Unconventional Finite Automata and Algorithms

Kaspars Balodis

Scientific advisor:
Dr. habil. math., Prof. Rūsiņš Mārtiņš Freivalds

EIROPAS SOCIĀLAIS FONDS
EIROPAS SAVIENĪBA
LATVIJAS UNIVERSITĀTE
ANNO 1919
IEGULDĪJUMS TAVĀ NĀKOTNĒ

March 31, 2016

What is this thesis about?

Unconventional models of computation

- Probabilistic and Alternating Two-Way Automata
- Ultrametric Automata and Query Algorithms
- Frequency Automata and Algorithms

Introduction

Unconventionality

- Typically computer is viewed as a deterministic machine
- Other models – nondeterminism, alternation
- Inspired by nature (physics and biology):
  - Randomized and quantum computing
  - Cellular automata
  - DNA computing
  - Neural networks
- Mathematical formal systems:
  - Lambda calculus
  - Markov algorithms
  - Wang tiles

Introduction

- What should be considered unconventional?

Introduction

- What should be considered unconventional?

Arora, Barak, "Computational Complexity: A Modern Approach"

"One should note that, unlike standard Turing machines, nondeterministic Turing machines are not intended to model any physically realizable computation device."

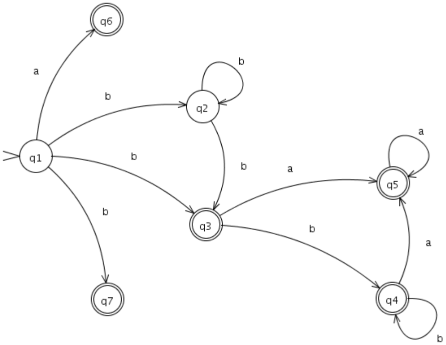
Why research unconventional models?

- Different degrees of unconventionality are possible
- Sometimes results are not readily applicable at the time of discovery
  - Ancient Greek mathematicians, prime numbers, cryptography
- Sometimes results in one field have unexpected applications in other fields.
  - Methods of quantum computing to prove classical results
- Therefore it is not impossible that unconventional models of computation, however unimaginable as physical devices, can later turn out useful in unexpected ways.

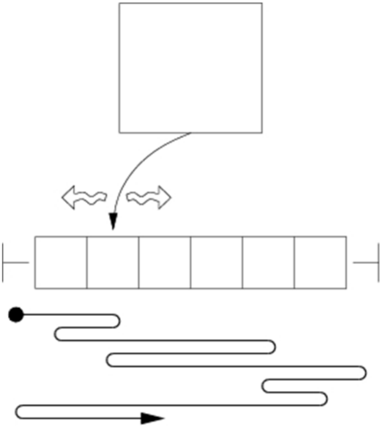
Outline

Two-Way Finite Automata

Finite automata



Two-Way Finite Automata



Classical automata theory

- It is well known that one-way deterministic finite automata (1DFAs) can recognize regular languages
- The same holds for nondeterministic (1NFA) and alternating (1AFA) finite automata
- Even for two-way versions – 2DFAs, 2NFAs and 2AFAs
- Why consider anything else than 1DFA?

Classical automata theory

- It is well known that one-way deterministic finite automata (1DFAs) can recognize regular languages
- The same holds for nondeterministic (1NFA) and alternating (1AFA) finite automata
- Even for two-way versions – 2DFAs, 2NFAs and 2AFAs
- Why consider anything else than 1DFA?

SIZE COMPLEXITY!

Two-Way Finite Automata Complexity Theory

- Consider:
  - Two-way automata
  - Deterministic, nondeterministic, alternating, probabilistic, ...
  - Families of languages
  - Complexity measure – the number of states
- Suddenly we get a fully-fledged complexity theory with complexity classes, reductions, complete problems, similar to Turing machine time (or space) complexity theory

Two-Way Finite Automata Complexity Theory

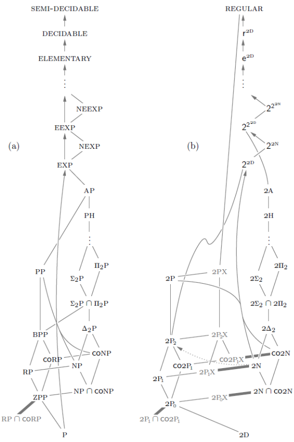


Image by Christos A. Kapoutsis

Two-Way Finite Automata Complexity Theory

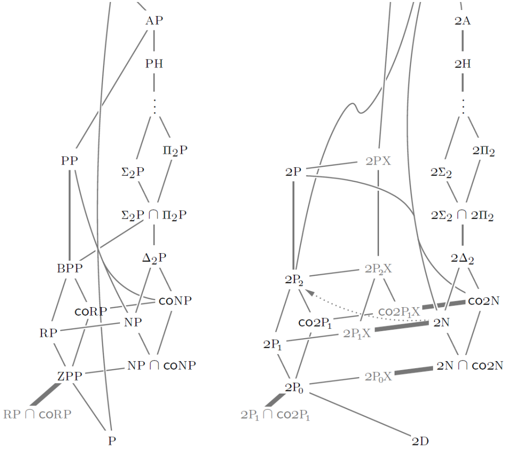


Image by Christos A. Kapoutsis

Two-Way Finite Automata Complexity Theory

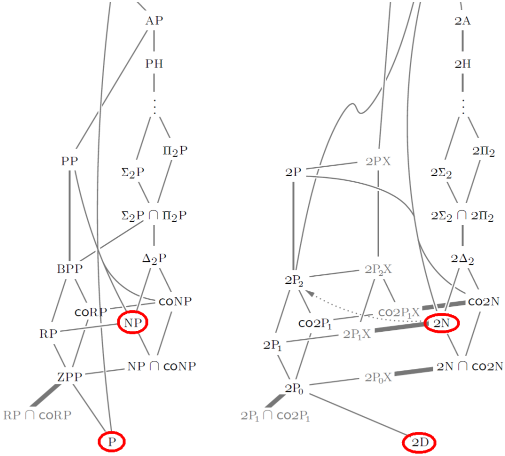


Image by Christos A. Kapoutsis

Two-Way Finite Automata Complexity Theory

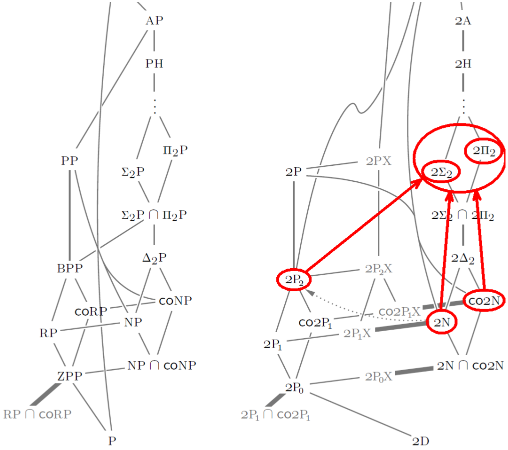


Image by Christos A. Kapoutsis

Alternating Automata

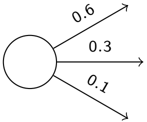
- Existential (∃) and universal (∀) states

- ∃ branching – there **must exist** a computation path that leads to an accepting state
- ∀ branching – **all** computation paths must lead to an accepting state

- Nondeterministic automaton – alternating automaton with only existential states

Probabilistic Automata

- Probabilistic transitions



- **Bounded error**:
  $w \in L \Rightarrow Pr[w \text{ is accepted}] \geq {}^2/_3$
  $w \notin L \Rightarrow Pr[w \text{ is accepted}] \leq {}^1/_3$
- **Fast** – the expected runtime is polynomial in the length of the input word

Result

Theorem

There exists a family of languages that can be accepted by a family of linear-sized one-way alternating automata (using only 1 alternation), but cannot be accepted by any family of polynomial-size bounded-error fast two-way probabilistic automata (or polynomial-size two-way nondeterministic automata).

Corollary

Neither $1\Sigma_2$ *nor* $1\Pi_2$ *is contained in* $2P_2 \cup 2N \cup co\text{-}2N$.

Two-Way Finite Automata Complexity Theory

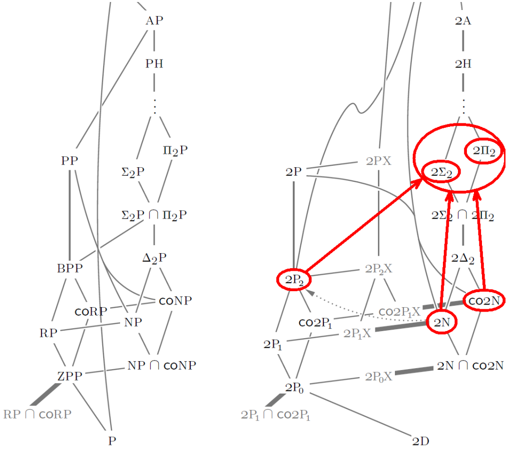


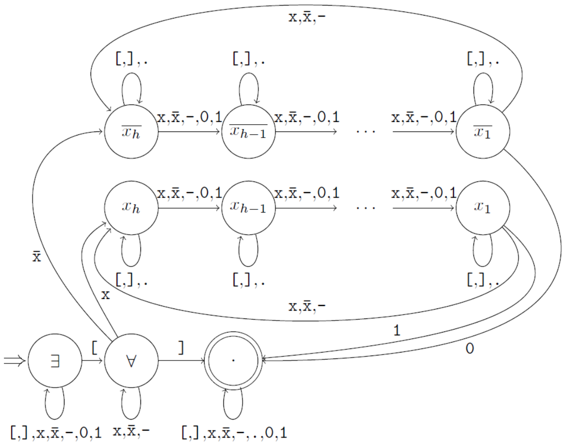
Image by Christos A. Kapoutsis

Result

Language

- Encoding of a DNF of a Boolean function f
  e.g., $f = (x_1 \wedge \overline{x_2} \wedge x_4 \wedge x_5) \vee (x_1 \wedge x_5) \vee (\overline{x_2} \wedge x_4 \wedge \overline{x_5})$
- Values of the variables: $x_1, x_2, \ldots, x_h$
- Accept, if $f(x_1, \ldots, x_h) = 1$
  Reject, if $f(x_1, \ldots, x_h) = 0$
- [x$\bar{\text{x}}$-xx][x---x][-$\bar{\text{x}}$-x$\bar{\text{x}}$].10010 $\in L$
  [x$\bar{\text{x}}$-xx][x---x][-$\bar{\text{x}}$-x$\bar{\text{x}}$].01010 $\notin L$
- $L = \{f.x \mid f(x) = 1\}$

Alternating automaton



Counting argument

- If a 2NFA or 2PFA recognizes $L = \{f.x \mid f(x) = 1\}$, it can be modified to recognize $L_f = \{x \mid f(x) = 1\}$ for any fixed f
- There are 2^{2^h} different Boolean functions on h variables
- There are only $2^{poly(h)}$ different 2NFAs with $poly(h)$ states
- However, there is an infinite number of even 2-state 2PFAs
- If the 2PFA is bounded-error and fast, it can be shown that there are only $2^{poly(h)}$ essentially different 2PFAs with $poly(h)$ states

Ultrametric Finite Automata

p-adic numbers

Let p be an arbitrary prime number.
A p-adic digit is a natural number between 0 and $p-1$ (inclusive).
A p-adic integer is a sequence $(a_i)_{i \in \mathbb{N}}$ of p-adic digits.

$$\cdots a_i \cdots a_2 a_1 a_0$$

$$\begin{array}{r} \cdots 333334 \\ \cdots 333334 \\ \hline \hline \cdots 000001 \\ \hline \end{array}$$

p-adic norm

Definition

The p-norm of a rational number $\alpha = \pm 2^{\alpha_2} 3^{\alpha_3} 5^{\alpha_5} 7^{\alpha_7} \cdots$ where $\alpha_i \in \mathbb{Z}$ is:

$$\|\alpha\|_p = \begin{cases} p^{-\alpha_p}, & \text{if } \alpha \neq 0 \\ 0, & \text{if } \alpha = 0. \end{cases}$$

Ultrametric Automata

Definition

A p-ultrametric finite automaton (U_pFA) is a sextuple $\langle Q, \Sigma, q_0, \delta, Q_A, Q_R \rangle$ where

Q is a finite set – the set of states,

Σ is a finite set – input alphabet,

$q_0 : Q \to \mathbb{Q}_p$ is the initial amplitude distribution,

$\delta : \Sigma \times Q \times Q \to \mathbb{Q}_p$ is the transition function,

$Q_A, Q_R \subseteq Q$ are the sets of accepting and rejecting states, respectively.

Ultrametric Automata

Definition

$$s_\varepsilon = q_0$$

$$s_{w_1 \dots w_i}(q) = \sum_{q' \in Q} s_{w_1 \dots w_{i-1}}(q') \cdot \delta\left(w_i, q', q\right)$$

$$\sum_{q \in Q_A} \|s_w(q)\|_p > \sum_{q \in Q_R} \|s_w(q)\|_p \Leftrightarrow w \text{ is accepted}$$

Regulated Ultrametric Automata

Definition

If for U_pFA $M = \langle Q, \Sigma, s_0, \delta, Q_A, Q_R \rangle$ all transition amplitudes in δ are p-adic integers and there exist constants $d_1, d_2 \in \mathbb{Z}$ such that on any word $w \in \Sigma^*$ in any state $q \in Q$ either the amplitude $s_w(q)$ in state q after reading word w is equal to 0 or $p^{-d_2} \leq \|s_w(q)\|_p \leq p^{-d_1}$ then we call the automaton regulated (or more specifically – (d_1, d_2)-regulated).

Theorem

If a k-state (d_1, d_2)-regulated U_pFA $M = \langle Q, \Sigma, s_0, \delta, Q_A, Q_R \rangle$ recognizes a language L, then there exists a DFA with $2^{k(d_2 - d_1 + 1)\log_2 p}$ states recognizing L.

Ultrametric Automata

Ultrametric automata can have fewer states than deterministic!

Theorem

For every k, m there is a language $L_{k,m}$ such that:

- *Every deterministic finite automaton recognizing $L_{k,m}$ needs at least k^m states.*
- *For every prime p there is a regulated U_pFA recognizing $L_{k,m}$ with $(k+1) \cdot m - 1$ states.*
- *For every prime $p > m$ there is a U_pFA recognizing $L_{p,m}$ with $m + 1$ states.*

Ultrametric Threshold Automata

Definition

A finite p-ultrametric threshold automaton (U_pFTA) is a sextuple $\langle Q, \Sigma, s_0, \delta, F, \Lambda \rangle$, where

Q is a finite set – the set of states,

Σ is a finite set, ($\$ \notin \Sigma$) – the input alphabet,

$q_0 : Q \to \mathbb{Q}_p$ is the initial amplitude distribution,

$\delta : (\Sigma \cup \{\$\}) \times Q \times Q \to \mathbb{Q}_p$ is the transition function,

$F \subseteq Q$ is the set of final states,

$\Lambda = (\lambda, \diamond)$ is the acceptance condition where $\lambda \in \mathbb{R}$ is the acceptance threshold and $\diamond \in \{\leq, \geq\}$.

Ultrametric Threshold Automata

There is no need for endmarker.

Theorem

For every U_pFTA $M = (Q, \Sigma, q_0, \delta, F, \Lambda)$ there exists a U_pFTA $M' = (Q', \Sigma, q_0', \delta', F', \Lambda)$ with $|Q| + |F|$ states such that for every word w: $\sum_{q \in F} \|s_{w\$}(q)\|_p = \sum_{q \in F'} \|s'_w(q)\|_p$, where s and s' are the amplitude distributions of U_pFTAs M and M', respectively.

Ultrametric Threshold Automata

Threshold can be simulated with accepting and rejecting states!

Theorem

If a language L is recognized (without endmarker) by a U_pFTA $M = (Q, \Sigma, q_0, \delta, F, (\lambda, \diamond))$ such that there exists $\lambda' = \sum_{i=a}^{b} l_i \cdot p^i$ such that $\forall w \in \Sigma^* \ \sum_{q \in F} \|s_w(q)\|_p \diamond \lambda \Leftrightarrow \sum_{q \in F} \|s_w(q)\|_p \tilde{\diamond} \lambda'$ (where $\tilde{\leq}$ is $<$ and $\tilde{\geq}$ is $>$) then there exists a U_pFA M' with $|Q| + \sum_{i=a}^{b} l_i$ states which recognizes L.

Multihead Ultrametric Automata

Definition

A finite k-head two-way p-ultrametric automaton ($2u_pfa(k)$) is a septuple $\langle S, \Sigma, k, s_0, \delta, Q_A, Q_R \rangle$ where

- S is a finite set of states,
- Σ is a finite set ($\triangleright, \triangleleft \notin \Sigma$) – the input alphabet ($\triangleright$ and $\triangleleft$ are the left and right endmarkers, respectively),
- $k \geq 1$ is the number of heads,
- $s_0 : S \to \mathbb{Q}_p$ is the initial distribution of amplitudes,
- $\delta : S \times (\Sigma \cup \{\triangleright, \triangleleft\})^k \times S \times \{-1, 0, 1\}^k \to \mathbb{Q}_p$ is the transition function, and
- $Q_A, Q_R \subseteq S$ are the sets of accepting and rejecting states, respectively.

Multihead Ultrametric Automata

Theorem

For every $k \geq 1 \in \mathbb{N}$, there exists a language L_k such that:

- *for every prime p there exists a $1u_pfa(1)$ that recognizes L_k,*
- *L_k cannot be recognized by any $1nfa(k)$.*

Multihead Ultrametric Automata

Theorem

$$\mathcal{L}(2U_pFA(k)) \subsetneq \mathcal{L}(2U_pFA(k+1)).$$

Counting With Automata

Counting Problem

Definition

$C_n = \{1^n\}$.

Probabilistic Automata

Theorem

For each n, there exists a 1PFA that recognizes C_n with 3 states with an isolated cutpoint.

Theorem

If $n > 1$ then any 1PFA that recognizes C_n has at least 3 states.

Theorem

There exists a constant c such that for every $\varepsilon > 0$ and for each n there exists a 2PFA that recognizes C_n with c states with probability $1 - \varepsilon$.

Ultrametric Automata

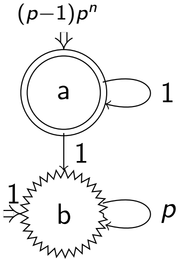
Theorem

For each n and each prime p there exists a regulated U_pFA that recognizes C_n with 2 states.

Theorem

If $n > 0$ then any U_pFA that recognizes C_n has at least 2 states.

Ultrametric Automata



w	$s_w(a)$	$\|s_w(a)\|_p$	$s_w(b)$	$\|s_w(b)\|_p$
ε	...0600000	7^{-5}	...000000000000001	1
1	...0600000	7^{-5}	...000000000600010	7^{-1}
11	...0600000	7^{-5}	...000000006600100	7^{-2}
111	...0600000	7^{-5}	...000000066601000	7^{-3}
1111	...0600000	7^{-5}	...000000666610000	7^{-4}
11111	...0600000	7^{-5}	...000010000000000	7^{-10}
111111	...0600000	7^{-5}	...000100000600000	7^{-5}
1111111	...0600000	7^{-5}	...001000006600000	7^{-5}
11111111	...0600000	7^{-5}	...010000066600000	7^{-5}

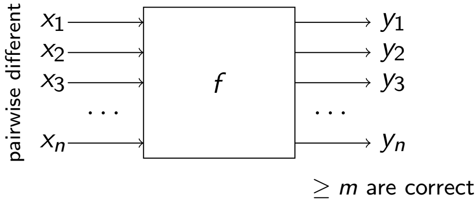
The amplitudes and norms for states a and b on different words w with $p = 7$ and $n = 5$.

Two-Way Frequency Automata

Frequency Computation

Definition (Rose, 1960)

A set A is (m, n)-computable iff there is a total recursive function f which assigns to all distinct inputs $x_1, x_2, \ldots, x_n$ a binary vector $(y_1, y_2, \ldots, y_n)$ such that at least m of the equations $\chi_A(x_1) = y_1, \chi_A(x_2) = y_2, \ldots, \chi_A(x_n) = y_n$ hold.



Frequency Computation

Theorem (Trakhtenbrot, 1964)

If $\frac{m}{n} > \frac{1}{2}$ then every (m, n)-computable set is recursive.

If $\frac{m}{n} \leq \frac{1}{2}$ then there is a continuum of (m, n)-computable sets.

Theorem (Kinber, 1976; Austinat et al., 2005)

For one-way finite automata:

If $\frac{m}{n} > \frac{1}{2}$ then every (m, n)-computable set is regular.

If $\frac{m}{n} \leq \frac{1}{2}$ then there is a continuum of (m, n)-computable sets.

Two-Way Frequency Automata

Definition

For natural numbers m, n $(1 \leq m \leq n)$ a two-way (m, n)-frequency finite automaton ((m, n)-*2FFA*) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where

- Q is the finite set of states,
- Σ is the input alphabet,
- $\delta : Q \times (\Sigma \cup \{\vdash, \dashv\})^n \to Q \times \{L, N, R\}^n$ is the transition function, where $\vdash, \dashv \notin \Sigma$ are the left and right endmarkers, respectively,
- $q_0 \in Q$ is the starting state, and
- $F : Q \to \{0, 1\}^n$ is the acceptance function.

Two-Way Frequency Automata

Definition

We say that a language $L \subseteq \Sigma^*$ is recognized by an (m, n)-*2FFA* $\mathcal{A}$ if for every n distinct input words $x_1, \ldots, x_n \in \Sigma^*$ the automaton $\mathcal{A}$ when started on $x_1, \ldots, x_n$ gives an output $(y_1, \ldots, y_n) \in \{0, 1\}^n$ such that at least m of the following hold:

$$\begin{aligned} y_1 = 1 &\Leftrightarrow x_1 \in L \\ y_2 = 1 &\Leftrightarrow x_2 \in L \\ &\vdots \\ y_n = 1 &\Leftrightarrow x_n \in L \end{aligned}$$

Two-Way Frequency Automata

Theorem

$\mathcal{L}((n,n)\text{-}2FFA) = \mathcal{L}((1,1)\text{-}2FFA) = REG$

Two-Way Frequency Automata

$2BCA(k)$ – 2DFA with k linearly bounded counters

Theorem

For $n > k$:

$$\mathcal{L}((n-k, n)\text{-}2FFA) \supseteq \mathcal{L}(2BCA(k))$$

Two-Way Frequency Automata

Corollary

For any $n > 1$ the languages

$$\left\{1^{2^m} \mid m \geq 0\right\},$$
$$\left\{1^{2^{2^m}} \mid m \geq 0\right\},$$
$$\left\{1^{42^{m^2}} \mid m \geq 0\right\},$$
$$\left\{1^{11^p} \mid p \text{ is a prime}\right\},$$
$$\left\{0^m 1^{m^2} \mid m \geq 0\right\},$$
$$\left\{0^m 1^{2^m} \mid m \geq 0\right\}$$

can be recognized by an $(n-1, n)$-2FFA.

Two-Way Frequency Automata

Theorem

$$\forall L \in LOGSPACE \;\; \exists k \; \forall n > k \;\; L \in \mathcal{L}((n-k, n)\text{-}2FFA)$$

Two-Way Frequency Automata

Can $(n-k,n)$-*2FFA* do something more than an automaton with k linearly-bounded counters?

Theorem

$$(\mathcal{L}(2BCA(k+1)) \setminus \mathcal{L}(2BCA(k))) \cap \mathcal{L}((1,k+1)\text{-}2FFA) \neq \emptyset \Rightarrow$$
$$\forall n > k \quad \mathcal{L}((n-k,n)\text{-}2FFA) \supsetneq \mathcal{L}(2BCA(k))$$

Ultrametric Query Algorithms

Ultrametric Query Algorithms

A p-ultrametric algorithm is described by a directed acyclic graph (DAG). There is exactly one vertex (root) which has no incoming edges. The nodes with no outgoing edges are leafs and they are the final (accepting) states of the algorithm.

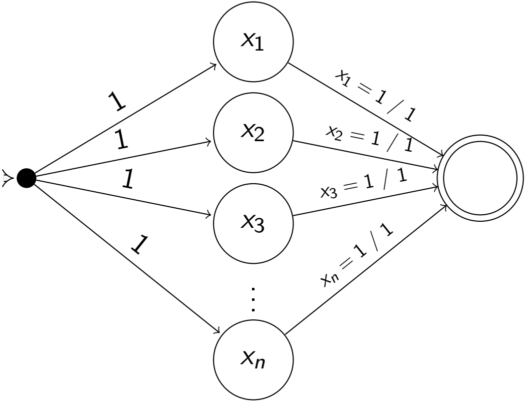
Definition

We say that a p-ultrametric query algorithm is **one-endpoint** if it has exactly one accepting state.

Definition

We say that a p-ultrametric query algorithm is **exact** if for every input the sum of norms of the final amplitudes is either 0 or 1.

Ultrametric Query Algorithms



2-ultrametric query algorithm for XOR_n

Ultrametric Query Algorithms

Theorem

$U^1_{p,E}(f) \leq deg(f) \leq 2Q_E(f)$

Definition

Let us denote by $deg_2(f)$ the binary polynomial degree of function f, i.e., the minimal degree of a polynomial $p(x)$ such that $p(x) \equiv f(x) \pmod 2$.

Theorem

$U^1_2(f) \leq deg_2(f)$

Ultrametric Query Algorithms

Theorem

For every prime $p > n$ there exists a one-endpoint exact p-ultrametric query algorithm with complexity 1 for OR_n.

Definition

$$NDIV_{n,p}(x) = \begin{cases} 0, & \text{if } \overline{x_n x_{n-1} \ldots x_1} \text{ is divisible by } p \\ 1, & \text{otherwise} \end{cases}$$

Theorem

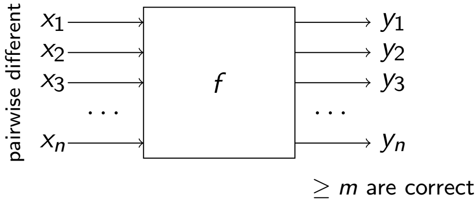
For any n and any prime p there exists a one-endpoint p-ultrametric query algorithm with complexity 1 that computes $NDIV_{n,p}$.

Structured Frequency Algorithms

Frequency Computation

Definition (Rose, 1960)

A set A is (m, n)-computable iff there is a total recursive function f which assigns to all distinct inputs $x_1, x_2, \dots, x_n$ a binary vector $(y_1, y_2, \dots, y_n)$ such that at least m of the equations $\chi_A(x_1) = y_1, \chi_A(x_2) = y_2, \dots, \chi_A(x_n) = y_n$ hold.



Frequency Computation

Theorem (Trakhtenbrot, 1964)

If $\frac{m}{n} > \frac{1}{2}$ *then every* (m, n)*-computable set is recursive.*

If $\frac{m}{n} \leq \frac{1}{2}$ *then there is a continuum of* (m, n)*-computable sets.*

Structured Frequency Computation

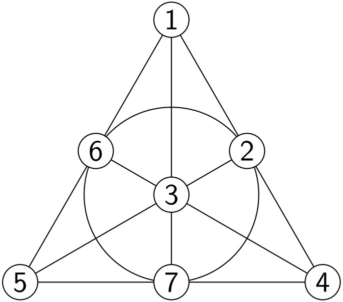
Definition

By a *structure* of a finite set K we call a set of K's subsets $S \subseteq 2^K$. Assume $K = \{1, 2, \ldots, n\}$.

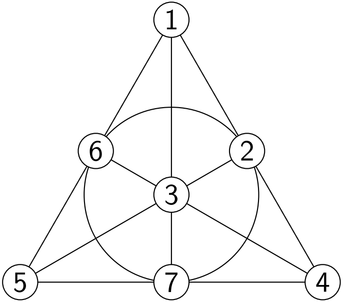
Definition

A set A is (S, K)-computable (or computable with a structure S) iff there is a total recursive function f which assigns to all distinct inputs $x_1, x_2, \ldots, x_n$ a binary vector $(y_1, y_2, \ldots, y_n)$ such that $\exists B \in S\ \forall b \in B\ \chi_A(x_b) = y_b$

Fano Frequency Computation



Fano Frequency Computation



Theorem

A set A is Fano-computable iff it is recursive.

Observation

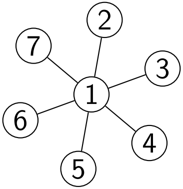
$\frac{3}{7} < \frac{1}{2}$

Some Properties of Fano structure

Definition

By the *size* of a structure $S \subseteq 2^K$ we denote the size of the smallest subset - $\min_{A \in S} |A|$. We call the structure *size consistent* iff $\neg \exists K' \subseteq K \ \ \min_{A' \in S} \frac{|A' \cap K'|}{|K'|} > \min_{A \in S} \frac{|A|}{|K|}$

To avoid such cases:



Some Properties of Fano structure

Definition

We call a structure $S \subseteq 2^K$ *overlapping* iff $\forall A, B \in S \; A \cap B \neq \emptyset$.

Overlapping Structures

Theorem

If a set A is computable with an overlapping structure then A is recursive.

Theorem

For any set K of size $n = q^2 + q + 1$ where q is a prime power there exists a size consistent overlapping structure of size $q + 1$.

Theorem

Every size consistent overlapping structure $S \subseteq 2^K$ has size at least $\sqrt{n}$ where $n = |K|$.

Overlapping Structures

The algorithm is asked to give the correct answer on a small fraction of inputs – $O\left(\frac{\sqrt{n}}{n}\right) = O\left(\frac{1}{\sqrt{n}}\right)$ – (instead of Trakhtenbrot's $\frac{1}{2}$) however only recursive set can be computed.

Graph Structures

Definition

We call a structure $S \subseteq 2^K$ a *graph structure* iff $\forall A \in S \ \ |A| = 2$.

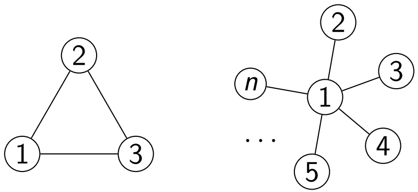
A natural question

For which graphs G are the G-computable sets recursive?

Recursive Graphs

Proposition

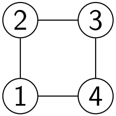
If the graph G is either a triangle C_3 or a star graph S_n then every G-computable set is recursive.

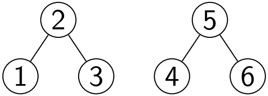


Continuum Implying Subgraphs

Theorem

If a graph G contains as a subgraph a cycle of length 4 (C_4) or two vertex-disjoint paths of length 3 then there is a continuum of G-computable sets, namely, every $(1,2)$-computable set is also G-computable.

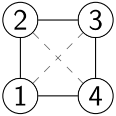


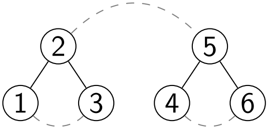


Continuum Implying Subgraphs

Theorem

If a graph G contains as a subgraph a cycle of length 4 (C_4) or two vertex-disjoint paths of length 3 then there is a continuum of G-computable sets, namely, every $(1,2)$-computable set is also G-computable.

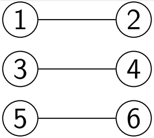




Two pairs vs three pairs

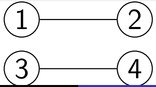
Theorem

If a graph G contains as a subgraph three vertex-disjoint paths of length 2 then there is a continuum of G-computable sets.



Theorem

If the graph G is two vertex-disjoint paths of length 2 then every G-computable set is recursive.



Publications

1. Kaspars Balodis, Jānis Iraids, Rūsiņš Freivalds.
   Structured Frequency Algorithms
   Proceedings of TAMC 2015, Lecture Notes in Computer Science, vol. 9076, pp. 1–12, 2015.
2. Rihards Krišlauks, Kaspars Balodis.
   On the Hierarchy Classes of Finite Ultrametric Automata
   Proceedings of SOFSEM 2015: Theory and Practice of Computer Science, Lecture Notes in Computer Science, vol. 8939, pp. 314–326, 2015.
3. Kaspars Balodis. **Counting With Probabilistic and Ultrametric Finite Automata**
   Computing with New Resources: Essays Dedicated to Jozef Gruska on the Occasion of His 80th Birthday, Lecture Notes in Computer Science, vol. 8808, pp. 1-14, 2014.
4. Kārlis Jēriņš, Kaspars Balodis, Rihards Krišlauks, Kristīne Cīpola, Rūsiņš Freivalds. **Ultrametric query algorithms**
   Proceedings of SOFSEM 2014, Volume II: Student Research Forum, pp. 21-29, 2014.
5. Kaspars Balodis. **One Alternation Can Be More Powerful Than Randomization in Small and Fast Two-Way Finite Automata**
   Proceedings of FCT 2013, Lecture Notes in Computer Science, vol. 8070, pp. 40-47, 2013.
6. Kaspars Balodis, Anda Beriņa, Kristīne Cīpola, Maksims Dimitrijevs, Jānis Iraids, Kārlis Jēriņš, Vladimirs Kacs, Jānis Kalējs, Rihards Krišlauks, Kārlis Lukstiņš, Reinholds Raumanis, Natālija Somova, Irina Ščeguļnaja, Anna Vanaga, Rūsiņš Freivalds. **On the State Complexity of Ultrametric Finite Automata**
   Proceedings of SOFSEM 2013, Volume II: Student Research Forum, pp. 1-9, 2013.
7. Kaspars Balodis, Anda Beriņa, Gleb Borovitsky, Rūsiņš Freivalds, Ginta Garkāje, Vladimirs Kacs, Jānis Kalējs, Ilja Kucevalovs, Jānis Ročāns, Madars Virza. **Probabilistic and Frequency Finite-State Transducers**
   Proceedings of SOFSEM 2012, Volume II: Student Research Forum, pp. 1-12, 2012.

Presentations

1. SOFSEM 2015 (41st International Conference on Current Trends in Theory and Practice of Computer Science), Pec pod Sněžkou, Czech Republic, 2015.
   Presentation: **On the Hierarchy Classes of Finite Ultrametric Automata.**
2. Joint Estonian-Latvian Theory Days, Ratnieki, Latvia, 2014.
   Presentation: **Structured Frequency Algorithms.**
3. Latvijas Universitātes 72. konference, Rīga, Latvia, 2014.
   Presentation: **Par divvirzienu galīgiem automātiem.**
4. SOFSEM 2014 (40th International Conference on Current Trends in Theory and Practice of Computer Science), Nový Smokovec, High Tatras, Slovakia, 2014.
   Poster presentation: **Ultrametric query algorithms.**
5. FCT 2013 (19th International Symposium on Fundamentals of Computation Theory), Liverpool, United Kingdom, 2013.
   Presentation: **One Alternation Can Be More Powerful Than Randomization in Small and Fast Two-Way Finite Automata.**
6. SOFSEM 2013 (39th International Conference on Current Trends in Theory and Practice of Computer Science), Špindlerův Mlýn, Czech Republic, 2013.
   Poster presentations: **On the State Complexity of Ultrametric Finite Automata**, and **Ultrametric Turing Machines with Limited Reversal Complexity.**
7. Latvijas Universitātes 70. konference, Rīga, Latvia, 2012.
   Presentation: **Par galīgiem automātiem uz bezgalīgas lentas.**
8. SOFSEM 2012 (38th International Conference on Current Trends in Theory and Practice of Computer Science), Špindlerův Mlýn, Czech Republic, 2012.
   Poster presentation: **Probabilistic and Frequency Finite-State Transducers.**

Thank you!
Questions?