Structured Frequency Algorithms

Kaspars Balodis, Jānis Iraids, Rūsiņš Freivalds

October 3, 2014

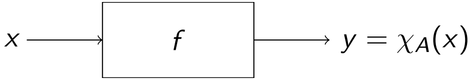
Recursive Sets

Definition

$$A \subseteq \mathbb{N}, \quad \chi_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

Definition

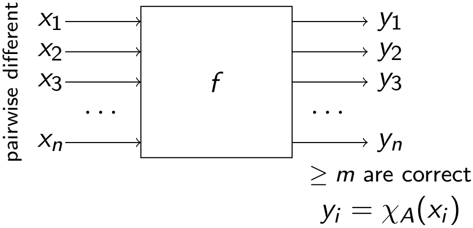
A set A is recursive iff there is a Turing machine (an algorithm) that computes $\chi_A(x)$.



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.



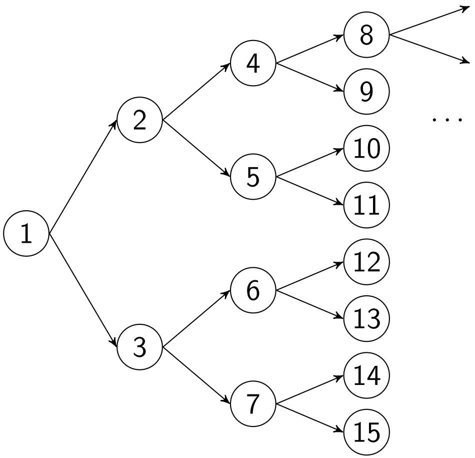
Frequency Computation

Theorem (Trakhtenbrot, 1964)

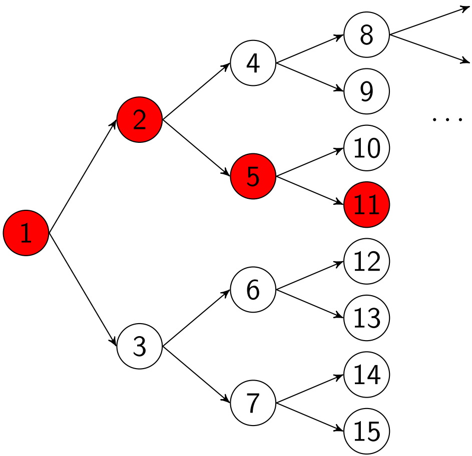
If $\frac{m}{n} > \frac{1}{2}$ then every (m, n)-computable set is recursive.

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

(1, 2)-computation



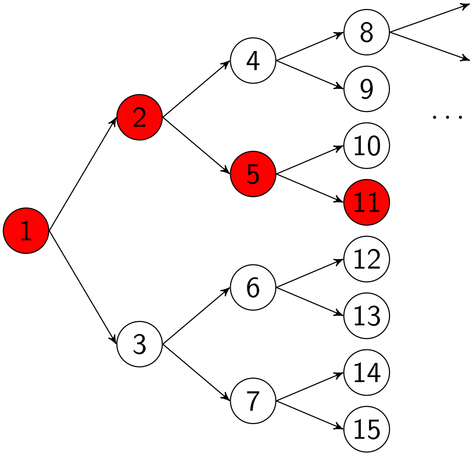
$(1, 2)$-computation



Choose an infinite branch T.
$x \in A_T$ iff x is on the branch T.

$A_T = \{1, 2, 5, 11, \ldots\}$

(1, 2)-computation

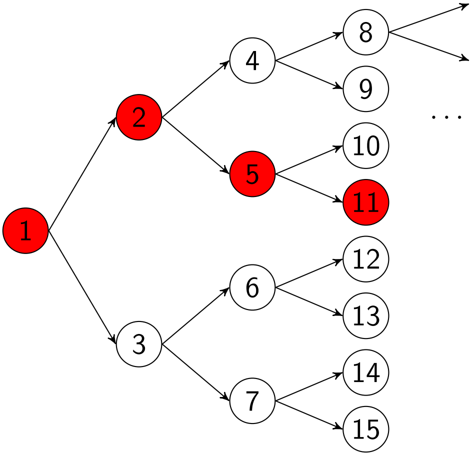


Choose an infinite branch T.
$x \in A_T$ iff x is on the branch T.

$A_T = \{1, 2, 5, 11, \dots\}$

(1, 2)-algorithm receives 2 different numbers and on at least one of them has to give the correct answer.

(1, 2)-computation

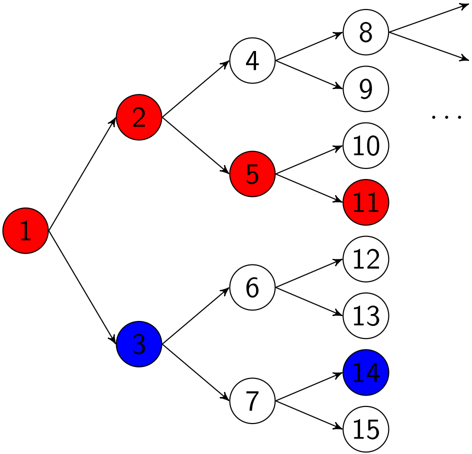


Algorithm

Assume $x_1 < x_2$.

- If there is a branch which contains both x_1 and x_2 then output: $x_1 \in A_T,\ x_2 \notin A_T$.
- Otherwise output: $x_1 \notin A_T,\ x_2 \notin A_T$.

$(1, 2)$-computation

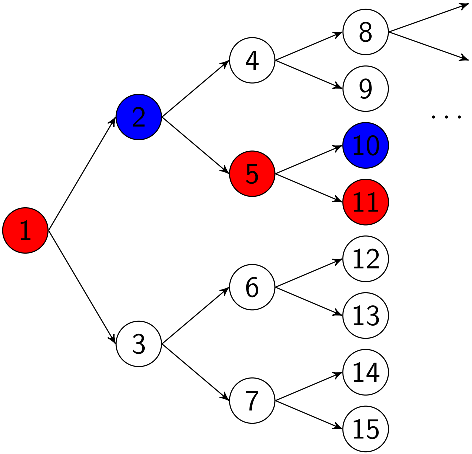


Algorithm

Assume $x_1 < x_2$.

- If there is a branch which contains both x_1 and x_2 then output: $x_1 \in A_T$, $x_2 \notin A_T$.
- Otherwise output: $x_1 \notin A_T$, $x_2 \notin A_T$.

(1, 2)-computation

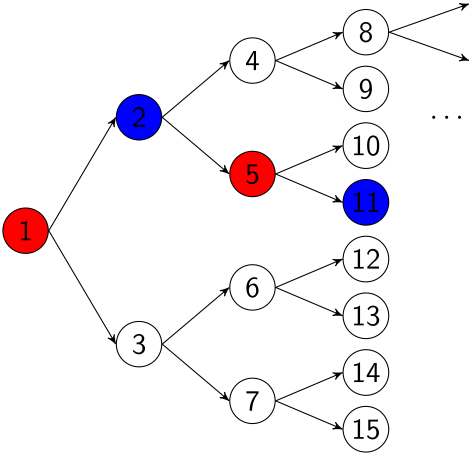


Algorithm

Assume $x_1 < x_2$.

- If there is a branch which contains both x_1 and x_2 then output: $x_1 \in A_T$, $x_2 \notin A_T$.
- Otherwise output: $x_1 \notin A_T$, $x_2 \notin A_T$.

(1, 2)-computation

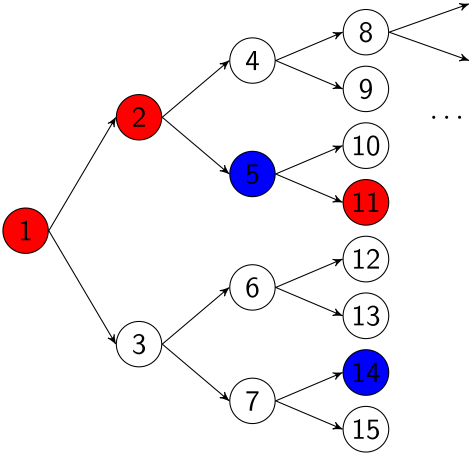


Algorithm

Assume $x_1 < x_2$.

- If there is a branch which contains both x_1 and x_2 then output: $x_1 \in A_T,\ x_2 \notin A_T$.
- Otherwise output: $x_1 \notin A_T,\ x_2 \notin A_T$.

$(1, 2)$-computation

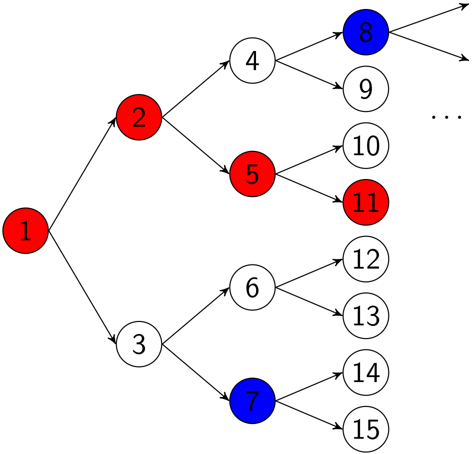


Algorithm

Assume $x_1 < x_2$.

- If there is a branch which contains both x_1 and x_2 then output: $x_1 \in A_T,\ x_2 \notin A_T$.
- Otherwise output: $x_1 \notin A_T,\ x_2 \notin A_T$.

(1, 2)-computation

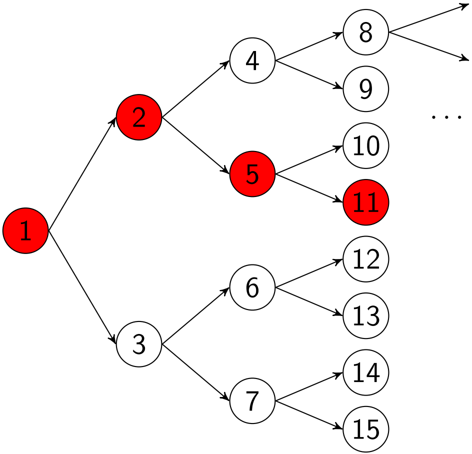


Algorithm

Assume $x_1 < x_2$.

- If there is a branch which contains both x_1 and x_2 then output: $x_1 \in A_T$, $x_2 \notin A_T$.
- Otherwise output: $x_1 \notin A_T$, $x_2 \notin A_T$.

(1, 2)-computation



The algorithm does not depend on T.

T can be chosen in a continuum different ways.

The algorithm $(1, 2)$-computes a continuum of different sets.

There are only countably many recursive sets.

Structured Frequency Computation

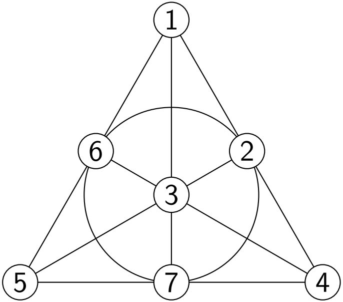
Definition

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

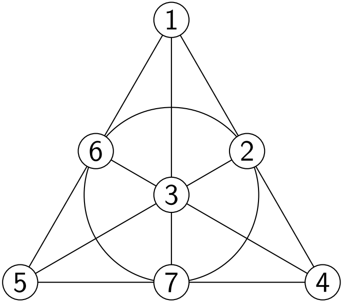
Definition

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

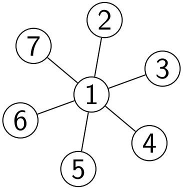
$\frac{3}{7} < \frac{1}{2}$

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 (Projective plane of order q)

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 $\forall A \in S \ \ |A| = 2$.

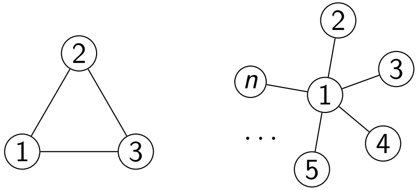
A natural question

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

Recursive Graphs

Proposition

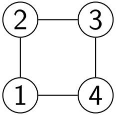
If the graph G is either a triangle C_3 or a star graph S_n then every G-computable set is recursive.

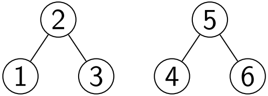


Continuum Implying Subgraphs

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.

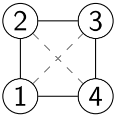


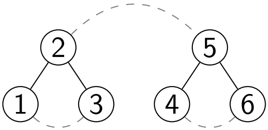


Continuum Implying Subgraphs

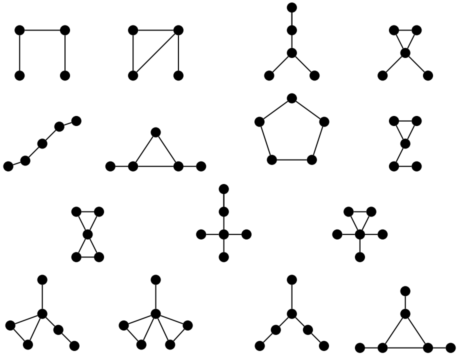
Theorem

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.





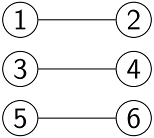
Small Connected Graphs ($\leq$ 6 vertices)



Recent Developments

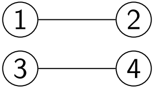
Theorem

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

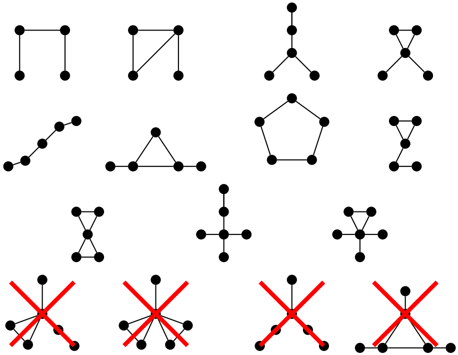


Theorem

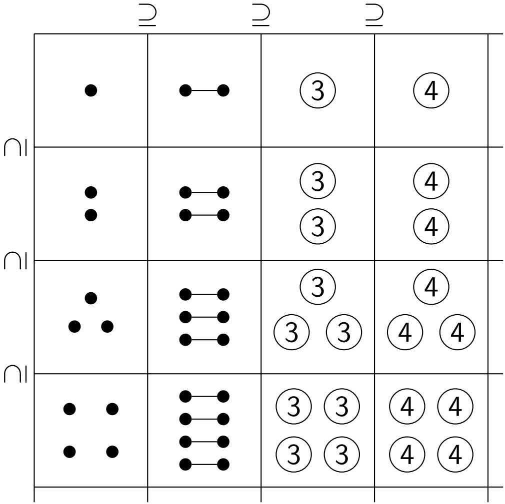
If the graph G is two vertex-disjoint paths of length 2 then every G-computable set is recursive.



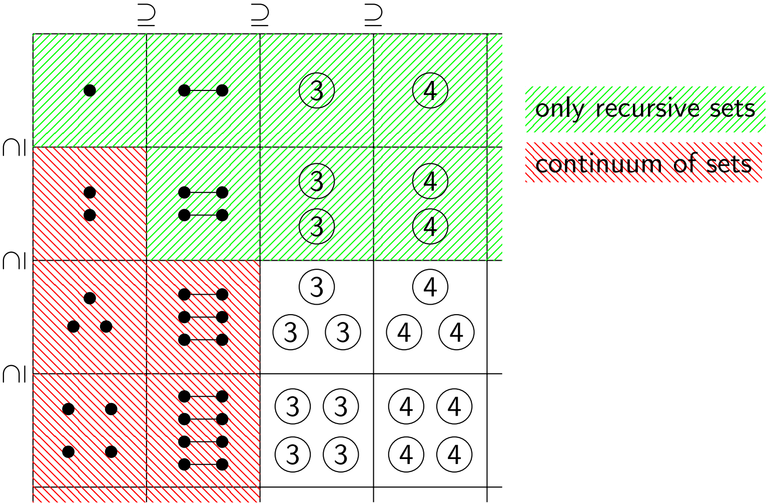
Small Connected Graphs ($\leq$ 6 vertices)



Generalizations



Generalizations



Open Problems

Open Problems

- Are there other size consistent non-overlapping structures of size less than $\sqrt{n}$ that allow only computability of recursive sets? If so then what is the smallest possible fraction of correct answers attainable?
- For graph frequency computation obtain a complete classification of all graphs G and classes of G-computable sets.
- What other types of structures are interesting and worth considering and what classes of sets are computable with them?

Thank you!
Questions?