

# Modelle der Parallelverarbeitung

## 6. Alternierende Turingmaschinen

Thomas Worsch

Institut für Theoretische Informatik  
Karlsruher Institut für Technologie

Sommersemester 2020

# Überblick

## Das Modell

Grundlagen

Erkennung formaler Sprachen

Erstes Beispiel

Komplexitätsmaße

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

Beziehungen zu sequenziellen TM

# Überblick

## Das Modell

### Grundlagen

Erkennung formaler Sprachen

Erstes Beispiel

Komplexitätsmaße

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

## Beziehungen zu sequenziellen TM

# Alternierende Turingmaschinen: Ursprünge



D. Kozen

On parallelism in Turing machines

*17<sup>th</sup> IEEE Symp. on Foundations of Computer Science*, 89-97,  
1976.



A. K. Chandra, L. J. Stockmeyer

Alternation

*17<sup>th</sup> IEEE Symp. on Foundations of Computer Science*, 98-108,  
1976.



A. K. Chandra, D. Kozen, L. J. Stockmeyer

Alternation

*Journal of the ACM* 28:114-133, 1981

# Alternierende Turingmaschinen: Idee

- ▶ Verallgemeinerung nichtdeterministischer TM
- ▶ kein simples «es gibt eine akzeptierende Berechnung»
- ▶ sondern z. B. auch so etwas wie:

# Alternierende Turingmaschinen: Idee

- ▶ Verallgemeinerung nichtdeterministischer TM
- ▶ kein simples «es gibt eine akzeptierende Berechnung»
- ▶ sondern z. B. auch so etwas wie:  
«alle Berechnung sind akzeptierend»

## Definition: Bestandteile von $\mathbb{W}_1$ -ATM

Grundbestandteile wie bei TM:

- ▶ *Bandalphabet*  $B$
- ▶ *Zustandsmenge*  $S$ ,
  - ▶ weiter unterteilt; siehe Folie 10
- ▶ *Überföhrungsrelation*  $\delta \subset (S \times B) \times (S \times B \times D)$
- ▶ *Konfiguration* (wie bei TM):  
 $c = (s, b, p)$  mit
  - ▶ Zustand  $s \in S$
  - ▶ Bandbeschriftung  $b : \mathbb{Z} \rightarrow B$ , i. e.  $b \in B^{\mathbb{Z}}$
  - ▶ Kopfposition  $p \in \mathbb{Z}$

schreibe  $C_T$  oder  $C$  für die Menge aller Konfigurationen

- ▶ analog für  $*\mathbb{W}_*$ -ATM

# Definition: Schrittrelation

## *mögliche globale Schritte*

- ▶ Relation  $\vdash \subseteq C \times C$ , wobei  
 $(s, b, p) \vdash (s', b', p')$  naheliegend definiert sei



# Überblick

## Das Modell

Grundlagen

Erkennung formaler Sprachen

Erstes Beispiel

Komplexitätsmaße

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

Beziehungen zu sequenziellen TM

## Eingabe von Wörtern

- ▶ Anfangszustand  $s_0 \in S$
- ▶ Eingabealphabet  $A \subset B$
- ▶ Blanksymbol  $\square$
- ▶ *Anfangskonfiguration* zu  $w$  wie bei TM

## Definition

- ▶ Zustandsmenge  $S$  unterteilt:
  - ▶  $S = S_U \dot{\cup} S_E \dot{\cup} F_+ \dot{\cup} F_-$
  - ▶  $S_U$ : *universelle* Zustände,  
 $S_E$ : *existenzielle* Zustände
- ▶ Eine Konfiguration  $(s, b, p)$  heißt
  - ▶ *existenziell*, falls  $s \in S_E$  ist,
  - ▶ *universell*, falls  $s \in S_U$  ist,
  - ▶ *akzeptierende Endkonfiguration*, falls  $s \in F_+$  ist,
  - ▶ *ablehnende Endkonfiguration*, falls  $s \in F_-$  ist.

## Definition Akzeptieren

- ▶ *Berechnungsbaum* für  $c_r$  ist ein Baum mit Eigenschaften:
  - ▶ Wurzel ist  $c_r$ .
  - ▶ alle Blätter und nur die sind Endkonfigurationen.
  - ▶ Jede existenzielle Konfiguration  $c$  hat genau eine Konfiguration  $c'$  mit  $c \vdash c'$  als Baum-Nachfolger.
  - ▶ Jede universelle Konfiguration  $c$  hat alle Konfigurationen  $c'$  mit  $c \vdash c'$  als Baum-Nachfolger.
- ▶ Ein Berechnungsbaum heißt *akzeptierend*, wenn
  - ▶ er endlich ist und
  - ▶ alle Blätter akzeptierende Endkonfigurationen sind.
- ▶ Eingabewort  $w$  gehört zur von  $T$  *akzeptierten Sprache*  $L(T)$ , wenn es akzeptierenden Berechnungsbaum mit Wurzel  $\text{init}(w)$  gibt.

# Überblick

## Das Modell

Grundlagen

Erkennung formaler Sprachen

**Erstes Beispiel**

Komplexitätsmaße

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

Beziehungen zu sequenziellen TM

## «Paralleles» Überprüfen mehrerer Möglichkeiten

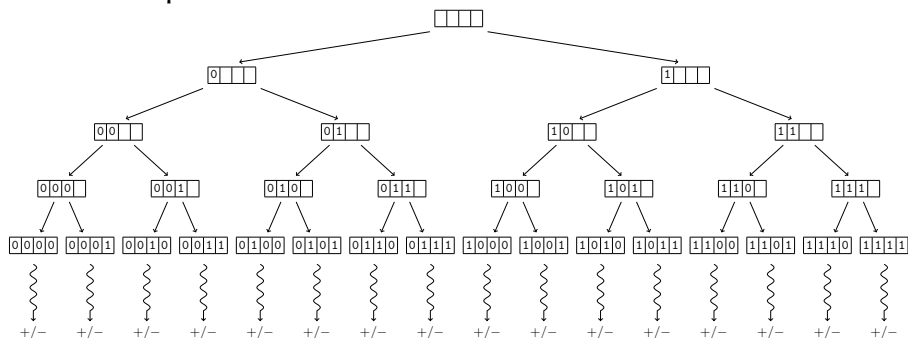
- ▶ auf dem Band sei ein endlicher Bereich markiert
- ▶ Kopf auf erstem Feld am linken Ende
- ▶  $\delta$  erlaubt
  1. Phase «Initialisierung»
    - ▶ immer im gleichen Zustand  $s_i$
    - ▶ Schreiben einer 0 bzw. 1
    - ▶ Kopfbewegung nach rechts
    - ▶ bis Ende des markierten Bereichs erreicht
    - ▶ dann deterministisch zurück an linkes Ende
  2. Phase «Test»
    - ▶ Berechnung für die in Phase 1 erzeugte Zahl

## «Paralleles» Überprüfen von Möglichkeiten (2)

- ▶ Was passiert?

# «Paralleles» Überprüfen von Möglichkeiten (2)

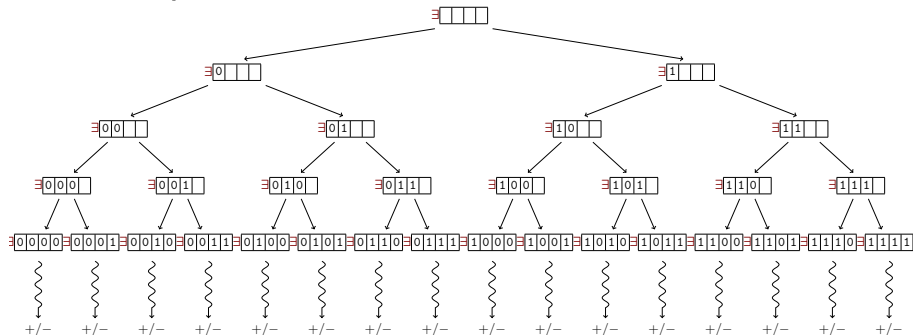
## ► Was passiert?





# «Paralleles» Überprüfen von Möglichkeiten (2)

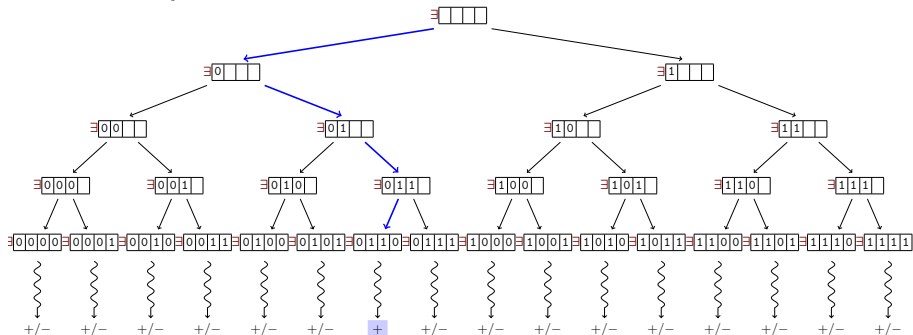
## ► Was passiert?



## ► Was passiert, wenn $s_i \in S_E$ ?

## «Paralleles» Überprüfen von Möglichkeiten (2)

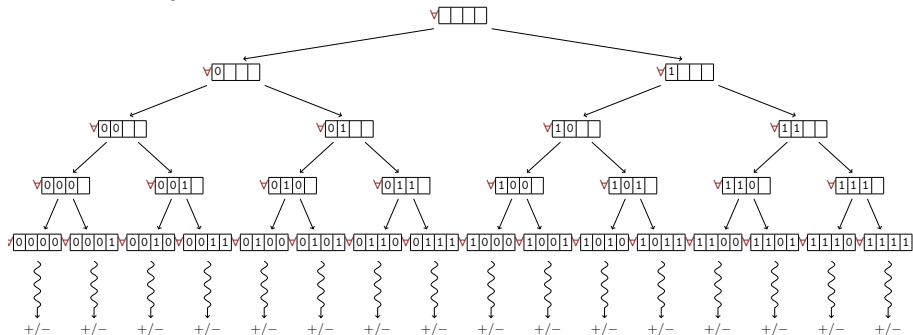
- Was passiert?



- Was passiert, wenn  $s_i \in S_E$ ?      TM akzeptiert,  
falls für *eine* Zahl am Ende akzeptierende Konfiguration

## «Paralleles» Überprüfen von Möglichkeiten (2)

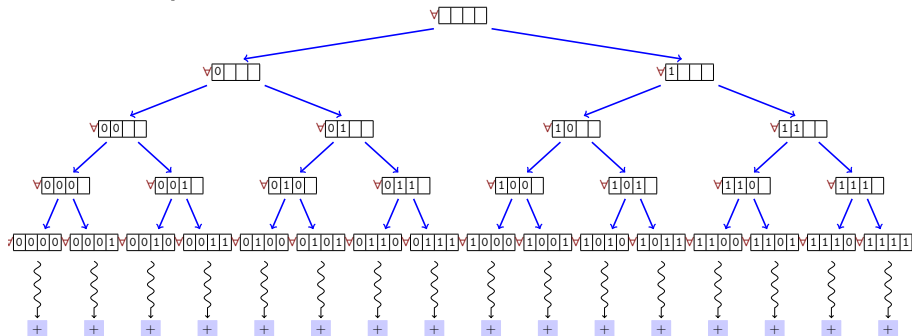
- ▶ Was passiert?



- ▶ Was passiert, wenn  $s_i \in S_E$ ?      TM akzeptiert,  
falls für *eine* Zahl am Ende akzeptierende Konfiguration
- ▶ Was passiert, wenn  $s_i \in S_U$ ?

# «Paralleles» Überprüfen von Möglichkeiten (2)

## ► Was passiert?



- Was passiert, wenn  $s_i \in S_E$ ?      TM akzeptiert,  
falls für *eine* Zahl am Ende akzeptierende Konfiguration
- Was passiert, wenn  $s_i \in S_U$ ?      TM akzeptiert,  
falls für *alle* Zahlen am Ende akzeptierende Konfiguration

# Überblick

## Das Modell

Grundlagen

Erkennung formaler Sprachen

Erstes Beispiel

**Komplexitätsmaße**

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

Beziehungen zu sequenziellen TM

# Ressourcenbeschränkung von ATM

- ▶ Eine formale Sprache  $L$  gehört zu

$$\text{ATM-SPC}(s(n))\text{-TIME}(t(n))$$

wenn es eine ATM gibt, bei der für jedes Wort  $w \in L$  ein akzeptierender Berechnungsbaum existiert,

- ▶ dessen Höhe kleiner oder gleich  $t(|w|)$  ist und
  - ▶ bei dem alle auftretenden Konfigurationen einen Platzbedarf von kleiner oder gleich  $s(|w|)$  Feldern hat.
- ▶ analog, wenn nur eine Ressource beschränkt ist

# Erweiterung von ATM für sublineare Zeitschranken

- ▶ PRAM, UCIR: jedes Eingabebit «schnell» zugreifbar
- ▶ ATM, bisher:  $n$  Schritte, bis letztes Bit lesbar
- ▶ Abhilfe: statt Lesekopf auf dem Eingabeband ein *Indexband*:
  - ▶ Beschriftung wird als eine Zahl  $i$  interpretiert
  - ▶ in speziellem Zustand  $q_{index}$  der TM wird  $i$ -tes Eingabesymbol zur Verfügung gestellt
  - ▶ oder: TM «sieht» zu jedem Zeitpunkt jeweils  $i$ -tes Eingabesymbol
- ▶ manchmal «random access ATM» genannt

# Übliche Abkürzungen

$$\text{ALOGTIME} = \text{ATM-TIME}(\log n)$$

$$\text{AL} = \text{ALOGSPACE} = \text{ATM-SPC}(\log n)$$

$$\text{AP} = \text{APTIME} = \text{ATM-TIME}(\text{Pol}(n))$$

$$\text{APSPACE} = \text{ATM-SPC}(\text{Pol}(n))$$



# Überblick

## Das Modell

Grundlagen

Erkennung formaler Sprachen

Erstes Beispiel

Komplexitätsmaße

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

Beziehungen zu sequenziellen TM

# $L_{vv}$ : Algorithmusidee

## $L_{vv}$ : Algorithmusidee

verzweige universell für alle Zahlen  $i$  zwischen 0 und  $|w|/3$  und überprüfe,

- ▶ ob die Symbole  $i$  und  $i + 2|w|/3$  gleich sind und keine 2, und
- ▶ ob Symbol  $i + |w|/3$  eine 2 ist

Zeitbedarf:

- ▶ ATM ohne Indexband: linear
- ▶ ATM mit Indexband: logarithmisch

# Überblick

## Das Modell

## Beziehungen zu sequenziellen TM

Zusammenhang zwischen  $TM\text{-SPC}$  und  $ATM\text{-TIME}$

Zusammenhang zwischen  $ATM\text{-SPC}$  und  $TM\text{-TIME}$

# Überblick

## Das Modell

## Beziehungen zu sequenziellen TM

Zusammenhang zwischen TM-SPC und ATM-TIME

Zusammenhang zwischen ATM-SPC und TM-TIME

# Simulation NTM-SPC $\rightsquigarrow$ ATM-TIME

## Satz

Falls  $s(n) \geq n$  ist, ist

$$\text{NTM-SPC}(s(n)) \subseteq \text{ATM-TIME}(O((s(n))^2))$$

Für ATM mit Indexband gilt das Ergebnis für  $s(n) \geq \log n$ .

# Simulation NTM-SPC $\rightsquigarrow$ ATM-TIME (1)

## Beweisskizze

# Simulation NTM-SPC $\rightsquigarrow$ ATM-TIME (1)

## Beweisskizze

- ▶ ATM «rät» (jeweils existenzielle Verzweigungen)
  - ▶ potenzielle Anfangskonfiguration  $c$  der NTM,
  - ▶ eine Schrittzahl  $t$  und
  - ▶ potenzielle Endkonfiguration  $c'$  der NTM



# Simulation NTM-SPC $\rightsquigarrow$ ATM-TIME (1)

## Beweisskizze

- ▶ ATM «rät» (jeweils existenzielle Verzweigungen)
  - ▶ potenzielle Anfangskonfiguration  $c$  der NTM,
  - ▶ eine Schrittzahl  $t$  und
  - ▶ potenzielle Endkonfiguration  $c'$  der NTM
- ▶ ATM prüft,
  - ▶ ob  $c$  zur Eingabe passende Anfangskonfiguration
  - ▶ ob  $c'$  eine Endkonfiguration
  - ▶ ob für die NTM  $c \vdash^t c'$  gilt
- ▶ ATM akzeptiert, falls
  - ▶ Überprüfungen positiv
  - ▶  $c'$  akzeptierend

## Simulation NTM-SPC $\rightsquigarrow$ ATM-TIME (2)

### Beweisskizze

- ▶ zu zwei Konfigurationen  $c, c'$  und  $t$   
überprüft die ATM  $c \vdash^t c'$  folgendermaßen
  - ▶ Fall  $t = 1$ : klar

# Simulation NTM-SPC $\rightsquigarrow$ ATM-TIME (2)

## Beweisskizze

- ▶ zu zwei Konfigurationen  $c, c'$  und  $t$  überprüft die ATM  $c \vdash^t c'$  folgendermaßen
  - ▶ Fall  $t = 1$ : klar
  - ▶ Fall  $t > 1$ : ATM rät existenziell eine Konfiguration  $\bar{c}$  und
  - ▶ überprüft rekursiv  $c \vdash^{\lceil t/2 \rceil} \bar{c}$  und  $\bar{c} \vdash^{\lfloor t/2 \rfloor} c'$ 
    - ▶ universell nach dem gleichen Verfahren

# Simulation ATM-TIME $\rightsquigarrow$ TM-SPC

## Satz

Falls  $t(n) \geq n$  ist, ist

$$\text{ATM-TIME}(t(n)) \subseteq \text{TM-SPC}(t(n))$$

Für ATM mit Indexband gilt das Ergebnis für  $t(n) \geq \log n$ .

# Simulation ATM-TIME $\rightsquigarrow$ TM-SPC

## Beweisskizze

- ▶ DTM durchläuft Baum aller möglichen ATM-Berechnungen bis maximal zur Tiefe  $t(n)$
- ▶ im wesentlichen zu jedem Zeitpunkt gespeichert:
  - ▶ die Konfiguration eines Baumknotens
  - ▶ der Pfad von der Wurzel zu diesem Knoten

# Korollar

Falls  $s(n) \geq n$  ist, ist

$$\text{TM-SPC}(\text{Pol}(s(n))) = \text{ATM-TIME}(\text{Pol}(s(n)))$$

Für ATM mit Indexband gilt das Ergebnis für  $s(n) \geq \log n$ .

# Überblick

## Das Modell

## Beziehungen zu sequenziellen TM

Zusammenhang zwischen TM-SPC und ATM-TIME

Zusammenhang zwischen ATM-SPC und TM-TIME

# Simulation ATM-Spc $\rightsquigarrow$ TM-TIME

## Satz

Falls  $s(n) \geq \log n$  ist, ist

$$\text{ATM-Spc}(s(n)) \subseteq \text{TM-TIME}(2^{O(s(n))})$$

Der Beweis ist uninteressant.



# Simulation $TM\text{-TIME} \rightsquigarrow ATM\text{-SPC}$

## Satz

Falls  $t(n) \geq n$  ist, ist

$$TM\text{-TIME}(t(n)) \subseteq ATM\text{-SPC}(O(\log t(n)))$$

# Simulation TM-TIME $\rightsquigarrow$ ATM-SPC

## Beweisskizze (1)

- ▶ ersetze TM durch  $\mathbb{W}_1$ -TM
  - ▶ die nie Felder  $i < 0$  betritt
  - ▶  $t'(n) \leq t(n)^2$
  - ▶ nur ein akzeptierender Zustand  $q_a$
- ▶ codiere Konfigurationen ähnlich im Kapitel über PRAM:
  - ▶  $\$b[0 \dots i-1] \cdot q \cdot b[i \dots m]$
  - ▶ wenn TM Feld  $i$  in Zustand  $q$  besucht;
- ▶  $g_{t,j}$  bezeichne Symbol  $j$  in Codierung zum Zeitpunkt  $t$
- ▶  $g_{t+1,j}$  ergibt sich eindeutig aus  $g_{t,j-1}g_{t,j}g_{t,j+1}g_{t,j+2}$

# Simulation TM-Time $\rightsquigarrow$ ATM-Spc

## Beweisskizze (2)

- ▶ ATM rät zunächst finales  $t$  und Kopfposition  $j$
- ▶ überprüft ob  $g_{t,j} = q_a$ , und zwar so:
- ▶ allgemein wird  $g_{t+1,j}$  bestimmt durch
  - ▶ existenzielles Raten eines Quadrupels  $(g_{t,j-1}g_{t,j}g_{t,j+1}g_{t,j+2})$
  - ▶ rekursiv universelles Überprüfen der geratenen Werte und
  - ▶ Überprüfung dass  $g_{t+1,j}$  aus  $g_{t,j-1}g_{t,j}g_{t,j+1}g_{t,j+2}$  folgt
  - ▶ Fälle  $t = 0$  und  $j = 0$  trivial
- ▶ Platzbedarf dominiert durch die Zähler für  $j$  und  $t$

# Simulation TM-Time $\rightsquigarrow$ ATM-Spc

## Beweisskizze (2)

- ▶ ATM rät zunächst finales  $t$  und Kopfposition  $j$
- ▶ überprüft ob  $g_{t,j} = q_a$ , und zwar so:
- ▶ allgemein wird  $g_{t+1,j}$  bestimmt durch
  - ▶ existenzielles Raten eines Quadrupels  $(g_{t,j-1}g_{t,j}g_{t,j+1}g_{t,j+2})$
  - ▶ rekursiv universelles Überprüfen der geratenen Werte und
  - ▶ Überprüfung dass  $g_{t+1,j}$  aus  $g_{t,j-1}g_{t,j}g_{t,j+1}g_{t,j+2}$  folgt
  - ▶ Fälle  $t = 0$  und  $j = 0$  trivial
- ▶ Platzbedarf dominiert durch die Zähler für  $j$  und  $t$

!! Determinismus der seq. TM wichtig

- ▶ wo / warum ?

## Die Ergebnisse im Bild

$$\begin{array}{l} \text{sequenziell:} \quad L \subseteq \mathbf{P} \subseteq \mathbf{PSPACE} \subseteq \text{EXPTIME} \\ \text{parallel:} \quad \quad \quad \parallel \quad \quad \parallel \quad \quad \parallel \\ \quad \quad \quad \text{AL} \subseteq \quad \quad \text{AP} \quad \quad \subseteq \text{APSPACE} \end{array}$$

Diese Hierarchie setzt sich nach oben fort.

# Zusammenfassung

- ▶ Durch Einführung eines Indexbandes kann man bei ATM sinnvoll Zeitkomplexitäten bis hinunter zu  $\log n$  betrachten.
- ▶ polynomieller Zusammenhang von sequenziellem Platzbedarf und alternierendem Zeitbedarf