Modelle der Parallelverarbeitung

5. Uniforme Schaltkreisfamilien

Thomas Worsch

Institut für Theoretische Informatik
Karlsruher Institut für Technologie

Sommersemester 2018

Überblick

Überblick

Schaltkreisfamilien
- Schaltkreise und Schaltkreisfamilien
- Komplexitätsmaße
- Weitere Beispiele
- Synchronizität

Uniformität

Beziehungen zu Mehrkopf-TM

Überblick

Definition Schaltkreis

$S^{(n,m)}$: gerichteter *azyklischer* Graph mit

- n *Eingabeknoten*
  - Ausgangsgrad beliebig
- *innere Knoten*
  - $\neg$ mit Eingangsgrad 1
  - $\vee$, $\wedge$ mit Eingangsgrad 2
  - Ausgangsgrad beliebig
- m *Ausgangsknoten* mit Eingangsgrad 1
- durch S *berechnete Funktion* $f_S : \mathbb{B}^n \longrightarrow \mathbb{B}^m$
  auf naheliegende Weise definiert

Definition Schaltkreisfamilie

- *Schaltkreisfamilie* $C = (C_n)_{n \in \mathbb{N}_+}$
  - wobei jedes C_n Schaltkreis mit n Eingängen und 1 Ausgang
- durch C *berechnete Funktion*

$$f_C : \mathbb{B}^+ \longrightarrow \mathbb{B} : w \mapsto f_{C_{|w|}}(w)$$

- von C *erkannte Sprache*

$$L(C) = \{w \in \mathbb{B}^+ \mid f_C(w) = 1\}$$

Beispiel Palindromerkennung

an der Tafel

Überblick

Maße für einzelne Schaltkreise

Maße für einzelne Schaltkreise

- *Größe* $\mathrm{size}(S)$:
  Gesamtzahl innerer Knoten
- Tiefe eines inneren Knotens v:
  Länge längster Pfade von Eingabeknoten zu v
- *Tiefe* $\mathrm{depth}(S)$:
  Maximum der Tiefen innerer Knoten
- Breite der Schicht d ($1 \leq d \leq \mathrm{depth}(S)$):
  Anzahl innerer Knoten mit Tiefe $\leq d$, deren Ausgaben von einem Knoten mit Tiefe $> d$ benötigt werden
- *Breite* $\mathrm{width}(S)$:
  Maximum der Schichtbreiten

Maße für Schaltkreisfamilien

- *Größe* $\mathrm{Size} : \mathbb{N}_+ \longrightarrow \mathbb{N}_+ : n \mapsto \mathrm{size}(C_n)$
- *Tiefe* $\mathrm{Depth} : \mathbb{N}_+ \longrightarrow \mathbb{N}_+ : n \mapsto \mathrm{depth}(C_n)$
- *Breite* $\mathrm{Width} : \mathbb{N}_+ \longrightarrow \mathbb{N}_+ : n \mapsto \mathrm{width}(C_n)$

Überblick

Quadrieren einer Booleschen Matrix

- Eingabe: $X \in \{0,1\}^{N \times N}$
- Ausgabe: $X^2 \in \{0,1\}^{N \times N}$
  - $(X^2)_{ij} = (x_{i1} \wedge x_{1j}) \vee (x_{i2} \wedge x_{2j}) \vee \cdots \vee (x_{iN} \wedge x_{Nj})$
- Schaltkreis:

Quadrieren einer Booleschen Matrix

- Eingabe: $X \in \{0,1\}^{N \times N}$
- Ausgabe: $X^2 \in \{0,1\}^{N \times N}$
  - $(X^2)_{ij} = (x_{i1} \wedge x_{1j}) \vee (x_{i2} \wedge x_{2j}) \vee \cdots \vee (x_{iN} \wedge x_{Nj})$
- Schaltkreis: z. B.
  - für jedes Paar (i, j)
  - ein (möglichst balancierter) Baum von $\vee$-Gattern
  - Blätter $\wedge$-Gatter

Komplexitäten dann:

Quadrieren einer Booleschen Matrix

- Eingabe: $X \in \{0,1\}^{N \times N}$
- Ausgabe: $X^2 \in \{0,1\}^{N \times N}$
  - $(X^2)_{ij} = (x_{i1} \wedge x_{1j}) \vee (x_{i2} \wedge x_{2j}) \vee \cdots \vee (x_{iN} \wedge x_{Nj})$
- Schaltkreis: z. B.
  - für jedes Paar (i, j)
  - ein (möglichst balancierter) Baum von $\vee$-Gattern
  - Blätter $\wedge$-Gatter

  Komplexitäten dann:
  - Größe: $\Theta(N^2 \cdot N)$
  - Breite: $\Theta(N^2 \cdot N)$
  - Tiefe: $\Theta(\log N)$

Reflexiv-transitive Hülle einer Relation

- Boolesche $N \times N$-Matrix X beschreibe binäre Relation
- $X^* = \bigvee_{i=0}^{N} X^i =$

Reflexiv-transitive Hülle einer Relation

- Boolesche $N \times N$-Matrix X beschreibe binäre Relation
- $X^* = \bigvee_{i=0}^{N} X^i = (E \vee X)^N =$

Reflexiv-transitive Hülle einer Relation

- Boolesche $N \times N$-Matrix X beschreibe binäre Relation
- $X^* = \bigvee_{i=0}^{N} X^i = (E \vee X)^N = (E \vee X)^{N+\cdots} =$

Reflexiv-transitive Hülle einer Relation

- Boolesche $N \times N$-Matrix X beschreibe binäre Relation
- $X^* = \bigvee_{i=0}^{N} X^i = (E \vee X)^N = (E \vee X)^{N+\cdots} =$
  $$= (\cdots((E \vee X) \overbrace{^2)^2 \cdots)^2}^{\lceil \log N \rceil \text{ mal}}$$
- Schaltkreisfamilie:

Reflexiv-transitive Hülle einer Relation

- Boolesche $N \times N$-Matrix X beschreibe binäre Relation
- $X^* = \bigvee_{i=0}^{N} X^i = (E \vee X)^N = (E \vee X)^{N+\cdots} =$
  $$= (\cdots((E \vee X) \overbrace{^2)^2 \cdots)^2}^{\lceil \log N \rceil \text{ mal}}$$
- Schaltkreisfamilie: $\log N$ Quadrierer hintereinander
  Komplexitäten:

Reflexiv-transitive Hülle einer Relation

- Boolesche $N \times N$-Matrix X beschreibe binäre Relation
- $X^* = \bigvee_{i=0}^{N} X^i = (E \vee X)^N = (E \vee X)^{N+\cdots} =$

  $$= (\cdots((E \vee X) \overbrace{{}^2)^2 \cdots)^2}^{\lceil \log N \rceil \text{ mal}}$$
- Schaltkreisfamilie: $\log N$ Quadrierer hintereinander
  Komplexitäten:
  - Größe: $\Theta(N^3 \log N)$
  - Breite: $\Theta(N^2 \cdot N)$
  - Tiefe: $\Theta((\log N)^2)$

Überblick

Definition

- Schaltkreis *synchron*, wenn jeder Knoten mit Tiefe $d > 0$ nur mit Knoten der Tiefe $d - 1$ und Eingabeknoten verbunden
- Schaltkreisfamilie *synchron*, wenn jeder Schaltkreis synchron

Lemma

Zu jeder Schaltkreisfamilie C gibt es eine synchrone Schaltkreisfamilie C' mit

- $f_{C'} = f_C$
- $\mathrm{depth}_{C'} \in \Theta(\mathrm{depth}_C)$
- $\mathrm{width}_{C'} \in \Theta(\mathrm{width}_C)$
- $\mathrm{size}_{C'} \in O(\mathrm{size}_C^2)$.

Beweisidee

Beweisidee

- ersetze jede Kante,
  die «mehr als eine Schicht überbrückt»
- durch einen Pfad der Länge 2 mit einem
  neuen $\wedge$-Gatter «in der Mitte»

Beweisidee

- ersetze jede Kante,
  die «mehr als eine Schicht überbrückt»
- durch einen Pfad der Länge 2 mit einem
  neuen $\wedge$-Gatter «in der Mitte»

- man iteriere das, bis sich nichts mehr ändert

Überblick

Problem

Welche formalen Sprachen können mit Schaltkreisfamilien erkannt werden?

Problem

Welche formalen Sprachen können mit Schaltkreisfamilien erkannt werden?

Lemma

- Für *jede(!)* formale Sprache $L \subseteq \mathbb{B}^+$ gibt es eine Schaltkreisfamilie C, die L erkennt.
- C kann dabei so gewählt werden, dass es synchron ist.

Problem

Welche formalen Sprachen können mit Schaltkreisfamilien erkannt werden?

Lemma

- Für *jede(!)* formale Sprache $L \subseteq \mathbb{B}^+$ gibt es eine Schaltkreisfamilie C, die L erkennt.
- C kann dabei so gewählt werden, dass es synchron ist.
- C kann sogar so gewählt werden, dass *zusätzlich (!)* Width_C durch eine Konstante beschränkt ist.

Beweisskizze

- schreibe Formel für $L \cap \mathbb{B}^n$ in disjunktiver Normalform
- konstante Breite: an der Tafel

Diskussion

- eben gezeigtes Lemma sehr unerfreulich

Problem:

Diskussion

- eben gezeigtes Lemma sehr unerfreulich

Problem:

- *Nichtuniformität:*
  Schaltkreise für verschiedene Eingabegrößen unabhängig voneinander, also …

Diskussion

- eben gezeigtes Lemma sehr unerfreulich

Problem:

- *Nichtuniformität:*
  Schaltkreise für verschiedene Eingabegrößen unabhängig voneinander, also …
- im allgemeinen *keine endliche Beschreibung* einer ganzen Schaltkreisfamilie möglich

Abhilfe:

- endliche Beschreibbarkeit zusätzlich fordern
- auf das «Wie?» kommt es auch an

Schaltkreiscodierung

- Knoten durchnummeriert, so dass jeder *Knoten einer Schicht d eine größere Nummer hat als alle Knoten der Schicht $d-1$*
- Knoten beschrieben durch Quadrupel $[\bar{v}, \bar{t}, \bar{v}_l, \bar{v}_r]$
  - $\bar{v}$: eigene Nummer
  - $\bar{t}$: Beschreibung des Knotentyps
  - $\bar{v}_l$: Nummer des linken Vorgängerknotens
  - $\bar{v}_r$: Nummer des rechten Vorgängerknotens, falls $\bar{t} = \vee$ oder $\bar{t} = \wedge$, sonst $\bar{v}_r = \bar{v}_l$
  - bei Ein- und Ausgabeknoten sei $\bar{v}_r = \bar{v}_l$ die Nummer des relevanten Bits
- $\bar{S}$ sei die aufsteigend (nach $\bar{v}$) sortierte Liste der Knotenbeschreibungen

Schaltkreiskonstruktoren

- benutze $\mathbb{EA}$-TM
  - Eingabe: 1^n
  - Ausgabe: Codierung $\bar{C_n}$ von C_n
- Welche Schaltkreiskonstruktoren erlaubt (bzw. verbietet) man warum?

Schaltkreiskonstruktoren

- benutze $\mathbb{EA}$-TM
  - Eingabe: 1^n
  - Ausgabe: Codierung $\bar{C_n}$ von C_n
- Welche Schaltkreiskonstruktoren erlaubt (bzw. verbietet) man warum?
  - jedenfalls verboten: beliebiger Aufwand

Schaltkreiskonstruktoren

- benutze $\mathbb{EA}$-TM
  - Eingabe: 1^n
  - Ausgabe: Codierung $\bar{C_n}$ von C_n
- Welche Schaltkreiskonstruktoren erlaubt (bzw. verbietet) man warum?
  - jedenfalls verboten: beliebiger Aufwand
    - $\leadsto$ Konstruktor macht «die ganze Arbeit»
    - $\leadsto$ Schaltkreise trivial, in DNF

Schaltkreiskonstruktoren

- benutze $\mathbb{EA}$-TM
  - Eingabe: 1^n
  - Ausgabe: Codierung $\bar{C_n}$ von C_n
- Welche Schaltkreiskonstruktoren erlaubt (bzw. verbietet) man warum?
  - jedenfalls verboten: beliebiger Aufwand
    - $\rightsquigarrow$ Konstruktor macht «die ganze Arbeit»
    - $\rightsquigarrow$ Schaltkreise trivial, in DNF
  - sinnvoll: *Platzbedarf* beschränkt durch $O(\log \mathrm{Size}(C))$

Schaltkreiskonstruktoren

- benutze $\mathbb{EA}$-TM
  - Eingabe: 1^n
  - Ausgabe: Codierung $\bar{C_n}$ von C_n
- Welche Schaltkreiskonstruktoren erlaubt (bzw. verbietet) man warum?
  - jedenfalls verboten: beliebiger Aufwand
    - $\leadsto$ Konstruktor macht «die ganze Arbeit»
    - $\leadsto$ Schaltkreise trivial, in DNF
  - sinnvoll: *Platzbedarf* beschränkt durch $O(\log \mathrm{Size}(C))$
    - die Ergebnisse werden «schön»
    - Motiviation duch genaueres Hinsehen

Definition

Eine *Schaltkreisfamilie* $C = \left(C^{(n)}\right)_{n \in \mathbb{N}_+}$ heiße genau dann *uniform*, wenn sie einen Schaltkreiskonstruktur besitzt.

Definition

- Wir schreiben

$$\mathrm{UC\text{-}W{\scriptsize ID}}(w(n))\text{-}\mathrm{S{\scriptsize IZE}}(z(n))\text{-}\mathrm{D{\scriptsize EP}}(d(n))$$

  für die Familie aller formalen Sprachen L, für die es eine uniforme Schaltkreisfamilie C gibt mit
  - $L(C) = L$
  - $\mathrm{Width}_C(n) \leq w(n)$
  - $\mathrm{Size}_C(n) \leq z(n)$
  - $\mathrm{Depth}_C(n) \leq d(n)$
  - $\mathrm{Depth}_C \in \Omega(\log \mathrm{Size}_C)$ (banal) und
    $\mathrm{Width}_C \in \Omega(\log \mathrm{Size}_C)$ (nicht banal)

… und nun …

Beziehung(en) zwischen TM und uniformen Schaltkreisfamilien …

Erwartungen?

Überblick

Überblick

Simulation UC-Wid $\rightsquigarrow$ TM-Spc

Satz

Für Funktionen w und z mit $w \geq \log z$ gilt:

$$\text{UC-Wid}(w)\text{-Size}(z) \subseteq \mathbb{E}\text{-TM-Spc}(\Theta(w^2))\text{-Time}(\mathrm{Pol}(z))$$

UC-WID $\rightsquigarrow$ TM-SPC: Konstruktion

- Es sei C eine entsprechende Schaltkreisfamilie.
- Für ihren Schaltkreiskonstruktor K gilt:
  - $\text{Space}_K \in O(\log z) \subseteq O(w)$
  - $\text{Time}_K \in \text{Pol}(z)$
- Eine TM T mit Eingabe w der Länge n, die C «simuliert», benötigt ab und zu die Beschreibung eines Gatters von C_n mit einer bestimmten Nummer m.
  - Dazu simuliert T den Konstruktor K und verwirft alle Ausgaben außer der Beschreibung des m-ten Gatters.
  - Platz- und Zeitbedarf sind $O(w)$ und $\text{Pol}(z)$ respektive.

UC-Wid $\rightsquigarrow$ TM-Spc: Konstruktion

Konstruktion der TM:

- berechne die Ausgaben der Gatter des Schaltkreises
  - «Schicht für Schicht»
  - und in jeder Schicht jeweils «Gatter für Gatter»
- speichere berechnete und noch benötigte Ausgabebits
  - in Liste W von Paaren $(\langle Bit \rangle, \langle Gatternr \rangle)$
  - Platzbedarf $\mathrm{width}(C_n) \log(\mathrm{size}(C_n)) \in O(\mathrm{width}(C_n)^2)$
- bei der Bearbeitung von $[\bar{v}, \bar{t}, \bar{v}_l, \bar{v}_r]$:
  - suche Ausgabebits von $\bar{v}_l$ und $\bar{v}_r$ in W
  - berechne Ausgabe von $\bar{v}$
  - füge Ergebnis an W an
  - prüfe, ob $\bar{v}_l$ und $\bar{v}_r$ aus W gestrichen werden können

UC-Wid $\rightsquigarrow$ TM-Spc: kleine Verbesserung

- man muss in W nicht zu jedem Bit die Gatternummer speichern
- Platz w genügt
  - Liste der noch benötigten Ausgabebits,
    nach Gatternummern sortiert

UC-Wid $\rightsquigarrow$ TM-Spc: kleine Verbesserung

- man muss in W nicht zu jedem Bit die Gatternummer speichern
- Platz w genügt
  - Liste der noch benötigten Ausgabebits,
    nach Gatternummern sortiert
- Auffinden der Ausgabe von Gatter x
  - prüfe für jedes $i < x$, ob Ausgabe von Gatter i
  - nach aktuell betrachtem Gatter y noch benötigt wird
  - bestimme Anzahl z nicht mehr benötigter Gatter
    - d. h. schon gestrichener Bits
  - gesuchtes Bit an Stelle $x - z$ in Bitliste

Simulation TM-Spc $\rightsquigarrow$ UC-Wid

Satz

Es seien die Funktionen s und t mit Platzbedarf $\log(s)$ berechenbar, $s \in \Omega(\log(t))$ und $s \geq \log$. Dann gilt:

$$\mathbb{E}\text{-TM-}\textsc{Spc}(s)\text{-}\textsc{Time}(t) \subseteq \text{UC-}\textsc{Wid}(\Theta(s))\text{-}\textsc{Size}(\text{Pol}(t))$$

TM-Spc $\rightsquigarrow$ UC-Wid: Konstruktion

- «o. B. d. A. »: betrachte nur $\mathbb{W}_1$-TM
- Beweisskizze an der Tafel (siehe auch: Buch)

Folgerungen

Korollar

Wenn die Funktionen w und z gewisse Voraussetzungen erfüllen, dann gilt:

$$\mathbb{E}\text{-TM-}\textsc{Spc}(\mathrm{Pol}(w))\text{-}\textsc{Time}(\mathrm{Pol}(z)) = \text{UC-}\textsc{Wid}(\mathrm{Pol}(w))\text{-}\textsc{Size}(\mathrm{Pol}(z))$$

$$\mathbb{E}\text{-TM-}\textsc{Spc}(\mathrm{Pol}(w)) = \text{UC-}\textsc{Wid}(\mathrm{Pol}(w))$$

$$\mathbb{E}\text{-TM-}\textsc{Time}(\mathrm{Pol}(z)) = \text{UC-}\textsc{Size}(\mathrm{Pol}(z))$$

Überblick

Definition

partielle Konfiguration einer $\mathbb{E}$-TM:
man «vergesse» die Beschriftung des Eingabebandes

- Beispiel: allen Anfangskonfigurationen entspricht dieselbe partielle Anfangskonfiguration
- die Kopfposition auf dem Eingabeband bleibt relevant

Lemma

Jede $\mathbb{E}$-TM kann umgebaut werden zu einer, die

- die gleiche Sprache akzeptiert,
- größenordnungsmäßig die gleiche Raumkomplexität hat,
- größenordnungsmäßig die gleiche Zeitkomplexität hat und
- nur eine partielle akzeptierende Endkonfiguration besitzt.

Beweisidee

- am Ende der Berechnung die Arbeitsbänder löschen und
- alle Köpfe auf ihre Anfangspositionen fahren:
  - auf den Arbeitsbändern: zu Beginn markieren
  - auf dem Eingabeband: leicht zu finden

Simulation TM-Spc $\rightsquigarrow$ UC-Dep

Satz

Es sei $s \geq \log$ und von einer $\mathbb{EA}$-TM mit Platzbedarf s berechenbar. Dann gilt:

$$\begin{aligned} \mathbb{E}\text{-TM-S{\scriptsize PC}}(s) &\subseteq \text{UC-S{\scriptsize IZE}}(\operatorname{Exp}(s))\text{-D{\scriptsize EP}}(\Theta(s^2)) \\ &\subseteq \text{UC-D{\scriptsize EP}}(\Theta(s^2)) \end{aligned}$$

Simulation TM-Spc $\rightsquigarrow$ UC-Dep

Beweisidee:

- sei X die Adjazenzmatrix des Graphen mit
  - Knoten: partielle Konfigurationen der TM
  - Kanten: Existenz nur abhängig vom jeweils gelesenen Eingabesymbol
- berechne X^*
- betrachte X^*_{ae}
  - a Nummer *der* partiellen Anfangskonfiguration
  - e Nummer *der* partiellen Endkonfiguration
- Berechung von X: … [siehe Tafel] …

Simulation UC-Dep $\rightsquigarrow$ TM-Spc

Satz

Für $s \geq \log$ gilt:

$$\text{UC-Dep}(s) \subseteq \mathbb{E}\text{-TM-Spc}(\Theta(s^2))$$

Beweisskizze

- Berechnung der Ausgaben aller Gatter durch
- Depth-first-Durchlauf des Schaltkreises
  von seinem Ausgang «rückwärts»

Folgerungen (1)

Satz

Für gewisse Funktionen $s \geq \log$ gilt:

$$\mathbb{E}\text{-TM-}\textsc{Spc}(\mathrm{Pol}(s)) = \text{UC-}\textsc{Dep}(\mathrm{Pol}(s))$$

Folgerungen (2)

Satz
Für gewisse Funktionen $s \geq \log$ gilt:

$$\text{UC-}\textsc{Wid}(\mathrm{Pol}(s)) = \text{UC-}\textsc{Dep}(\mathrm{Pol}(s))$$

Überblick

Schaltkreisfamilien

Uniformität

Beziehungen zu Mehrkopf-TM

- UC-WID versus TM-SPC
- UC-DEP versus TM-SPC
- Ausblick

Ein weiteres Komplexitätsmaß für TM

- Bei uniformen Schaltkreisfamilien haben wir drei Komplexitätsmaße betrachtet,
- bei Turingmaschinen bisher nur zwei;
- also …

Umkehrkomplexität von TM

- Kopf führt eine *Umkehr* (engl. *reversal*) aus,
  wenn er sich in die entgegengesetzte Richtung bewegt
  wie bei der vorangegangenen Bewegung.
- definiere:

$$\begin{aligned} \mathrm{rev} : A^+ &\longrightarrow \mathbb{N}_+ \\ w &\longmapsto |\{t \le \mathrm{time}(w) \mid \text{in Schritt } t \text{ passiert eine Umkehr}\}| \\ \mathrm{Rev} : \mathbb{N}_+ &\longrightarrow \mathbb{N}_+ \\ n &\longmapsto \max\{\mathrm{rev}(w) \mid w \in A^n\} \end{aligned}$$

Zusammenhang zwischen Turingmaschinen und Schaltkreisfamilien

ohne Beweis:

Satz

Für gewisse Funktionen s, t und r gilt:

$$\begin{aligned}\mathbb{E}\text{-TM-}\textsc{Spc}(\mathrm{Pol}(s))\text{-}\textsc{Time}(\mathrm{Pol}(t))\text{-}\textsc{Rev}(\mathrm{Pol}(r)) \\ = \text{UC-}\textsc{Wid}(\mathrm{Pol}(s))\text{-}\textsc{Size}(\mathrm{Pol}(t))\text{-}\textsc{Dep}(\mathrm{Pol}(r))\end{aligned}$$

Zusammenfassung

- (parallele) Zeit bei uniformen Schaltkreisfamilien
  polynomiell verknüpft mit
- (sequenziellem) Platzbedarf bei Turingmaschinen
  - analog wie bei PRAM
- Erklärungsmöglichkeit:
  - polynomieller Zusammenhang UC-WID $\leftrightsquigarrow$ TM-SPC
  - polynomieller Zusammenhang UC-DEP $\leftrightsquigarrow$ UC-WID