x)$

Turingmaschine



<u>Def</u> Eine Konfiguration ist eine ~~Wort~~ Folge $wqv \in \Gamma^* \times Q \times \Gamma^*$.
Die Bedeutung ist $w \in \Gamma^*$ steht auf dem Turingband strikt links vom Lesekopf, $v \in \Gamma^*$ steht unter dem Lesekopf beginnend rechts vom Lesekopf.
Auf dem Turingband stehen nach der Konfiguration nur noch ε's.

$C_0 = qw$ Anfangskonf. w: Input.

Länge der Konf $\leq |w| +$ Rechenschritte

$$C_0' = q\,w\,\underbrace{\varepsilon \ldots \varepsilon}_{\varepsilon \notin \Sigma}, \qquad w \in \Sigma^*$$

wäre auch eine Anfangskonf.

Rechter „Rand"

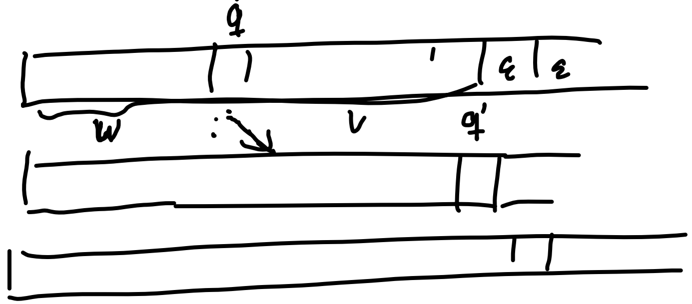
C_0

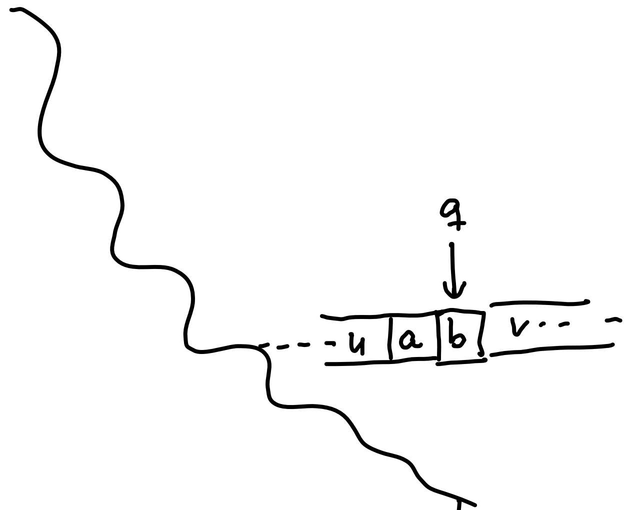
C_1

C_2

C_3

$\vdots$

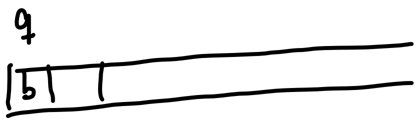




<u>Def</u>: Seien $u, v \underset{(+)}{\in} \Gamma^*$, $a, b, c \in \Gamma$. Eine Konfiguration $u\,a\,q\,b\,v$ hat als Nachfolgerkonfiguration

$u\,q'\,a\,c\,v$ gdw. $\delta(q, b) = (q', c, L)$

$u\,a\,c\,q'\,v$ " $\delta(q, b) = (q', c, R)$ ←



$q'cv$ gdw. $\delta(q, b) = (q', c, L)$ und q auf dem ersten Feld steht

$cq'v$ $\delta(q, b) = (q', c, R)$ "

Jede Turingtafel gibt angesetzt auf ein Eingabewort $w \in \Sigma^*$

eine Konfigurationsfolge $C_0 = q_0 w$ $Q \cap \Gamma = \emptyset$

$C_{i+1} =$ Nachf. konf von C_i

$\vdots$

$\vdots$

Die Folge bricht ab, wenn $q(C_n)$ ein Stoppzustand ist.

<u>Def</u>: Eine TM M akzeptiert einen Input w, gdw die Berechnungsfolge

in einem akzeptierenden Zustand endet.

$$A(M) = \{ \sigma \in \Sigma^* \mid M \text{ akzeptiert } \sigma \}$$

Def: $A \subseteq \Sigma^*$ heißt Turing-berechenbar gdw es eine (immer stoppende) TM M gibt, s.d. $A = A(M)$.

Mächtigkeitsargument für die Existenz unentscheidbarer Mengen:

$|\mathcal{P}(\Sigma^*)| = |\mathcal{P}(\mathbb{N})|$ überabzählbar

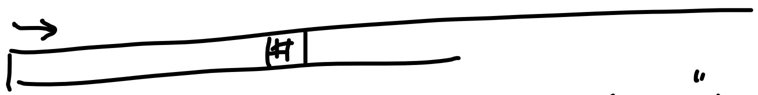
Es gibt nur abz. viele Turingmaschinen

Bsp

1) Ein M mit $A(M) = \{ \hat{w} \# \hat{w} \mid \hat{w} \in \{0,1\}^* \}$

$w_0 (w_1)^k \mid k \in \mathbb{N}$

Skizze:

1. 

M liest w, das Eingabewort, und prüft, ob es genau ein # hat.

2. 

M geht (Anzahl der Buchstaben vor #) oft hin und her mit

2. Lesegedächtnis und vergleicht jeweils den k. Buchstaben von links an mit dem k. Buchstaben nach #.
Wenn die Prüfung positiv ausfällt, dann überschreibt M die geprüften Buchstaben mit x und wiederholt diesen Rechenschritt. Im negativen Fall lehnt M den Input ab.

3. Wenn alle Buchstaben links von # durch x überschrieben sind oder wenn alle Buchstaben rechts des # durch x überschrieben sind, dann müssen auch auf der anderen Seite keine Buchstaben mehr sein, sonst lehnt M den Input ab. Im anderen Fall akzeptiert M den Input.

Beispiel: Ein M mit $A(M) = \{ 0^{2^n} \mid n \in \mathbb{N} \}$ $\quad 2^0 = 1$

$\Sigma = \{0\}$, $\quad \Gamma = \{0, x, \varepsilon\}$

1. Prüfe ob ~~das Wort~~ die Bandinschrift die Form $0x^k$ hat. $\quad x^k = x^{2^n - 1}$

   Wenn ja akzeptiere. Wenn nein, gehe weiter

2.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 x 0 x 0 x 0 x 0 x 0 x 0 x 0 x 0 x 0 x Ende der Schleife

0 x x x 0 x x x 0 x x x 0 x x x 0 x x x ?

$2^n - 1$ viele x Muster.

0 x 0 x 0

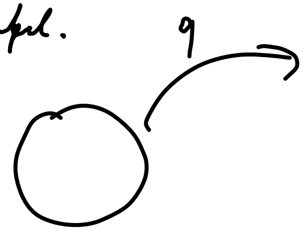
0 x 0 … ?

0 $2^n - 1$ viele x

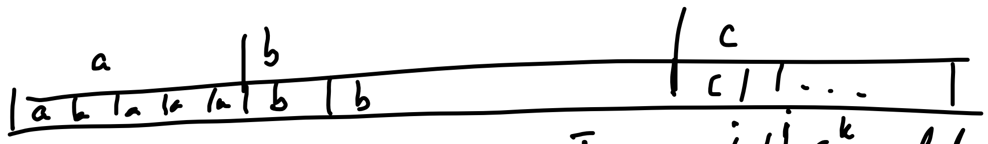
2. Gehe von links nach rechts über das Wort und schreibe ein x statt jeder zweiten Null, die anderen Nullen bleiben stehen.
Wenn die Zeile bei einer unpaarigen Null (und einigen x'en danach) endet, dann lehne ab. Sonst gehe zu Schritt 1.

Geduldsaufgabe: Schreiben Sie eine Turingtafel.

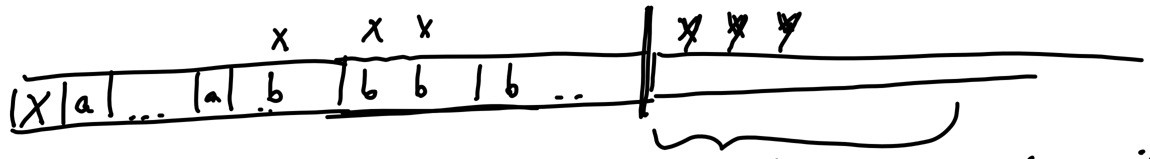
oder ein Flussdiagramm.



Bsp: $A(M) = \{ a^i b^j c^k \mid i \cdot j = k \text{ und } i,j,k > 0 \}$, $\Sigma = \{a,b,c\}$
$\Gamma = \{a,b,c,x,y,\varepsilon\}$

1. 

Prüfe dies. Wenn nicht der Form $a^i b^j c^k$, lehne ab.

2. 

b viele c's durch x überschreiben

M löscht ein a, und der Lesekopf geht nach rechts, bis b auftritt

Dann löscht M abwechselnd ein b und ein c, bis kein b mehr da ist.
Falls dies nicht aufgeht, lehne ab.

3. M schreibt die b wieder hin.

Gehe zu 2.

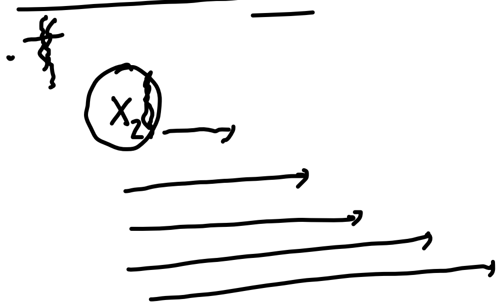
4. Falls kein a mehr da ist, prüfe ob alle c danach y's ersetzt sind

Wenn ja akzeptiere, wenn nein lehne ab.

Beispiel 4:

$$A(M) = \{ x_1 \# x_2 \# \dots \# x_\ell \mid \ell \in \mathbb{N},\ x_i \in \{0,1\}^*,\ x_i \neq x_j \text{ für } i \neq j \}$$

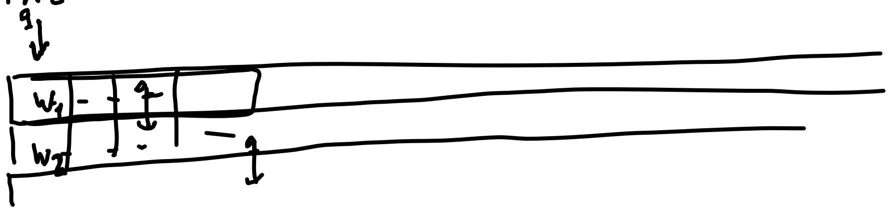
Skizze



Schleifen mit 2 „Marken". neue Hilfsbuchstaben.

Grob gesprochen: Turingmaschine, die nicht schreiben können, haben die Berechnungsstärke von endl. Automaten.

„Kleine" Variation von Turingmaschinen: Mehrbandturingmaschine



Def: Tafel einer Mehrbandturingmaschine

$$\delta: Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$$

$$\delta(q, \gamma_1, \ldots, \gamma_k) = (q', \gamma_1', \ldots, \gamma_k', \underbrace{L, R}_{k\text{-Tupel}})$$

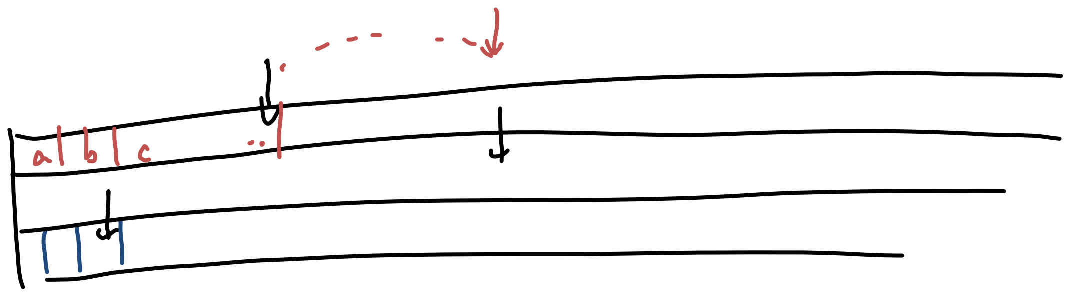
Satz Zu jeder Mehrband-Turingmaschine gibt es eine Ein-Band-TM, die dieselbe Akzeptierungsmenge hat.

Beweisskizze:

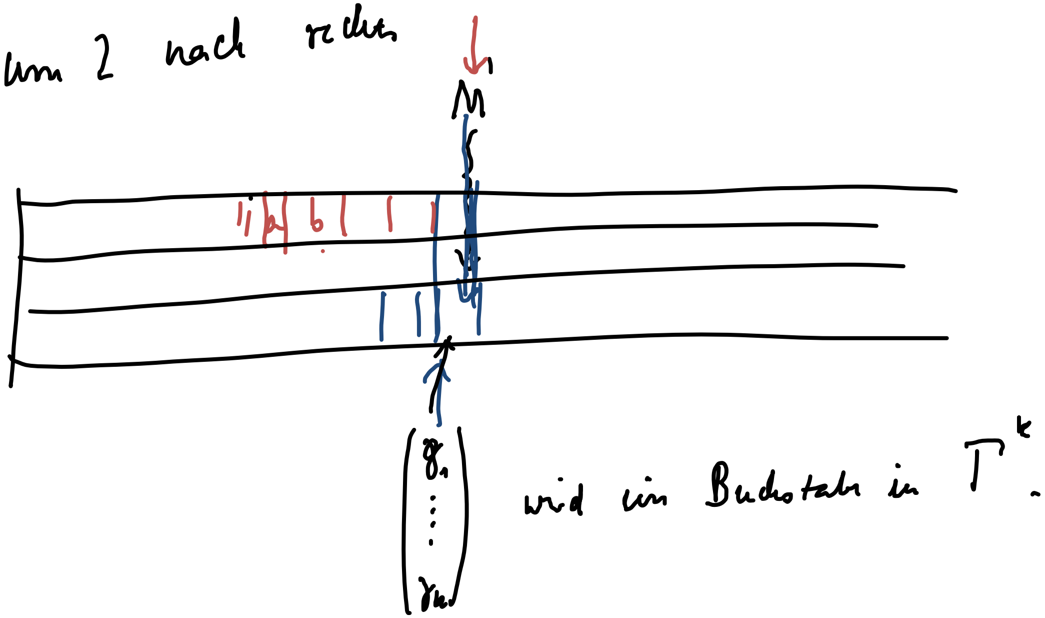
Alphabet der Einbandturingmaschine Γ^k. Gegeben M, eine ~~Mehr~~ k-Band TM über Γ.

Der Lesekopf der Einband-TM M' geht nach links genau wenn bei M alle Leseköpfe nach links gehen.

Im anderen Fall geht der Lesekopf von M' nach rechts.



Simulation: Verschiebe alle Bänder, auf denen M nach links gegangen war, um 2 nach rechts



wird ein Buchstabe in Γ^k.

Unterroutinen: Verschiebe Band Nummer i für ein $i \in \{1, \ldots, k\}$ um 2 Felder nach rechts. Sipser.

Größere Variation: Nichtdeterministische Turingmaschinen