

Modelle der Parallelverarbeitung

7. Baumförmige Zellularautomaten

Thomas Worsch

Institut für Theoretische Informatik
Karlsruher Institut für Technologie

Sommersemester 2020

Überblick

Baum-Zellularautomaten

Grundlagen

Erkennung formaler Sprachen

Hyperbolische Zellularautomaten

Überblick

Baum-Zellularautomaten

Grundlagen

Erkennung formaler Sprachen

Hyperbolische Zellularautomaten

Idee

- ▶ alles wie bei euklidischen ZA
- ▶ aber *Baum als Verbindungsstruktur* statt Gitter
- ▶ es gibt (kleine) Probleme zu lösen

Ursprünge und wichtige Arbeiten



K. Culik II, S. Yu

Iterative tree automata.

Theoretical Computer Science, 132:227–247, 1984.



A. Fellah, S. Yu

Iterative tree automata, alternating Turing machines, and uniform Boolean circuits: relationships and characterizations.

Proc. Symp. on Applied Computing, 1159–1166, 1992.

Formalisierungsmöglichkeiten der Verbindungsstruktur

- ▶ bekannte d -äre Bäume
 - ▶ Problem: Wurzel hat einen Nachbarn weniger als die anderen Knoten
 - ▶ Lösungen:
 - ▶ angepasstes δ für die Wurzel
 - ▶ benutze «freie» Verbindung für die Eingabe
- ▶ Cayley-Graph einer freien Gruppe, i. e. *alle* Knoten haben gleichen Grad
- ▶ im folgenden: binäre Bäume

Überblick

Baum-Zellularautomaten

Grundlagen

Erkennung formaler Sprachen

Hyperbolische Zellularautomaten

Eingaben

diverse Möglichkeiten (auch in der Literatur):

- ▶ sequenziell an der Wurzel
- ▶ parallel auf dem ersten Niveau unter der Wurzel, das genügen Knoten enthält
- ▶ entlang eines Pfades von der Wurzel weg
- ▶ bei der Wurzel beginnend Niveau für Niveau

Auswirkungen:

- ▶ vernachlässigbar falls genügend Zeit für Umsortierungen
- ▶ nicht in allen Details untersucht

Formalisierung

Betrachte als Beispiel binäre Bäume und sequenzielle Eingabe an der Wurzel: «*iterative tree automata*»

- ▶ Zustandsmenge Q
- ▶ Eingabealphabet A inkl. Symbol für «Eingabeende»
- ▶ lokale Überföhrungsfunktion für die Wurzel:
$$\delta_0 : A \times Q \times Q^2 \rightarrow Q$$
- ▶ lokale Überföhrungsfunktion für die anderen Zellen:
$$\delta : Q \times Q \times Q^2 \rightarrow Q$$

Formalisierung (1. Versuch)

Betrachte als Beispiel binäre Bäume und sequenzielle Eingabe an der Wurzel: «*iterative tree automata*»

- ▶ Zustandsmenge Q
- ▶ Eingabealphabet A inkl. Symbol für «Eingabeende»
- ▶ lokale Überföhrungsfunktion für die Wurzel:
$$\delta_0 : A \times Q \times Q^2 \rightarrow Q$$
- ▶ lokale Überföhrungsfunktion für die anderen Zellen:
$$\delta : Q \times Q \times Q^2 \rightarrow Q$$

Probleme?

Formalisierung (2. Versuch)

Betrachte als Beispiel binäre Bäume und sequenzielle Eingabe an der Wurzel: «*iterative tree automata*»

- ▶ Zustandsmenge S
- ▶ Eingabealphabet A inkl. Symbol für «Eingabeende»
- ▶ lokale Überföhrungsfunktion für die Wurzel:
 $\delta_0 : A \times Q \times Q^2 \rightarrow Q$
- ▶ lokale Überföhrungsfunktionen
 $\delta_1 : Q \times Q \times Q^2 \rightarrow Q$ und $\delta_2 : Q \times Q \times Q^2 \rightarrow Q$
 - ▶ δ_i wird vom i -ten Nachfolger eines Knotens benutzt

Erkennung formaler Sprachen

- ▶ Eingaben an der Wurzel in den ersten n Schritten, danach immer Endemarkierung
- ▶ Menge F_+ akzeptierender Endzustände
- ▶ Konfiguration sei akzeptierend, wenn die Wurzel in einem Zustand aus F_+

Ressourcenschranken

- ▶ Zeitbedarf: klar
 - ▶ aber: keine generelle Beschleunigung um konstanten Faktor!
- ▶ Platzbedarf: Anzahl benutzter Zellen
- ▶ Beachte: in t Schritten können 2^t Zellen benutzt werden
 - ▶ bei euklidischen ZA nur $\text{Pol}(t)$

SAT und QBF

- ▶ zuerst hat man nur gesehen, dass SAT in Polynomialzeit lösbar ist
- ▶ später, dass sogar QBF in Polynomialzeit lösbar ist

$$Q_1 x_1 \cdots Q_k x_k : F(x_1, \dots, x_k) \text{ mit } Q_i \in \{\forall, \exists\}$$

SAT und QBF

- ▶ zuerst hat man nur gesehen, dass SAT in Polynomialzeit lösbar ist
- ▶ später, dass sogar QBF in Polynomialzeit lösbar ist

$$Q_1x_1 \cdots Q_kx_k : F(x_1, \dots, x_k) \text{ mit } Q_i \in \{\forall, \exists\}$$

- ▶ Algorithmusidee:
 1. Verteilen der Formel von oben nach unten:
 - ▶ Entfernen und Merken des führenden Quantors
 - ▶ Ersetzung betroffener Variable überall durch 0 bzw. 1
 2. Auswerten der Formeln ohne Variablen
 3. Einsammeln der Ergebnisse von unten nach oben
 - ▶ technische Probleme?

schärfere Aussagen auf nächster Folie

Satz (Fellah/Yu (1992))

$$\text{TCA-TIME}(O(t(n))) = \text{ATM-TIME}(O(t(n)))$$

also

$$\text{TCA-TIME}(\text{Pol}(t(n))) = \text{ATM-TIME}(\text{Pol}(t(n)))$$

also

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

Satz (Fellah/Yu (1992))

Beweisidee: zeige beide Inklusionen

$$\text{ATM-TIME}(O(t(n))) \subseteq \text{TCA-TIME}(O(t(n)))$$

$$\text{TCA-TIME}(O(t(n))) \subseteq \text{ATM-TIME}(O(t(n)))$$

im folgenden Anmerkungen zum ersten Teil

Simulation ATM-TIME \rightsquigarrow TCA-TIME

- ▶ o. B. d. A.: die ATM hat nur ein Arbeitsband und in jeder Situation höchstens *zwei* Verzweigungsmöglichkeiten
- ▶ initial:
 - ▶ die Wurzel des TCA simuliert die Steuereinheit

Simulation ATM-TIME \rightsquigarrow TCA-TIME

- ▶ o. B. d. A.: die ATM hat nur ein Arbeitsband und in jeder Situation höchstens *zwei* Verzweigungsmöglichkeiten
- ▶ initial:
 - ▶ die Wurzel des TCA simuliert die Steuereinheit
 - ▶ Band am besuchten Feld «gefaltet», zwei Stacks, und
 - ▶ dieses Gebilde entlang *jedes* Pfades von der Wurzelzelle nach unten gespeichert.
 - ▶ vgl. Kap. 3, Folie 41 ff.

Simulation ATM-TIME \rightsquigarrow TCA-TIME

- ▶ o. B. d. A.: die ATM hat nur ein Arbeitsband und in jeder Situation höchstens *zwei* Verzweigungsmöglichkeiten
- ▶ initial:
 - ▶ die Wurzel des TCA simuliert die Steuereinheit
 - ▶ Band am besuchten Feld «gefaltet», zwei Stacks, und
 - ▶ dieses Gebilde entlang *jedes* Pfades von der Wurzelzelle nach unten gespeichert.
 - ▶ vgl. Kap. 3, Folie 41 ff.
- ▶ generell: wenn die ATM verzweigt,
 - ▶ werden die beiden Nachfolger der Zelle, die gerade die Steuereinheit simuliert, mit dieser Aufgabe betraut
 - ▶ und Stackpaar «eine Zelle nach unten geschoben»
 - ▶ nach und nach alles «eine Ebene nach unten»
- ▶ Akzeptieren/Ablehnen der ATM nach oben propagieren

Simulation TCA-TIME \rightsquigarrow ATM-TIME

Kernidee ähnlich wie im vorangegangenen Kapitel
Konstruktion TM-TIME \rightsquigarrow ATM-SPC

Überblick

Baum-Zellularautomaten

Hyperbolische Zellularautomaten

Hyperbolische Geometrie

Grundlagen von HCA

Überblick

Baum-Zellularautomaten

Hyperbolische Zellularautomaten

Hyperbolische Geometrie

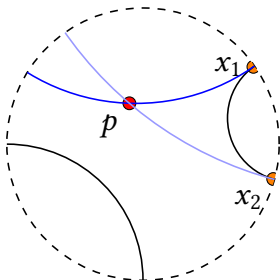
Grundlagen von HCA

Definition

- ▶ Axiome wie in der euklidischen Geometrie
- ▶ mit der *Ausnahme*:
Zu einer Geraden gibt es durch einen nicht inzidenten Punkt *mehrere* Parallelen.
- ▶ Problem:
 - ▶ Unsere Umwelt ist (jedenfalls näherungsweise?) euklidisch.
 - ▶ Visualisierung hyperbolischer Geometrie?

Kreisscheibenmodell von Poincaré

- ▶ offene Kreisscheibe mit Radius 1
 - ▶ Rand: «Unendlichkeit»
- ▶ «hyp. Punkte»: Punkte
- ▶ «hyp. Geraden»: Kreisbögen, auf Rand senkrecht
- ▶ «hyp. Parallelen»: Geraden mit «Schnitt» «im Unendlichen»



Überblick

Baum-Zellularautomaten

Hyperbolische Zellularautomaten

Hyperbolische Geometrie

Grundlagen von HCA

Zellularautomaten via Parkettierungen

- ▶ Einen \mathbb{Z}^2 -CA mit von-Neumann-Nachbarschaft mit Radius 1 kann man sich auch entstanden denken durch
 - ▶ *Parkettierung* der euklidischen Ebene
 - ▶ mit Quadraten, i. e. kongruenten regelmäßigen Vielecken als Zellen,
 - ▶ die miteinander Daten austauschen können, falls sie eine Kante gemeinsam haben.

Zellularautomaten via Parkettierungen

- ▶ Einen \mathbb{Z}^2 -CA mit von-Neumann-Nachbarschaft mit Radius 1 kann man sich auch entstanden denken durch
 - ▶ *Parkettierung* der euklidischen Ebene
 - ▶ mit Quadraten, i. e. kongruenten regelmäßigen Vielecken als Zellen,
 - ▶ die miteinander Daten austauschen können, falls sie eine Kante gemeinsam haben.
- ▶ Einen *hyperbolischen Zellularautomaten* man sich z. B. entstanden denken durch
 - ▶ *Parkettierung* der hyperbolischen Ebene
 - ▶ mit kongruenten regelmäßigen Vielecken als Zellen,
 - ▶ die miteinander Daten austauschen können, falls sie eine Kante gemeinsam haben.

Parkettierungen der hyperbolischen Ebene

- ▶ Satz von Poincaré:
Jedes Dreieck mit Winkeln $2\pi/a$, $2\pi/b$ und $2\pi/c$ mit $a, b, c \in \mathbb{N}_+$ und

$$\frac{1}{a} + \frac{1}{b} + \frac{1}{c} < \frac{1}{2}$$

erzeugt eine eindeutige Parkettierung der hyperbolischen Ebene.

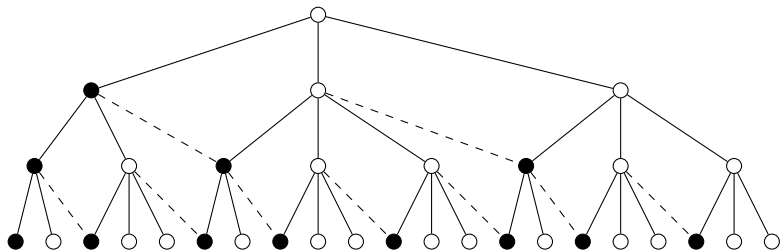
- ▶ Folge: Parkettierung mit regelmäßigen p -Ecken mit Eckwinkel $2\pi/q$, falls $1/p + 1/q < 1/2$
 - ▶ z. B. «*Pentagrid*:» 5-Ecke mit 5 rechten Winkeln

Nachbarschaftsbeziehungen im Pentagrid

Parkettierung eines «Viertels» der hyperbolischen Ebene mit Fünfecken:

- ▶ Baum mit zusätzlichen «Querverbindungen»
- ▶ 2 Sorten Baumknoten: «schwarze» und «weiße»
 - ▶ weißer Knoten hat 3 Baumnachfolger
 - ▶ schwarzer Knoten hat 2 Baumnachfolger
- ▶ der erste Nachfolger ist jeweils schwarz
- ▶ alle weiteren Nachfolger sind alle weiß
- ▶ zusätzlich eine Kante vom letzten Nachfolger eines Knotens x zum ersten Nachfolger des «rechten» Nachbarn von x

Nachbarschaftsbeziehungen im Pentagrid (2)

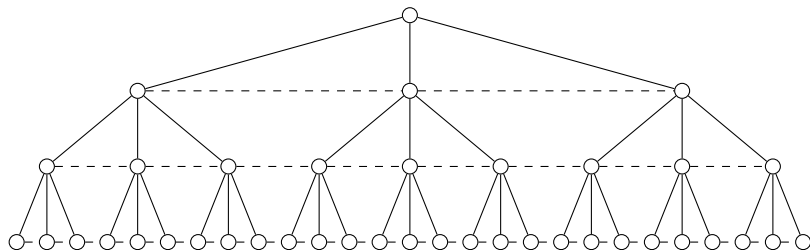


M. Margenstern

Cellular Automata in Hyperbolic Spaces

Old City Publishing, 2007.

Bäume mit horizontalen Verbindungen



Zusammenhänge zwischen verschiedenen erweiterten Bäumen

Beide Varianten erweiterter Bäume können sich auch bei verschiedenen Verzweigungsgraden in Linearzeit simulieren.



Th. Worsch

Simulations Between Cellular Automata on Trees Extended by Horizontal Edges

Fundamenta Informaticae, 58:241–260, 2003.

Zusammenhänge zwischen verschiedenen erweiterten Bäumen

Beide Varianten erweiterter Bäume können sich auch bei verschiedenen Verzweigungsgraden in Linearzeit simulieren.



Th. Worsch

Simulations Between Cellular Automata on Trees Extended by Horizontal Edges

Fundamenta Informaticae, 58:241–260, 2003.

Warum ist das nicht banal?

Zusammenfassung

- ▶ enger Zusammenhang von Baum-ZA und ATM
- ▶ hyperbolische ZA sind Baum-ZA «mit zusätzlichen Kanten»