Algorithmen in Zellularautomaten

2. Berechnungsmächtigkeit von Zellularautomaten

Thomas Worsch

Fakultät für Informatik
Institut für Theoretische Informatik

Sommersemester 2018

Überblick

Überblick

Simulation von Schaltwerken

Simulation von Turingmaschinen

Beispiel `WIREWORLD`

Beispiel `WIREWORLD`

«Elektronen laufen über Drähte von einem Gatter zum nächsten»

2.3 Satz

Es gibt einen Zellularautomaten (nämlich `WIREWORLD`) mit

$$R = \mathbb{Z}^2, \qquad N = M_1^{(2)}, \qquad |Q| = 4 \qquad \text{und geeignetem } \delta,$$

mit dem man jeden endlichen Automaten (mit beliebig großer Zustandsmenge und beliebiger Überführungsfunktion) simulieren kann.

Beachte

- hier keine Präzisierung des Begriffes «Simulation»
- **ein** Zellularautomat für **alle** endlichen Automaten
  sonst wäre es trivial …
- Kodierung des EA in der Anfangskonfiguration

2.4 Beweisidee

- endlichen Automaten aus WIREWORLD–Gattern zusammensetzen
- Problem: Timing (insbesondere: Inverter!)
- Lösung: Gatter einbetten in normierte Module:
  - Größe 24×24
  - Laufzeit Eingang $\longrightarrow$ Ausgang: 48 Takte
  - 1-Bit: Zeitintervall der Länge 24 mit einem Elektron
  - 0-Bit: Zeitintervall der Länge 24 ohne ein Elektron

2.4 Beweisidee

- endlichen Automaten aus `WIREWORLD`–Gattern zusammensetzen
- Problem: Timing (insbesondere: Inverter!)
- Lösung: Gatter einbetten in normierte Module:
    - Größe 24×24
    - Laufzeit Eingang $\longrightarrow$ Ausgang: 48 Takte
    - 1-Bit: Zeitintervall der Länge 24 mit einem Elektron
    - 0-Bit: Zeitintervall der Länge 24 ohne ein Elektron

alternative Konstruktion

2.4 Beweisidee

- endlichen Automaten aus `WIREWORLD`–Gattern zusammensetzen
- Problem: Timing (insbesondere: Inverter!)
- Lösung: Gatter einbetten in normierte Module:
  - Größe 24×24
  - Laufzeit Eingang $\longrightarrow$ Ausgang: 48 Takte
  - 1-Bit: Zeitintervall der Länge 24 mit einem Elektron
  - 0-Bit: Zeitintervall der Länge 24 ohne ein Elektron

alternative Konstruktion

- ein logischer Draht — zwei physikalische Drähte
- 1-Bit: Elektron auf dem einen phys. Draht
- 0-Bit: Elektron auf dem anderen phys. Draht
- Inverter — Kreuzung physikalischer Drähte

Eine ganze CPU

und weitere Hinweise findet man auf der Seite

`http://www.quinapalus.com/wi-index.html`

z. B. einen Ansatz, wie man das ganze auch unempfindlich gegenüber Verzögerungen auf Leitungen machen kann, fand man auf

`http://www.with-logic.co.uk/WireWorld.htm`

- Auf `http://www.archive.org` findet man noch Teile einer Version von April 2015.

2.5 Bemerkungen

- Mit `WIREWORLD` kann man sogar einen programmgesteuerten Universalrechner simulieren.
- **Problem:** Speicherplatz
  - Größe nicht beschränkbar
  - $\Rightarrow$ in der Anfangskonfiguration unendlich viel Draht
- **Aber:** Diese Anfangskonfiguration ist endlich beschreibbar!

2.5 Bemerkungen

- Mit `WIREWORLD` kann man sogar einen programmgesteuerten Universalrechner simulieren.
- **Problem:** Speicherplatz
  - Größe nicht beschränkbar

  $\Rightarrow$ in der Anfangskonfiguration unendlich viel Draht
- **Aber:** Diese Anfangskonfiguration ist endlich beschreibbar!

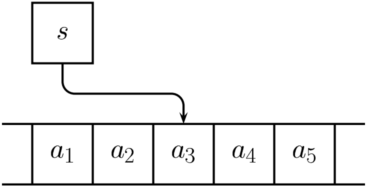
Alternativen

- `BANKS` und `LIFE`
  - gleiches Prinzip, kompliziertere Details
- *verbesserter ZA* von Banks:
  - *Erzeugung zusätzlicher Drähte* möglich

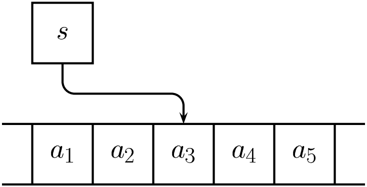
  $\Rightarrow$ endliche Anfangskonfiguration reicht

Überblick

Definition (Turingmaschine, 1 Band, 1 Kopf)



Definition (Turingmaschine, 1 Band, 1 Kopf)



- **Bestandteile**
  - endliche Zustandsmenge Q
  - endliches Bandalphabet B
  - Überführungsfunktion $\delta : Q \times B \rightarrow Q \times B \times \{-1, 0, 1\}$
- **Konfiguration einer TM**
  - $c = (s, b, p) \in Q \times B^{\mathbb{Z}} \times \mathbb{Z}$

Definition (Schritt einer TM)

Konfigurationsüberführungsfunktion

- durch $\delta : Q \times B \to Q \times B \times \{-1, 0, 1\}$ induziert

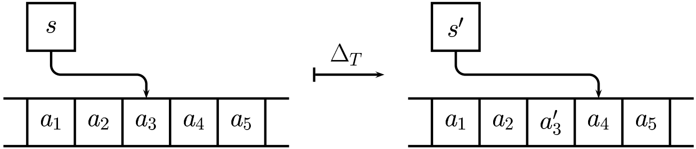
$$\Delta : Q \times B^{\mathbb{Z}} \times \mathbb{Z} \to Q \times B^{\mathbb{Z}} \times \mathbb{Z}$$

- Für $c = (s, b, p)$ ist $\Delta(c) = (s', b', p')$ mit
  - $s' = \delta(s, b(p))[1]$,
  - für alle $i \in \mathbb{Z}$ ist $b'(i) = \begin{cases} \delta(s, b(p))[2] & \text{falls } i = p \\ b(i) & \text{falls } i \neq p \end{cases}$
  - $p' = p + \delta(s, b(p))[3]$.

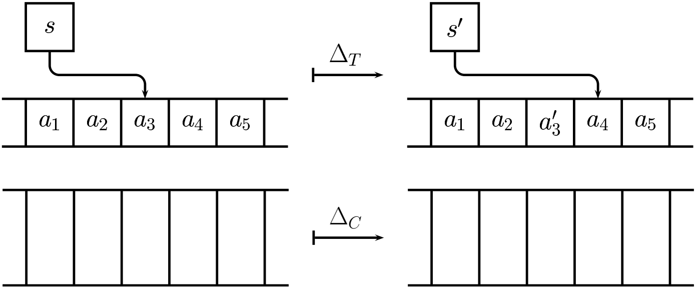
Satz

Zu jeder TM $T = (Q_T, B_T, \delta_T)$ existiert ein eindimensionaler ZA $C = (Q_C, \delta_C, H_1^{(1)})$, der T ohne Zeitverlust schrittweise simuliert.

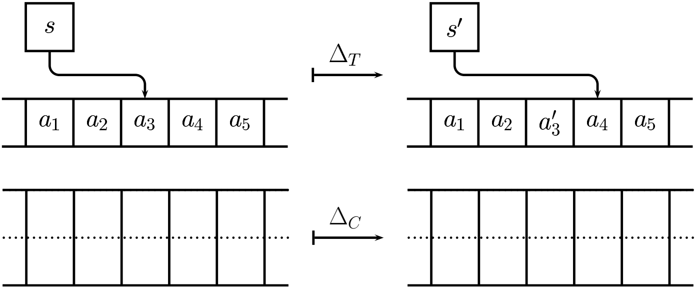
Beweisskizze: Idee



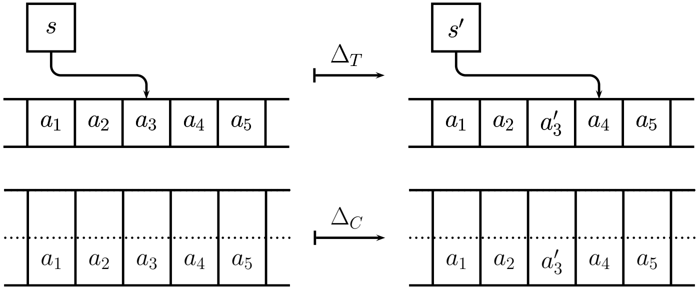
Beweisskizze: Idee



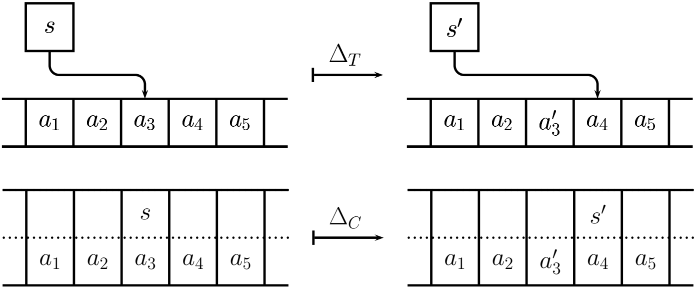
Beweisskizze: Idee



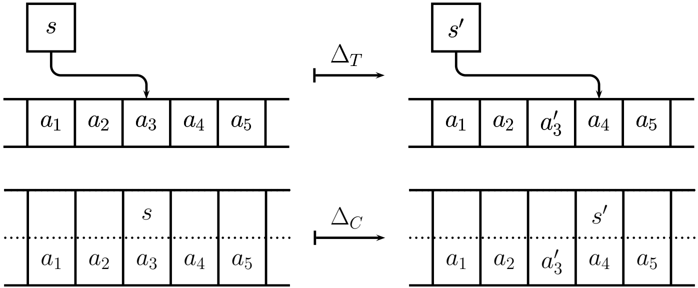
Beweisskizze: Idee



Beweisskizze: Idee



Beweisskizze: Idee



- setze $Q_C = (Q_T \dot{\cup} \{\_\}) \times B_T$
- repräsentiere $c_T = (s, b, p)$ durch c_C

$$\begin{array}{cccl} c_C & : & \mathbb{Z} & \longrightarrow & Q_C \\ & & i & \mapsto & \begin{cases} (s, b(p)) & \text{falls } i = p \\ (\_, b(i)) & \text{falls } i \neq p \end{cases} \end{array}$$

Beweisskizze: lokale Überführungsfunktion

Setze (unter anderem)

$$\delta_C((\_, a_l), (\_, a_m), (\_, a_r)) \quad = \quad (\_, a_m)$$

Beweisskizze: lokale Überführungsfunktion

Setze (unter anderem)

$$
\begin{aligned}
\delta_C((\_\,, a_l), (\_\,, a_m), (\_\,, a_r)) &= (\_\,, a_m) \\
\delta_C((s, a_l), (\_\,, a_m), (\_\,, a_r)) &= \begin{cases} (s', a_m) & \text{falls } \delta_T(s, a_l) = (s', a_l', 1) \\ (\_\,, a_m) & \text{sonst} \end{cases}
\end{aligned}
$$

Beweisskizze: lokale Überführungsfunktion

Setze (unter anderem)

$$
\begin{aligned}
\delta_C((\_, a_l), (\_, a_m), (\_, a_r)) &= (\_, a_m) \\
\delta_C((s, a_l), (\_, a_m), (\_, a_r)) &= \begin{cases} (s', a_m) & \text{falls } \delta_T(s, a_l) = (s', a_l', 1) \\ (\_, a_m) & \text{sonst} \end{cases} \\
\delta_C((\_, a_l), (s, a_m), (\_, a_r)) &= \begin{cases} (s', a_m') & \text{falls } \delta_T(s, a_m) = (s', a_m', 0) \\ (\_, a_m') & \text{falls } \delta_T(s, a_m) = (s', a_m', \pm 1) \end{cases} \\
\delta_C((\_, a_l), (\_, a_m), (s, a_r)) &= \begin{cases} (s', a_m) & \text{falls } \delta_T(s, a_r) = (s', a_r', -1) \\ (\_, a_m) & \text{sonst} \end{cases}
\end{aligned}
$$

Verallgemeinerungen

Das eben war der einfache Fall *eines* Bandes und *eines* Kopfes.

Man simuliere ohne Zeitverlust

- eine TM mit *zwei* Bändern (und je einem Kopf darauf)
- eine TM mit *zwei* Köpfen (auf einem Band)
- eine TM mit 19 Bändern und verschieden vielen Köpfen darauf

Verallgemeinerungen

Das eben war der einfache Fall *eines* Bandes und *eines* Kopfes.

Man simuliere ohne Zeitverlust

- eine TM mit *zwei* Bändern (und je einem Kopf darauf)
- eine TM mit *zwei* Köpfen (auf einem Band)
- eine TM mit 19 Bändern und verschieden vielen Köpfen darauf

Ideen:

Verallgemeinerungen

Das eben war der einfache Fall *eines* Bandes und *eines* Kopfes.

Man simuliere ohne Zeitverlust

- eine TM mit *zwei* Bändern (und je einem Kopf darauf)
- eine TM mit *zwei* Köpfen (auf einem Band)
- eine TM mit 19 Bändern und verschieden vielen Köpfen darauf

Ideen:

- verschiebe nicht die Köpfe, sondern die Bänder

Verallgemeinerungen

Das eben war der einfache Fall *eines* Bandes und *eines* Kopfes.

Man simuliere ohne Zeitverlust

- eine TM mit *zwei* Bändern (und je einem Kopf darauf)
- eine TM mit *zwei* Köpfen (auf einem Band)
- eine TM mit 19 Bändern und verschieden vielen Köpfen darauf

Ideen:

- verschiebe nicht die Köpfe, sondern die Bänder
- warte nicht, bis alles verschoben, sondern …

Verallgemeinerungen

Das eben war der einfache Fall *eines* Bandes und *eines* Kopfes.

Man simuliere ohne Zeitverlust

- eine TM mit *zwei* Bändern (und je einem Kopf darauf)
- eine TM mit *zwei* Köpfen (auf einem Band)
- eine TM mit 19 Bändern und verschieden vielen Köpfen darauf

Ideen:

- verschiebe nicht die Köpfe, sondern die Bänder
- warte nicht, bis alles verschoben, sondern …
- … simuliere den nächsten Schritt, sobald alle benötigten Informationen verfügbar

Übung

Man implementiere für eine Richtung

- das «Herholen» des Bandabschnitts,
  - i. e. das «Wegschieben» einer «Lücke»
- das «Wegschieben» eines Bandabschnitts

Band- statt Kopfverschiebung bei einem Band (1)

$\delta_T(s, \mathtt{d}) =$
$(s', \mathtt{x}, -1)$

…				s					…
…	a	b	c	d	e	f	g	h	…
…									…

…				s'					…
…	a	b	c	$\overleftarrow{\square}$	e	f	g	h	…
…				$\overrightarrow{\mathtt{x}}$					…

…				s'					…
…	a	b	$\overleftarrow{\square}$	c	x	f	g	h	…
…					$\overrightarrow{\mathtt{e}}$				…

…				s'					…
…	a	$\overleftarrow{\square}$	b	c	x	e	g	h	…
…						$\overrightarrow{\mathtt{f}}$			…

…				s'					…
…	$\overleftarrow{\square}$	a	b	c	x	e	f	h	…
…							$\overrightarrow{\mathtt{g}}$		…

Band- statt Kopfverschiebung bei einem Band (2)

…				s					…
…	a	b	c	d	e	f	g	h	…
…									…

$\delta_T(s, \mathtt{d}) =$
$(s', \mathtt{x}, -1)$

…				s'					…
…	a	b	c	$\overleftarrow{\square}$	e	f	g	h	…
…				$\overrightarrow{\mathtt{x}}$					…

…				s'					…
…	a	b	$\overleftarrow{\square}$	c	x	f	g	h	…
…					$\overrightarrow{\mathtt{e}}$				…

$\delta_T(s', \mathtt{c}) =$
$(s'', \mathtt{y}, -1)$

…				s''					…
…	a	$\overleftarrow{\square}$	b	$\overleftarrow{\square}$	x	e	g	h	…
…				$\overrightarrow{\mathtt{y}}$		$\overrightarrow{\mathtt{f}}$			…

…				s''					…
…	$\overleftarrow{\square}$	a	$\overleftarrow{\square}$	b	y	e	f	h	…
…					$\overrightarrow{\mathtt{x}}$		$\overrightarrow{\mathtt{g}}$		…

Band- statt Kopfverschiebung bei einem Band (3)

…				s					…
…	a	b	c	d	e	f	g	h	…
…									…

$\delta_T(s, \mathtt{d}) =$
$(s', \mathtt{x}, -1)$

…				s'					…
…	a	b	c	$\overleftarrow{\square}$	e	f	g	h	…
…				$\overrightarrow{\mathtt{x}}$					…

…				s'					…
…	a	b	$\overleftarrow{\square}$	c	x	f	g	h	…
…					$\overrightarrow{\mathtt{e}}$				…

$\delta_T(s', \mathtt{c}) =$
$(s'', \mathtt{y}, +1)$

…				s''					…
…	a	$\overleftarrow{\square}$	b	$\overrightarrow{\square}$	x	e	g	h	…
…				$\overleftarrow{\mathtt{y}}$		$\overrightarrow{\mathtt{f}}$			…

…				s''					…
…	$\overleftarrow{\square}$	a	y	x	$\overrightarrow{\square}$	e	f	h	…
…			$\overleftarrow{\mathtt{b}}$				$\overrightarrow{\mathtt{g}}$		…

Zusammenfassung

- Es gibt Zellularautomaten, die einen klassischen Universalrechner simulieren können.

Zusammenfassung

- Es gibt Zellularautomaten, die einen klassischen Universalrechner simulieren können.
- Zellularautomaten können jede Funktion berechnen, die Turingmaschinen berechnen können.

Zusammenfassung

- Es gibt Zellularautomaten, die einen klassischen Universalrechner simulieren können.
- Zellularautomaten können jede Funktion berechnen, die Turingmaschinen berechnen können.
- *Und umgekehrt?*