

# Modelle der Parallelverarbeitung

## 8. Registermaschinen mit mächtigen Erweiterungen

Thomas Worsch

Institut für Theoretische Informatik  
Karlsruher Institut für Technologie

Sommersemester 2018

# Überblick

Registermaschinen mit Vektoren

Registermaschinen mit mächtigen Maschinenbefehlen

Registermaschinen mit Editor-Erweiterung

# Ursprung von VRAM



V. R. Pratt, M. O. Rabin, L. J. Stockmeyer

A characterization of the power of vector machines

*Proc. 6<sup>th</sup> Symp. on Theory of Computing*, 122–134, 1974.



V. R. Pratt, L. J. Stockmeyer

A characterization of the power of vector machines

*Journal of Computer and System Sciences*, 198–221, 1976.

## Idee von VRAM

- ▶ ähnlich sequenzieller RAM
- ▶ zwei Sorten von Registern:
  - ▶ Indexregister: Inhalt als Zahl interpretiert
  - ▶ Vektorregister: Inhalt als Bitvektor interpretiert

## Neue Maschinenbefehle bei VRAM

- ▶ bitweise logische Operationen auf Vektorregistern
- ▶ verschiebe Bits in Vektorregister um  $k$  Stellen;  
 $k$  stammt aus Indexregister
- ▶ *kein Transfer* von Vektor- in Indexregister
- ▶ Annahme: *Alle Maschinenbefehle können in konstanter Zeit ausgeführt werden.*

# Satz

Für  $t(n) \geq n$  gilt:

$$\text{VRAM-TIME}(\text{Pol}(t(n))) = \text{TM-SPC}(\text{Pol}(t(n)))$$

Beweis in der Originalarbeit: technisch etwas aufwändig

# Überblick

Registermaschinen mit Vektoren

Registermaschinen mit mächtigen Maschinenbefehlen

Registermaschinen mit Editor-Erweiterung

# Ursprung von MRAM



J. Hartmanis, J. Simon

On the Power of Multiplication In Random Access Machines

*15<sup>th</sup> Ann. Symp. on Switching and Automata Theory*, 13–23,  
1974.



## Idee von MRAM

- ▶ ähnlich sequenzieller RAM
- ▶ Beobachtung: Verschiebungen um 1 Bit bei VRAM entsprechen Multiplikation mit 2 und Division durch 2
- ▶ erlaube beliebige Multiplikationen und Divisionen
- ▶ Annahme: *Alle Maschinenbefehle können in konstanter Zeit ausgeführt werden.*

## Variationen des Befehlssatzes bei MRAM

- ▶ indirekte Adressierungen „überflüssig“  
(modulo polynomiellen Zeitbedarfs)
- ▶ Verschiebungen nach links genügen  
(nach rechts „überflüssig“ modulo polynomiellen Zeitbedarfs)
- ▶ Multiplikation und Division machen bitweise Operationen überflüssig

## Satz

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

Eine etwas mühsame Konstruktion.

# Überblick

Registermaschinen mit Vektoren

Registermaschinen mit mächtigen Maschinenbefehlen

Registermaschinen mit Editor-Erweiterung

# Ursprung



R. A. Stegwee, L. Torenvliet, P. van Emde Boas

The power of your editor

IBM Research Report RJ 4711 (50179) 1985.

- ▶ Viel Glück bei der Suche :- (
- ▶ das beste, was ich noch finden konnte:
  - ▶ <http://staff.science.uva.nl/~peter/teaching/thmod03.html>  
und dortiger Link
  - ▶ <http://staff.science.uva.nl/~peter/teaching/seqpar95.ps> (Achtung: Seiten in umgekehrter Reihenfolge)
- ▶ das zweitbeste: Kapitel 1 von Band A des *Handbook of Theoretical Computer Science*.

## Idee von EDITRAM

- ▶ sequenzielle RAM und zusätzlich
- ▶ endliche viele Textdateien
- ▶ Registerinhalte können als Cursor-Position in Textdatei interpretiert werden

## Neue Maschinenbefehle bei EDITRAM

- ▶ lese Symbol an Cursor-Position
- ▶ schreibe Symbol an Cursor-Position
- ▶ kopiere Textsegment (durch zwei Cursor-Positionen markiert) an eine Cursor-Position
- ▶ ersetze in einer Textdatei *alle Vorkommen* eines *konstanten* Strings durch anderen *konstanten* String
- ▶ Annahme: *Alle Maschinenbefehle können in konstanter Zeit ausgeführt werden.*

## Satz

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



## Beweis

⊆: benutze nichtdeterministische TM, ... alles Standard ...

⊇: zeige: QBF in Polynomialzeit lösbar

Hauptideen:

# Beweis

⊆: benutze nichtdeterministische TM, ... alles Standard ...

⊇: zeige: QBF in Polynomialzeit lösbar

Hauptideen:

- ▶ Quantorenelimination von innen nach außen  
bei gleichzeitiger Verdoppelung der Formellänge:
  - ▶ aus  $\forall x_i : F(\dots, x_i, \dots)$   
macht man  $(F(\dots, 0, \dots) \wedge F(\dots, 1, \dots))$
  - ▶ aus  $\exists x_i : F(\dots, x_i, \dots)$   
macht man  $(F(\dots, 0, \dots) \vee F(\dots, 1, \dots))$
  - ▶ technisches Problem:  $i$  nicht konstant, sondern binär codiert
- ▶ Auswerten durch Ersetzung wie z. B.
  - ▶  $\neg 0 \rightsquigarrow 1$  etc.
  - ▶  $0 \wedge 1 \rightsquigarrow 0$  etc.
  - ▶  $(0) \rightsquigarrow 0$  etc.

# Zusammenfassung

- ▶ Zeitmaß 1 für alle Maschinenbefehle, einschließlich solcher, die „große“ Datenmengen manipulieren, führen zu mächtigen Varianten von „sequenziellen“ Registermaschinen
- ▶ Polynomialzeit bei allen erwähnten Modellen ebenso mächtig wie Polynomialraum bei sequenziellen TM