

# Weitere universelle Berechnungsmodelle



- Mehrband Turingmaschine
- Nichtdeterministische Turingmaschine
- RAM-Modell
- Vektoradditionssysteme
- $\lambda$ -Kalkül
- $\mu$ -rekursive Funktionen

# Varianten der dtm



- Mehrkopf dtm



- Mehrspur dtm



- dtm mit mehrdimensionalem Band

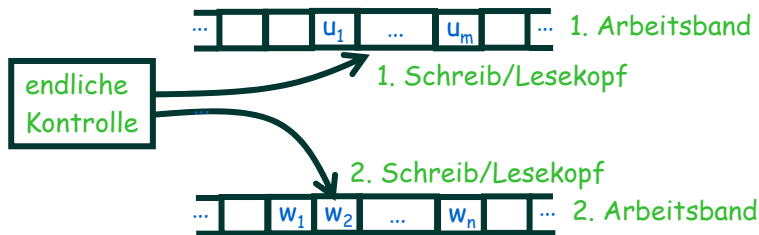
→ können alle durch 1-Band dtm simuliert werden!

# Deterministische $k$ -Band Turing-Maschine

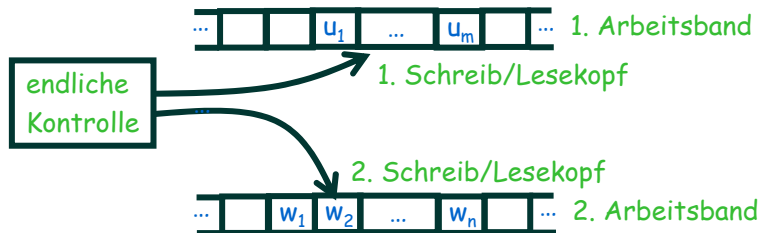


## Deterministische $k$ -Band Turingmaschine

- hat endlich viele innere Zustände
- hat  $k$  beidseitig unendliche Arbeitsbänder  
(1 Zeichen pro Zelle, zu jedem Zeitpunkt nur endlich viele Zellen beschriftet)
- hat für jedes Band einen beweglichen Schreib/Lesekopf



## Arbeitsweise der $k$ -Band Turing-Maschine



von den Arbeitsbändern gelesene Symbole

(an den aktuellen Kopffpositionen) +

aktueller innerer Zustand

→ schreibe neue Symbole an aktuelle Positionen +

bewege Schreib-/Leseköpfe um  $\leq 1$  Feld +

gehe in neuen inneren Zustand

# Deterministische $k$ -Band Turing-Maschine



Def.: Eine **deterministische  $k$ -Band Turing-Maschine  $M$  ( $k$ -dtm)**

ist ein 7-Tupel  $(Q, \Sigma, \Gamma, \delta, q_0, \underline{b}, F)$  mit

- $Q$  endliche Menge von **inneren Zuständen**
- $\Sigma$  endliches **Eingabealphabet**
- $\Gamma$  endliches **Bandalphabet**  $\Sigma \subset \Gamma$
- $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$   
**Zustandsüberföhrungsfunktion**  
*(L=links, R=rechts, S=stehenbleiben)*
- $q_0 \in Q$  **Anfangszustand**
- $\underline{b} \in \Gamma \setminus \Sigma$  **Blanksymbol**
- $F = F_{ja} \cup F_{nein} \subset Q$  Menge der **Endzustände** mit
  - $F_{ja}$  **akzeptierenden Endzustände**
  - $F_{nein}$  **nicht akzeptierenden Endzustände**

## Simulation von $k$ -dtm durch dtm



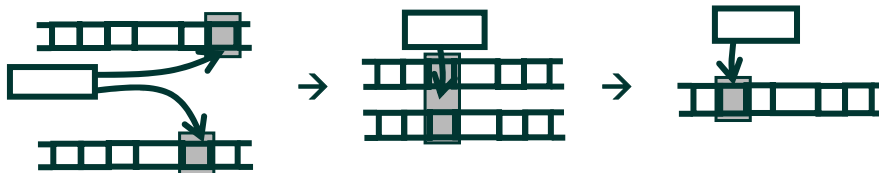
Satz:  $k$ -dtm  $M$  kann von einer 1-dtm  $S$  mit

$$t_s(n) = O((t_M(n))^2) \text{ und } s_s(n) = O(s_M(n))$$

simuliert werden.

Bew. (Idee):

- $k$ -Band dtm  $M$  auf  $k'$ -Spur dtm  $M'$  simulieren
- $k'$ -Spur dtm  $M'$  auf 1-Band dtm  $S$  simulieren



## Simulation $k$ -Band dtm durch $k'$ -Spur dtm



**Satz:**  $k$ -dtm  $M$  kann von einer  $(2k+1)$ -Spur dtm  $S$  mit  
 $t_S(n) = O((t_M(n))^2)$  und  $s_S(n) = O(s_M(n))$   
simuliert werden.



## Simulation $k$ -Band dtm durch $k'$ -Spur dtm



**Bew. (Skizze) ...:**

Invariante: vor der Simulation des Rechenschrittes  
 $t \leq t_M(n)$  von  $M$  gilt:

1. Spuren  $1, 3, \dots, 2k-1$  enthalten **Inhalt** der  $k$  Bänder von  $M$
2. Spuren  $2, 4, \dots, 2k$  enthalten genau einmal das Sonder-  
symbol  $\#$  als Markierung der **Kopfpositionen** von  $M$
3. Spur  $2k+1$  enthält zwei Sondersymbole  $L$  und  $R$  die die  
linke bzw. rechte **Begrenzung** der Bandinhalte von  $M$   
markieren



## Simulation $k$ -Band dtm durch $k'$ -Spur dtm



... **Bew.** (Skizze) ...:

*Simulation:*

1. Initialisieren
2. Zu Beginn des  $t$ -Schrittes kennt  $S$  den Zustand von  $M$  und steht in Spur  $2k+1$  auf  $L$  (linker Rand)
3.  $S$  geht nach rechts bis  $R$  und "merkt" sich im endlichen Speicher was die Köpfe von  $M$  lesen würden ( $\rightarrow S$  kennt Informationen die  $M$  hat & kann dessen Aktionen nachvollziehen  $\rightarrow S$  kennt neuen Zustand von  $M$ )
4.  $S$  läuft nach links, simuliert dabei  $M$  und verschiebt ggf. in Spur  $2k+1$  die  $L$  und  $R$  Markierungen  
 $\rightarrow$  Invariante bleibt erhalten

## Simulation $k$ -Band dtm durch $k'$ -Spur dtm



... **Bew.** (Skizze):

Abstand zwischen  $L$  und  $R$  in Spur  $2k+1$  ist höchstens  $t_M(n)$

$\rightarrow$  jeder Schritt von  $M$  kann von  $M'$   
in  $O(t_M(n))$  Zeit simuliert werden

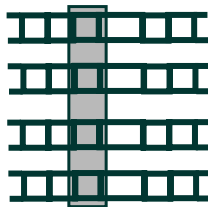
$\rightarrow M$  kann von  $M'$  in  $O(t_M^2(n))$  Zeit simuliert werden

## Simulation $k$ -Spur dtm durch 1-Band dtm



**Satz:**  $k$ -Spur dtm  $M$  kann von einer 1-dtm  $S$  mit  
 $t_S(n) = O(t_M(n))$  und  $s_S(n) = O(s_M(n))$   
simuliert werden.

**Bew. (Idee):**



durch  $\Gamma' := \Sigma \cup \Gamma^k$   
 $\underline{b}' := \underline{b}^k$  simulieren

## Abschlueigenschaften rekursiver Sprachen



**Satz:** Sind  $L_1, L_2$  entscheidbar, dann sind auch  $L_1 \cup L_2$  und  
 $L_1 \cap L_2$  entscheidbar

**Bew. :**

$M_1, M_2$  seien dtm die  $L_1$  bzw.  $L_2$  entscheiden

→ konstruiere 2-dtm  $M$  die  $M_1, M_2$  simultan simuliert:

- $i$ -tes Band von  $M$  entspricht dem Arbeitsband von  $M_i$
- Zustandsmenge von  $M = Q_1 \times Q_2$
- Eingabe wird zunachst auf 2-tes Band kopiert
- akzeptiere, genau dann wenn
  - **mindestens eine** der Maschinen akzeptiert  
→  $L_1 \cup L_2$
  - **beide** Maschinen akzeptieren →  $L_1 \cap L_2$

## Weitere Abschlußeigenschaften



### Satz:

- Sind  $L_1, L_2$  semi-entscheidbar, dann sind auch  $L_1 \cup L_2$  und  $L_1 \cap L_2$  semi-entscheidbar
- Sind  $L, L^c$  semi-entscheidbar, dann ist  $L$  entscheidbar

Bew.: gleiche Technik ( $\rightarrow$  Übung)

Bem.: Semi-entscheidbare Sprachen sind nicht unter Komplementbildung abgeschlossen !! ( $\rightarrow$  später)

## Nichtdeterministische Turing-Maschine



Def.: Eine nichtdeterministische Turing-Maschine  $M$  (*ntm*) ist ein 7-Tupel  $(Q, \Sigma, \Gamma, \delta, q_0, \underline{b}, F)$  mit

- $Q$  endliche Menge von inneren Zuständen
- $\Sigma$  endliches Eingabealphabet
- $\Gamma$  endliches Bandalphabet  $\Sigma \subset \Gamma$
- $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R, S\}}$   
Zustandsüberföhrungsfunktion  
( $L$ =links,  $R$ =rechts,  $S$ =stehenbleiben)
- $q_0 \in Q$  Anfangszustand
- $\underline{b} \in \Gamma \setminus \Sigma$  Blanksymbol
- $F = F_{ja} \cup F_{nein} \subset Q$  Menge der Endzustände mit
  - $F_{ja}$  akzeptierenden Endzustände
  - $F_{nein}$  nicht akzeptierenden Endzustände

## Berechnungspfad einer ntm $M$

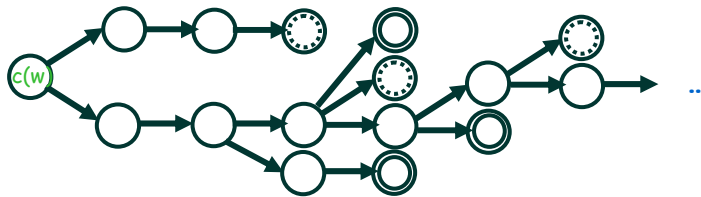


Def.: Ein **Berechnungspfad** von  $M$  auf  $w \in \Sigma^*$  ist die **Folge der Konfigurationen**  $B_M(w) = c_0, c_1, c_2, \dots$  die  $M$  bei einer möglichen Abarbeitung von  $w$  durchläuft:

- $c_0 = c(w)$
- $c_{i+1} \in N_M(c_i)$  falls  $c_i = uqv$  mit  $q \in F^c$

"Überlagerung" aller Berechnungspfade von  $M$  auf  $w \rightarrow$

**Berechnungsbaum**  $T_M(w)$  von  $M$  auf  $w$

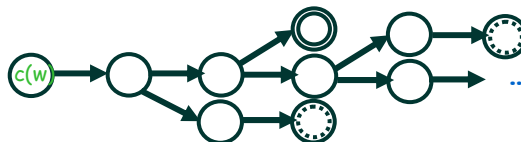


## Akzeptanzkriterium einer ntm $M$

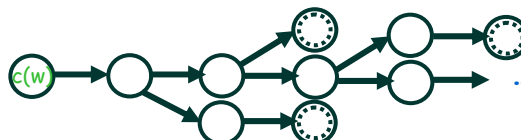


Def.:

- $M$  **akzeptiert**  $w \in \Sigma^*$ , falls es einen Berechnungspfad von  $M$  auf  $w$  **gibt**, bei dem  $M$  in einer Konfiguration  $uqv$  mit  $q \in F_{\text{ja}}$  **hält**



- $M$  **verwirft**  $w \in \Sigma^*$ , falls **alle** endlichen Berechnungspfade von  $M$  auf  $w$  in einer Konfiguration  $uqv$  mit  $q \in F_{\text{nein}}$  **enden**





## Rechenzeit & Platzbedarf einer ntm $M$



Def.: **Rechenzeit** von  $M$  auf  $w \in \Sigma^*$

$t_M(w)$

:= Länge eines **kürzesten** akzeptierenden Berechnungspfad  $B_M(w)$ , falls  $w$  von  $M$  **akzeptiert** wird

:= Länge eines **längsten** verwerfenden Berechnungspfad  $B_M(w)$ , falls  $M$  auf  $w$  **immer hält**, aber  $w$  **nicht akzeptiert**

:=  $\infty$ , sonst

$t_M(n) := \max_{w \in \Sigma^n} t_M(w)$  heisst **Rechenzeit** von  $M$

"worst-case"

## Simulation von ntm durch dtm



Satz: Eine ntm  $N$  kann von einer 3-dtm  $S$  mit

$$t_S(n) = O(t_N(n)2^{O(t_N(n))}) \quad \text{und} \quad s_S(n) = O(t_N(n))$$

simuliert werden.

Bew. (Idee):

1. Band: Eingabe
2. Band: Simulation eines Berechnungspfad
3. Band: Systematisches Generieren aller

Berechnungspfade

es gibt  $O(2^{O(t_N(n))})$  **verschiedene** Konfigurationen von  $N$  auf Eingaben der Länge  $n$