

# Modelle der Parallelverarbeitung

## 2. Parallele Turingmaschinen

Thomas Worsch

Institut für Theoretische Informatik  
Karlsruher Institut für Technologie

Sommersemester 2018

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steereinheiten

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen

Erkennung von  $L_{vv}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steereinheiten

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen zwischen TM und PTM

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steereinheiten

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen

Erkennung von  $L_{vv}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steereinheiten

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steuereinheiten

### Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen

Erkennung von  $L_{vv}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steuereinheiten

# Parallele Turingmaschinen: Idee

der einfache Fall: „ein Band, ein Kopf“:

- ▶ *ein* gemeinsames Arbeitsband, auf dem
- ▶ *mehrere* Steereinheiten
- ▶ *parallel*, d. h. gleichzeitig (zusammen)arbeiten

# Parallele Turingmaschinen: Ursprünge



A. Hemmerling

Systeme von Turing-Automaten und Zellularräume auf rahmbaren Pseudomustermengen.

*Elektronische Informationsverarbeitung und Kybernetik* **15**, S. 47–72, 1979.



J. Wiedermann

Paralelný Turingov stroj — Model distribuovaného počítača.

In J. Gruska (ed.): *Distribované a paralelné systémy*, 205–214, 1983.



J. Wiedermann

Parallel Turing Machines.

Technical Report RUU-CS-84-11, Univ. of Utrecht, 1984.

## Definition von $W_1$ -PTM

Bestandteile weitgehend wie bei TM:

- ▶ **Bandalphabet:**  $B$
- ▶ **Zustandsmenge:**  $S$
- ▶ **Konfiguration:**  $c : \mathbb{Z} \rightarrow 2^S \times B$        $2^S$ : Potenzmenge von  $S$   
 $c(i) = (T, b)$ :
  - ▶  $T$ : Menge der Zustände der Steereinheiten,
    - ▶ die gerade Feld  $i$  besuchen,
    - ▶ also *keine Unterscheidung mehrerer Steereinheiten im gleichen Zustand auf dem gleichen Feld!*
  - ▶  $b$ : Bandsymbol auf Feld  $i$
- ▶  $C = (2^S \times B)^{\mathbb{Z}}$       Menge aller Konfigurationen

## Lokale Arbeitsweise von $\mathbb{W}_1$ -PTM

- ▶ potenzielles Problem:



## Lokale Arbeitsweise von $\mathbb{W}_1$ -PTM

- ▶ potenzielles Problem: *Vermeidung von Schreibkonflikten*

## Lokale Arbeitsweise von $\mathbb{W}_1$ -PTM

- ▶ potenzielles Problem: *Vermeidung von Schreibkonflikten*  
Lösung: alle lokalen Änderungen werden „kollektiv vereinbart“

## Lokale Arbeitsweise von $\mathbb{W}_1$ -PTM

- ▶ potenzielles Problem: *Vermeidung von Schreibkonflikten*  
Lösung: alle lokalen Änderungen werden „kollektiv vereinbart“
- ▶ *lokale Überföhrungsfunktion*  $\delta : 2^S \times B \rightarrow 2^{S \times D} \times B$   
bzw.
  - ▶  $\delta_s : 2^S \times B \rightarrow 2^{S \times D}$
  - ▶  $\delta_b : 2^S \times B \rightarrow B$
- ▶ *Einschränkungen:*
  - ▶ Automaten entstehen nicht aus dem Nichts:  
 $\forall b \in B : \delta_s(\emptyset, b) = \emptyset$
  - ▶ Felder, die nicht besucht werden, bleiben unverändert:  
 $\forall b \in B : \delta_b(\emptyset, b) = b$

## Globale Arbeitsweise von $\mathbb{W}_1$ -PTM

- ▶ *ein globaler Schritt*  $\Delta : C \rightarrow C$ ,  
wobei  $\Delta(c)(i) = (T', b')$  mit
  - ▶  $T' = \{s \mid \exists d : (s, d) \in \delta_s(c(i-d))\}$
  - ▶  $b' = \delta_b(c(i))$
- ▶ Beachte:
  - ▶ Die Anzahl der Steuereinheiten kann schrumpfen,
    - ▶ weil  $\delta_s(T, b)$  kleiner ist als  $T$ , und
    - ▶ weil Steuereinheiten im gleichen Zustand beim Fahren auf das gleiche Nachbarfeld aus verschiedenen Richtungen „verschmelzen“.
  - ▶ Die Anzahl der Steuereinheiten kann wachsen,
    - ▶ weil  $\delta_s(T, b)$  größer ist als  $T$ .

## Alternative Definition der lokalen Arbeitsweise

- ▶ Spezifikation für jede Steuereinheit separat
- ▶ wie bei  $*W_*$ -TM Benutzung von  $B_- = B \cup \{-\}$  für „Schreiben“ oder eben „Nichts schreiben“
- ▶ lokale Überföhrungsfunktion  
 $\delta : S \times B \rightarrow 2^{S \times D} \times B_-$  oder gar  
 $\delta : S \times B \rightarrow 2^{S \times D \times B_-}$
- ▶ gleichzeitiges Schreiben verschiedener Symbole auf das gleiche Feld verboten
- ▶ sonstige plausible Einschränkungen wie eben
- ▶ *Behauptung:*  
Das liefert im Wesentlichen das gleiche Modell.

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steuereinheiten

Grundlagen

**Erkennung formaler Sprachen**

Komplexitätsmaße

Beziehungen

Erkennung von  $L_{vv}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steuereinheiten

## Eingabe von Wörtern

- ▶ Anfangszustand  $s_0 \in S$
- ▶ Eingabealphabet  $A \subset B \setminus \{\square\}$
- ▶ Blanksymbol  $\square$
- ▶ *Anfangskonfiguration* zu  $w$ :  
init :  $A^+ \rightarrow C : w_1 \cdots w_n \mapsto c$  mit

$$c(i) = \begin{cases} (\{s_0\}, w_1) & \text{falls } 1 = i \\ (\emptyset, w_i) & \text{falls } 2 \leq i \leq n \\ (\emptyset, \square) & \text{sonst} \end{cases}$$

# Halten und Akzeptieren

## Definition

- ▶ Menge von Endzuständen  $F \subset S$
- ▶  $\text{isFinal} : C \rightarrow \mathbb{B} : c \mapsto [c(1) \in (2^F \setminus \emptyset) \times B]$
- ▶ Menge akzeptierender Endzustände  $F_+ \subset F$
- ▶  $\text{isAccepting} : C \rightarrow \mathbb{B} : c \mapsto [c(1) \in (2^{F_+} \setminus \emptyset) \times B]$

## Beachte

- ▶ andere Definitionen von Endkonfiguration
  - ▶ z. B.:  $\text{isFinal} : C \rightarrow \mathbb{B} : c \mapsto [\forall i \in \mathbb{Z} : c(i) \in \{\emptyset\} \times B]$
- ▶ für unsere Belange äquivalent  
(bis auf konstanten Faktor beim Zeitbedarf)



## Von PTM erkannte Sprache

- ▶  $L(T) = \{w \in A^+ \mid \text{isAccepting}(\Delta^{\text{time}(w)}(\text{init}(w)))\}$
- ▶ Definition von  $\text{time}(w)$ , etc. kommt gleich

## Beispiel einer PTM

- ▶ Aufgabe: erkenne  $L_{121} = \{1^k 2 1^k \mid k \in \mathbb{N}_+\}$

## Beispiel einer PTM

- ▶ Aufgabe: erkenne  $L_{121} = \{1^k 2 1^k \mid k \in \mathbb{N}_+\}$
- ▶ drei Phasen:
  - ▶ suche die erste 2
  - ▶ vergleiche Abstand vom linken und rechten Rand  
prüfe ob ansonsten nur 1
  - ▶ Ergebnis ans linke Ende transportieren

## Beispiel-PTM für $L_{121}$ (1)

	m						
	1	1	2	1	1	1	
		m					
	1	1	2	1	1	1	
			m				
	1	1	2	1	1	1	

## Beispiel-PTM für $L_{121}$ (2)

		1	1	m 2	1	1	1		
		1	<l	2	r>	1	1	1	
		<l	1	2	1	r>	1	1	
	<l	1	1	2	1	1	r>	1	
		l>	1	2	1	1	1	r>	
		1	l>	2	1	1	<r	1	
		1	1	l>	2	<r	1	1	
		1	1	2	l>, <r	1	1	1	

## Beispiel-PTM für $L_{121}$ (3)

	1	1	2	l>,<r 1	1	1	
	1	1	f- 2	1	1	1	
	1	f- 1	2	1	1	1	
	f- 1	1	2	1	1	1	
f-	1	1	2	1	1	1	
	F- 1	1	2	1	1	1	

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steuerheiten

Grundlagen

Erkennung formaler Sprachen

**Komplexitätsmaße**

Beziehungen

Erkennung von  $L_{\forall\forall}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steuerheiten

## Zeitkomplexität von PTM

- ▶ Annahme wie immer: Jede Maschine hält für jede Eingabe.
- ▶ Dann sind die folgenden Funktionen total:

$$\text{time} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \min\{t \in \mathbb{N}_+ \mid \text{isFinal}(\Delta^t(\text{init}(w)))\}$$

$$\text{Time} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max\{\text{time}(w) \mid w \in A^n\}$$

- ▶ Das ist die gleiche Definition wie bei TM.



## Raumkomplexität von $\mathbb{W}_1$ -PTM

- ▶ Ziel: analog zu TM
- ▶ Was soll es heißen „ein Feld zu benutzen“?
- ▶ definiere:

$$\text{used-mem} : C \rightarrow 2^{\mathbb{Z}}$$

$$c \mapsto \{i \in \mathbb{Z} \mid c(i) \neq (\emptyset, \square)\}$$

$$\text{space} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \left| \bigcup_{t \leq \text{time}(w)} \text{used-mem}(\Delta^t(\text{init}(w))) \right|$$

$$\text{Space} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max\{\text{space}(w) \mid w \in A^n\}$$

## Prozessorkomplexität von $\mathbb{W}_1$ -PTM

- ▶ die Anzahl maximal gleichzeitig vorhandener Steuereinheiten
- ▶ definiere:

$$\text{used-proc} : C \rightarrow 2^{\mathbb{Z}}$$

$$c \mapsto \{i \in \mathbb{Z} \mid c(i) \in (2^S \setminus \emptyset) \times B\}$$

$$\text{proc} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \max \left\{ \left| \text{used-proc}(\Delta^t(\text{init}(w))) \right| \mid t \leq \text{time}(w) \right\}$$

$$\text{Proc} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max \{ \text{proc}(w) \mid w \in A^n \}$$

- ▶ Komplexitätsklassen:

$$\mathbb{W}_1\text{-PTM-PROC}(p(n)) - \dots$$

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steuereinheiten

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

**Beziehungen**

Erkennung von  $L_{\forall\forall}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steuereinheiten

## Zusammenhang von PTM mit TM (1)

- ▶ Es sollte klar sein, dass eine sequentielle TM im wesentlichen eine PTM mit Prozessorkomplexität 1 ist.
- ▶ Für alle vernünftigen (Übung: präzisieren) Komplexitätsschranken gilt:

$$\begin{aligned} \mathbb{W}_1\text{-TM-SPC}(\Theta(s(n))\text{-TIME}(\Theta(t(n)))) \\ = \mathbb{W}_1\text{-PTM-SPC}(\Theta(s(n))\text{-TIME}(\Theta(t(n))\text{-PROC}(1)) \end{aligned}$$

# Überblick

## Parallele Turingmaschinen mit Ein-Kopf-Steereinheiten

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen

Erkennung von  $L_{vv}$  mit  $\mathbb{W}_1$ -PTM

## Parallele Turingmaschinen mit Mehr-Kopf-Steereinheiten

## $L_{UV}$ : verschiedene Möglichkeiten der Erkennung

## $L_{VV}$ : verschiedene Möglichkeiten der Erkennung

- ▶ wie eine sequentielle TM

## $L_{UV}$ : verschiedene Möglichkeiten der Erkennung

- ▶ wie eine sequentielle TM
- ▶ schneller: durch Einsatz mehrerer Steereinheiten



## $L_{UV}$ : verschiedene Möglichkeiten der Erkennung

- ▶ wie eine sequentielle TM
- ▶ schneller: durch Einsatz mehrerer Steuereinheiten
- ▶ z. B. so: drei Teilaufgaben für Eingabe  $w$ :
  - ▶ prüfe, ob  $w \in \{0, 1\}^+ 2^+ \{0, 1\}^+$
  - ▶ prüfe, ob Längen der beiden ersten Blöcke gleich sind
  - ▶ prüfe, ob erster und dritter Block identisch sind
  - ▶ (Details an der Tafel)

# Ergebnisse

## Satz

$L_{vv} \in \mathbb{W}_1\text{-PTM-SPC}(\Theta(n))\text{-TIME}(\Theta(n))\text{-PROC}(\Theta(n))$

# Ergebnisse

## Satz

$$L_{vv} \in \mathbb{W}_1\text{-PTM-SPC}(\Theta(n))\text{-TIME}(\Theta(n))\text{-PROC}(\Theta(n))$$

## Verallgemeinerung

Für jede rationale Zahl  $a$  mit  $0 < a \leq 1$  gilt:

$$L_{vv} \in \mathbb{W}_1\text{-PTM-SPC}(\Theta(n))\text{-TIME}(\Theta(n^{2-a}))\text{-PROC}(\Theta(n^a))$$

- ▶ und untere Schranken ... ? ...

## Eine untere Schranke für $L_{VV}$

### Satz

Für jede PTM, die  $L_{VV}$  erkennt, gilt:

$$\text{Time}^2(n) \cdot \text{Proc}(n) \in \Omega(n^3 / (\log n)^2)$$

## Eine untere Schranke für $L_{VV}$

### Satz

Für jede PTM, die  $L_{VV}$  erkennt, gilt:

$$\text{Time}^2(n) \cdot \text{Proc}(n) \in \Omega(n^3 / (\log n)^2)$$

- ▶ beste obere Schranke aus erwähnter Verallgemeinerung:  
 $\text{Time}^2(n) \cdot \text{Proc}(n) \in \Theta(n^{4-a})$
- ▶ *Wer kann die Lücke verkleinern?*

# Überblick

Parallele Turingmaschinen mit Ein-Kopf-Steuerheiten

Parallele Turingmaschinen mit Mehr-Kopf-Steuerheiten

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen zwischen TM und PTM

# Überblick

Parallele Turingmaschinen mit Ein-Kopf-Steereinheiten

**Parallele Turingmaschinen mit Mehr-Kopf-Steereinheiten**

**Grundlagen**

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen zwischen TM und PTM

# Geschichte



J. Wiedermann

Parallel Turing Machines.

Technical Report RUU-CS-84-11, Univ. of Utrecht, 1984.



J. Wiedermann

Weak Parallel Machines: A new class of physically feasible parallel machine models.

Proc. 17<sup>th</sup> MFCS 1992, LNCS 629, pp. 95–111, 1992.

Verallgemeinerung von  $\mathbb{W}_1$ -PTM:

- ▶ mehrere Köpfe auf einem Band und/oder
- ▶ mehrere Bänder, auf denen jede Steuereinheit jeweils mindestens einen Kopf hat



## Definition von $\mathbb{W}_{h_1} \mathbb{W}_{h_2} \cdots \mathbb{W}_{h_k}$ -PTM

- ▶ *lokale Überföhrungsfunktion* analog zu  $\mathbb{W}_1$ -PTM, aber:  
$$\delta : S \times B^{h_1} \times \cdots \times B^{h_k} \rightarrow 2^{S \times (B \times D)^{h_1} \times \cdots \times (B \times D)^{h_k}}$$
- ▶ *globale Konfiguration*  $c = (b_1, \dots, b_k, u)$ 
  - ▶  $b_1, \dots, b_k$ : Bandbeschriftungen
  - ▶  $u = (u_1, \dots, u_m)$ , wobei jedes  $u_j \in S \times \mathbb{Z}^{h_1} \times \cdots \times \mathbb{Z}^{h_k}$  für eine Steereinheit Zustand und Positionen der Köpfe spezifiziert.
- ▶  $\delta$  induziert  $\Delta$  auf die „naheliegende“ Weise
- ▶ Schreibkonflikte verboten

# Überblick

Parallele Turingmaschinen mit Ein-Kopf-Steereinheiten

**Parallele Turingmaschinen mit Mehr-Kopf-Steereinheiten**

Grundlagen

**Erkennung formaler Sprachen**

Komplexitätsmaße

Beziehungen zwischen TM und PTM

# Anfangskonfigurationen

- ▶ Eingabewort auf erstem Band
- ▶ nur eine Steuereinheit vorhanden
- ▶ alle Köpfe auf Feld 1 des jeweiligen Bandes

# Halten und Akzeptieren

Diverse verschiedene Festlegungen sind wieder äquivalent, z. B.:

- ▶  $\text{isFinal}(c) \iff$  es gibt noch genau eine Steuereinheit und ihr Zustand ist in  $F$
- ▶  $\text{isAccepting}(c) \iff \text{isFinal}(c)$  und Zustand der Steuereinheit ist in  $F_+$

# Überblick

Parallele Turingmaschinen mit Ein-Kopf-Steereinheiten

**Parallele Turingmaschinen mit Mehr-Kopf-Steereinheiten**

Grundlagen

Erkennung formaler Sprachen

**Komplexitätsmaße**

Beziehungen zwischen TM und PTM

## Zeitkomplexität von $*W_*$ -PTM

wie bei  $W_1$ -PTM etc.

## Prozessorkomplexität von $*W_*$ -PTM

inhaltlich wie bei  $W_1$ -PTM:

die Anzahl maximal gleichzeitig vorhandener Steereinheiten

## Raumkomplexität von $*W_*$ -PTM:

- ▶ naheliegend:

$$\text{used-mem} : C \rightarrow 2^{\mathbb{N}_+ \times \mathbb{Z}}$$

$$c \mapsto \{(j, i) \in \mathbb{Z} \mid \text{in } c \text{ wird auf Band } j \text{ Feld } i \text{ benutzt}\}$$

$$\text{space} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \left| \bigcup_{t \leq \text{time}(w)} \text{used-mem}(\Delta^t(\text{init}(w))) \right|$$

$$\text{Space} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max\{\text{space}(w) \mid w \in A^n\}$$



## Raumkomplexität von $*W_*$ -PTM: 1. Versuch

- ▶ naheliegend:

$$\text{used-mem} : C \rightarrow 2^{\mathbb{N}_+ \times \mathbb{Z}}$$

$$c \mapsto \{(j, i) \in \mathbb{Z} \mid \text{in } c \text{ wird auf Band } j \text{ Feld } i \text{ benutzt}\}$$

$$\text{space} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \left| \bigcup_{t \leq \text{time}(w)} \text{used-mem}(\Delta^t(\text{init}(w))) \right|$$

$$\text{Space} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max\{\text{space}(w) \mid w \in A^n\}$$

- ▶ Was ist daran gut oder schlecht ... ? ...

## Raumkomplexität von $*W_*$ -PTM: 1. Versuch

- ▶ naheliegend:

$$\text{used-mem} : C \rightarrow 2^{\mathbb{N}_+ \times \mathbb{Z}}$$

$$c \mapsto \{(j, i) \in \mathbb{Z} \mid \text{in } c \text{ wird auf Band } j \text{ Feld } i \text{ benutzt}\}$$

$$\text{space} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \left| \bigcup_{t \leq \text{time}(w)} \text{used-mem}(\Delta^t(\text{init}(w))) \right|$$

$$\text{Space} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max\{\text{space}(w) \mid w \in A^n\}$$

- ▶ Was ist daran gut oder schlecht ... ? ...
- ▶ Hinweis: *Bandkomplexität* ist eine treffendere Bezeichnung als *Speicherplatzkomplexität* ...

## Bandkomplexität für $*W_*$ -PTM

- ▶ also:

$$\text{used-tape} : C \rightarrow 2^{\mathbb{N}_+ \times \mathbb{Z}}$$

$$c \mapsto \{(j, i) \in \mathbb{Z} \mid \text{in } c \text{ Feld } i \text{ auf Band } j \text{ benutzt}\}$$

$$\text{tape} : A^+ \rightarrow \mathbb{N}_+$$

$$w \mapsto \left| \bigcup_{t \leq \text{time}(w)} \text{used-tape}(\Delta^t(\text{init}(w))) \right|$$

$$\text{Tape} : \mathbb{N}_+ \rightarrow \mathbb{N}_+$$

$$n \mapsto \max\{\text{tape}(w) \mid w \in A^n\}$$

- ▶ *benutzt* heie wieder: mit Nichtblanksymbol beschriftet oder von einem Kopf besucht

## Speicherplatzkomplexität für $*W_*$ -PTM (1)

- ▶ ... und *Speicherplatzkomplexität* ... ? ...

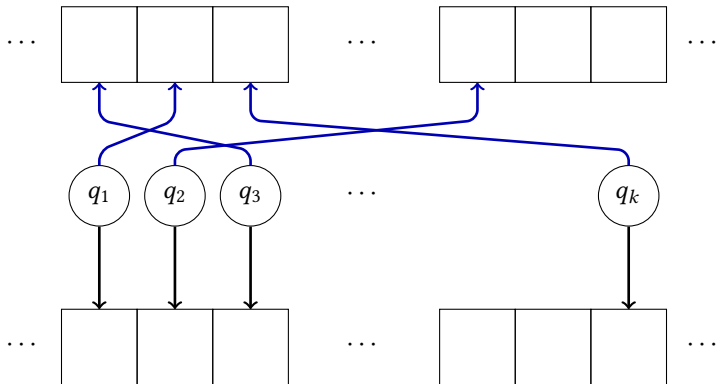
## Speicherplatzkomplexität für $*W_*$ -PTM (1)

- ▶ ... und *Speicherplatzkomplexität* ... ? ...
- ▶ Warum sollte man etwas anderes als die Bandkomplexität so bezeichnen?

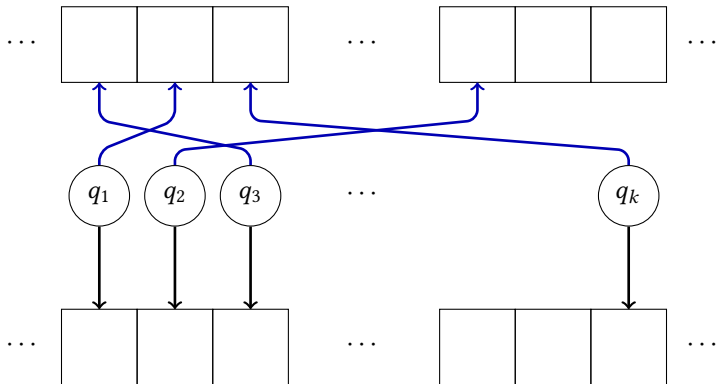
## Speicherplatzkomplexität für $*W_*$ -PTM (1)

- ▶ ... und *Speicherplatzkomplexität* ... ? ...
- ▶ Warum sollte man etwas anderes als die Bandkomplexität so bezeichnen?
- ▶ Man kann auch in der Graphstruktur, die durch die Steereinheiten aufgebaut ist, Informationen speichern!

# Informationsspeicherung in der Graphstruktur



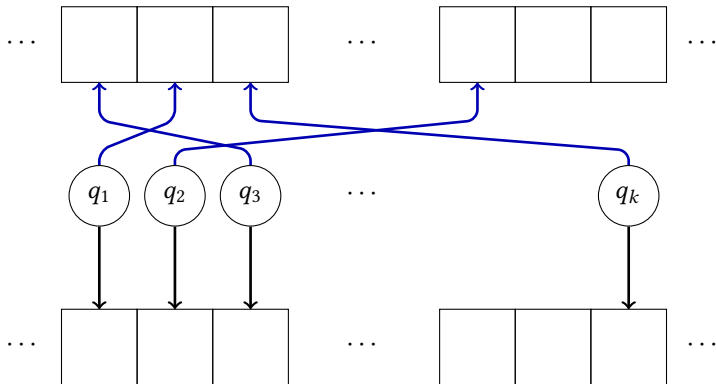
# Informationsspeicherung in der Graphstruktur



►  $k^k$  Kopfpositionen oben



# Informationsspeicherung in der Graphstruktur



- ▶  $k^k$  Kopfpositionen oben, also  $k \log k$  Bits Information

## Speicherkomplexität für $*W_*$ -PTM (2)

- ▶ Speicherkomplexität abhängig vom Verbrauch der Ressourcen „Felder“ und „Steereinheiten“
- ▶ *gegeben*: maximaler Feldverbrauch  $r_t$  pro Band und Prozessorverbrauch  $r_p$
- ▶ *gesucht*:  $r_s = \text{Space}(n)$  als  $r_s = f(r_t, r_p)$
- ▶ Konfiguration:  $c = (b_1, \dots, b_k, u_1, \dots, u_m)$
- ▶ Wieviele Konfigurationen  $c$  gibt es, deren Ressourcenverbrauch nicht  $r_t$  und  $r_p$  überschreitet?

## Speicherkomplexität für $\mathbb{W}_{h_1} \cdots \mathbb{W}_{h_k}$ -PTM (3)

- ▶ es sei  $h = h_1 + \cdots + h_k$ , und ein  $n$  fixiert
- ▶ Bandbeschriftungen:  $(\sum_{i=1}^{r_t} |B|^i)^k = \Theta(|B|^{kr_t})$
- ▶ Situationen für eine Steuereinheit:  $|Q|r_t^h$
- ▶ Situationen für maximal  $r_p$  Steuereinheiten:
  - ▶  $\sum_{i=1}^{r_p} \binom{|Q|r_t^h}{i}$
  - ▶ für  $r_p \leq |Q|r_t^h/2$  abschätzbar durch  $\binom{|Q|r_t^h}{r_p}$  von unten bzw.  $r_p \binom{|Q|r_t^h}{r_p}$  von oben
- ▶ Gesamtzahl Konfigurationen daher

$$\Theta(|B|^{kr_t} \binom{|Q|r_t^h}{r_p}) O(r_p)$$

## Speicherkomplexität für $\mathbb{W}_{h_1} \cdots \mathbb{W}_{h_k}$ -PTM (4)

- ▶ Anzahl Konfigurationen:  $\Theta(|B|^k r_t \binom{|Q| r_t^h}{r_p}) O(r_p)$
- ▶ wegen  $\frac{(x-y)^y}{y^y} \leq \binom{x}{y} \leq \frac{x^y}{y!} \leq \frac{x^y e^y}{y^y}$
- ▶ Logarithmus(Anzahl Konfigurationen) beschränkt durch:

$$\Theta(r_t + r_p + r_p \log \frac{|Q| r_t^h}{r_p})$$

- ▶ *Definition:*

$$\text{Space}(n) = \text{Tape}(n) + \text{Proc}(n) + \text{Proc}(n) \log \frac{|Q| \text{Tape}(n)^h}{\text{Proc}(n)}$$

## Speicherkomplexität: Beobachtungen

- ▶ Wenn  $\text{Proc}(n) \in O(\text{Tape}(n)/\log \text{Tape}(n))$ ,  
dann  $\text{Space}(n) \in \Theta(\text{Tape}(n))$ .
- ▶ Wenn  $\text{Proc}(n) \in \Theta(r(n))$  und  $\text{Tape}(n) \in \Theta(r(n))$ ,  
dann  $\text{Space}(n) \in \Theta(r(n) \log r(n))$ .
- ▶ Wenn  $\text{Proc}(n) \in \Theta(s(n)/\log s(n))$  und  $\text{Tape}(n) \in \Theta(s(n)/\log s(n))$ ,  
dann  $\text{Space}(n) \in \Theta(s(n))$ .

# Überblick

Parallele Turingmaschinen mit Ein-Kopf-Steuereinheiten

**Parallele Turingmaschinen mit Mehr-Kopf-Steuereinheiten**

Grundlagen

Erkennung formaler Sprachen

Komplexitätsmaße

Beziehungen zwischen TM und PTM

## Wie sinnvoll ist die Definition von Space?

- ▶ Grundlage war: Man *kann vielleicht* auf  $k$  Feldern mit  $k$  Steereinheiten  $k \log k$  Bits Information speichern.
- ▶ Kann „man“ wirklich?
- ▶ Können das PTM?
- ▶ Wenn ja: Wie schnell können die einzelnen Bits extrahiert werden?

## Von TM zu PTM

**Frage:** Ist effiziente Nutzung der Graphstruktur durch PTM beim Lesen und Schreiben möglich?



## Von TM zu PTM

**Frage:** Ist effiziente Nutzung der Graphstruktur durch PTM beim Lesen und Schreiben möglich?

**Satz (Worsch, 1997)**

$$\begin{aligned} & \mathbb{W}_1\text{-TM-SPC}(s)\text{-TIME}(t) \\ & \subseteq \mathbb{W}_2\text{-PTM-TAPE}(\Theta(s/\log s))\text{-PROC}(\Theta(s/\log s))\text{-TIME}(\Theta(t)) \\ & \subseteq \mathbb{W}_2\text{-PTM-SPC}(\Theta(s))\text{-TIME}(\Theta(t)) \end{aligned}$$

## Von TM zu PTM

**Frage:** Ist effiziente Nutzung der Graphstruktur durch PTM beim Lesen und Schreiben möglich?

~~Satz (Worsch, 1997)~~

Behauptung (Worsch, 2017)

$$\mathbb{W}_1\text{-TM-SPC}(s)\text{-TIME}(t)$$

$$\subseteq \mathbb{W}_2\text{-PTM-TAPE}(\Theta(s/\log s))\text{-PROC}(\Theta(s/\log s))\text{-TIME}(\text{Pol}(t))$$

$$\subseteq \mathbb{W}_2\text{-PTM-SPC}(\Theta(s))\text{-TIME}(\text{Pol}(t))$$

## Idee für effiziente Nutzung der Graphstruktur



D. R. Luginbuhl, M. C. Loui

Hierarchies and Space Measures for Pointer Machines.

*Information and Computation* **104**, S. 253–270, 1993.

Was ist in der folgenden Arbeit falsch?



T. Worsch

On parallel Turing machines with multi-head control units.

*Parallel Computing* **23**, S. 1683–1697, 1997.

# Effiziente Nutzung der Graphstruktur: Skizze

Bandblöcke  $B_1, \dots, B_{s/b}$

Blockgröße  $b = \log(s/\log s)$

Blockzahl  $s/b = O(s/\log s)$

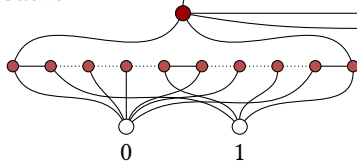
**Cache:** zwei Blöcke „decodiert“

**Blockset:** alle Blöcke „codiert“

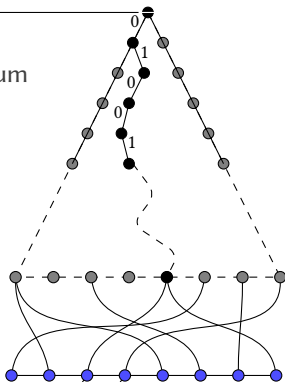
**Baum:** Codierungen

- ▶ Höhe:  $b$
- ▶ Knoten:  $\approx 2s/\log s$

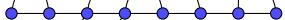
**Cache**



**Baum**



**Blockset**



## Von PTM zu TM

Aufgrund der gewählten Definition von Speicherkomplexität gilt umgekehrt:

### Satz

$$*W_*\text{-PTM-SPC}(\Theta(s))\text{-TIME}(\text{Pol}(t))$$

$$\subseteq *W_*\text{-TM-SPC}(\Theta(s))\text{-TIME}(\text{Pol}(t))$$

# Zusammenfassung

- ▶  $\mathbb{W}_1$ -PTM benötigen zur Erkennung von  $L_{vv}$  nur Linearzeit (im Gegensatz zu  $\mathbb{W}_1$ -TM).
- ▶  $*\mathbb{W}_*$ -PTM sind ein Modell, bei dem man sich Gedanken über die Definition des Speicherkomplexität machen sollte.
- ▶ Vergleich von PTM mit TM:
  - ▶ asymptotisch gleicher Speicherbedarf und gleichzeitig
  - ▶ polynomieller Zusammenhang beim Zeitbedarf