Balanced Structures and Scheduling Applications

Bruno Gaujal   Nicolas Gast

INRIA Rhône-Alpes

Aussois, 2008

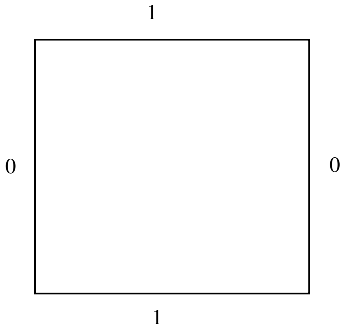
Sturmian Words: 3 equivalent definitions

Consider an infinite word:

$$001010010010100101 00100\ldots$$

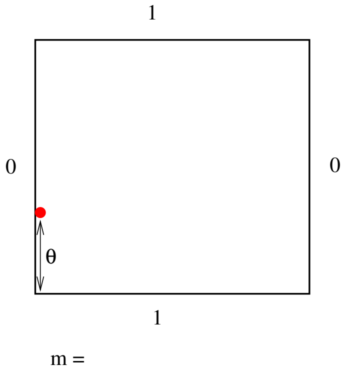
- minimal complexity : $n+1$ factors of length n.
  example: 4 factors of length 3: 001, 010, 100 and 101.
- balanced : number of 1 only differ by 1 in factors of same length.
  - length 3: 1 or 2.
  - length 4: 1 or 2.
  - ...
- mechanical:
  - for all i: $w_i = \lfloor \alpha(i+1)+\theta \rfloor - \lfloor \alpha i + \theta \rfloor$
    or for all i: $w_i = \lceil \alpha(i+1)+\theta \rceil - \lceil \alpha i + \theta \rceil$

Construction using a billiard sequence

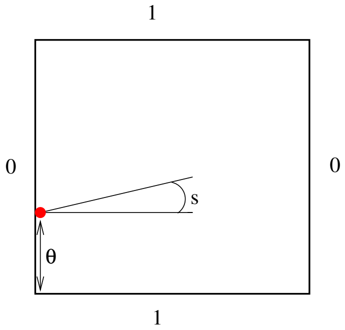


m =

Construction using a billiard sequence

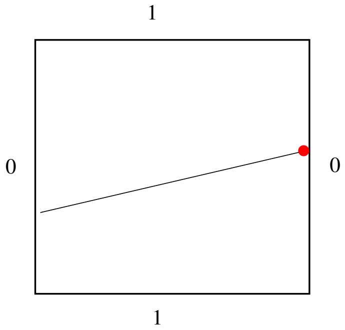


Construction using a billiard sequence



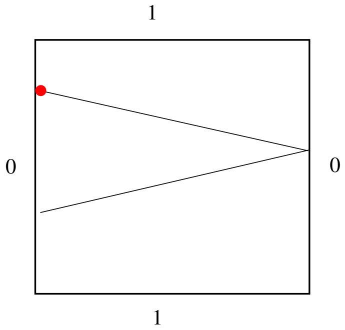
m =

Construction using a billiard sequence



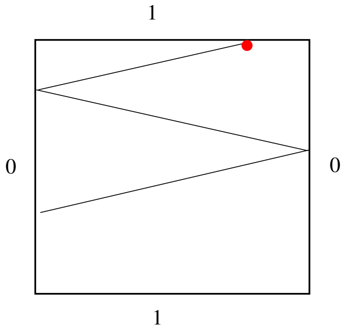
m = 0

Construction using a billiard sequence



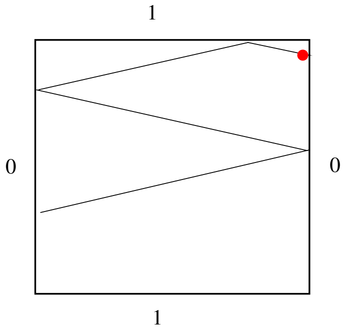
m = 0 0

Construction using a billiard sequence



m = 0 0 1

Construction using a billiard sequence



m = 0 0 1 0

Application to mapping

Consider a scheduling problem with two processors, one bag of tasks with stationary release times and stationary service times, independent of the release times.

A simple deterministic case is when the tasks are released at every time unit and the service times are S^1 and S^2 on both processors respectively (with $1/S^1 + 1/S^2 > 1$).
The objective is to minimize the (expected) flow-time of the tasks.

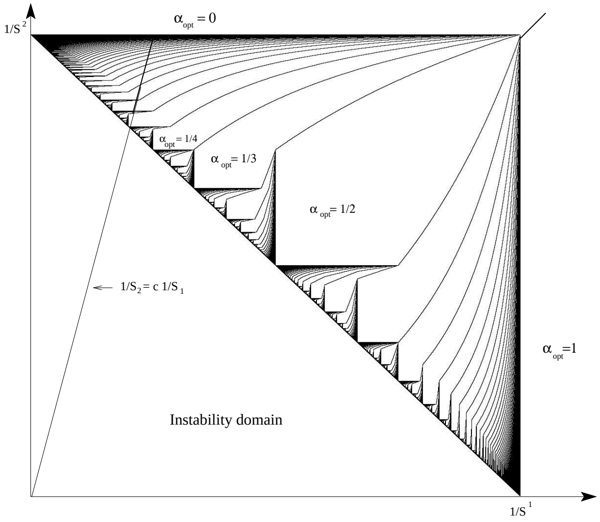
Application to mapping

Consider a scheduling problem with two processors, one bag of tasks with stationary release times and stationary service times, independent of the release times.

A simple deterministic case is when the tasks are released at every time unit and the service times are S^1 and S^2 on both processors respectively (with $1/S^1 + 1/S^2 > 1$).
The objective is to minimize the (expected) flow-time of the tasks.

The solution is given by a Sturmian sequence with density $\alpha = f(S^1, S^2)$. (Altman, G., Hordijk, 2001).

Computing the optimal density



Application to Polling

Consider a scheduling problem with one processor, two bags of tasks with stationary release times (common to the two bags of tasks) and stationary service times.

A simple deterministic case is when the two bags release one task at every time unit and the service times are S^1 and S^2 respectively (with $S^1 + S^2 < 1$).
The objective is to minimize the (expected) flow-time of the tasks.

Application to Polling

Consider a scheduling problem with one processor, two bags of tasks with stationary release times (common to the two bags of tasks) and stationary service times.

A simple deterministic case is when the two bags release one task at every time unit and the service times are S^1 and S^2 respectively (with $S^1 + S^2 < 1$).
The objective is to minimize the (expected) flow-time of the tasks.

The solution is given by a Sturmian sequence with density $\alpha = g(S^1, S^2)$. (G., Hordijk, Van der Laan, 2007).

Several extensions

Sturmian sequences in dimension higher than 1 have been studied in (Cassaigne, 2000, Fernique, 2007, Vuillon, 2005, Berthé 2005) on a theoretical point of view.

Several extensions

Sturmian sequences in dimension higher than 1 have been studied in (Cassaigne, 2000, Fernique, 2007, Vuillon, 2005, Berthé 2005) on a theoretical point of view.

As for scheduling problems, it is rather natural to extend this notion to other dependency structures.

Several extensions

Sturmian sequences in dimension higher than 1 have been studied in (Cassaigne, 2000, Fernique, 2007, Vuillon, 2005, Berthé 2005) on a theoretical point of view.

As for scheduling problems, it is rather natural to extend this notion to other dependency structures.

Here we consider the case where tasks are released according to a partial order forming a binary tree. All inter-release times are i.i.d.

Several extensions

Sturmian sequences in dimension higher than 1 have been studied in (Cassaigne, 2000, Fernique, 2007, Vuillon, 2005, Berthé 2005) on a theoretical point of view.

As for scheduling problems, it is rather natural to extend this notion to other dependency structures.

Here we consider the case where tasks are released according to a partial order forming a binary tree. All inter-release times are i.i.d. Two processors with different speeds can execute the tasks.

Several extensions

Sturmian sequences in dimension higher than 1 have been studied in (Cassaigne, 2000, Fernique, 2007, Vuillon, 2005, Berthé 2005) on a theoretical point of view.

As for scheduling problems, it is rather natural to extend this notion to other dependency structures.

Here we consider the case where tasks are released according to a partial order forming a binary tree. All inter-release times are i.i.d. Two processors with different speeds can execute the tasks.

Question: how to allocate the tasks to the processors in order to minimize the expected flow time?

Problem

Can we extend balancedness to trees?

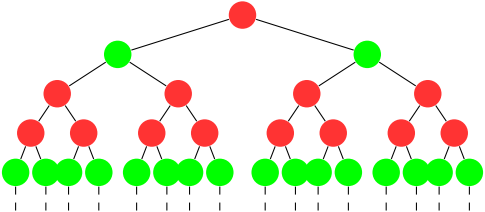
- Sturmian
- balanced
- mechanical

Previous Work

Definition (Berstel, Boasson, Carton and Fagnot, 2007)

A Sturmian tree is a tree with $n+1$ subtrees of size n.

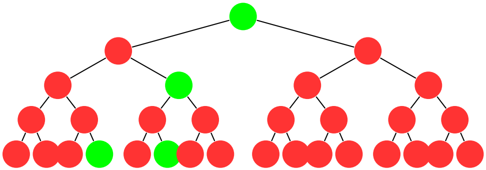
Simple example:



Example: The uniform tree corresponding to 0100101 . . .

Properties

- Link with language theory
- Interesting examples:



Example: Dyck Trees

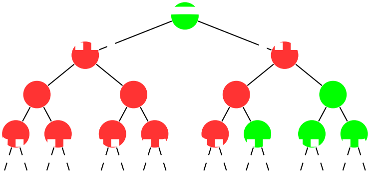
But

- the *balanced* property is lost (important in optimization problems)
- no simple equivalent characterization

Infinite Labeled Non-Planar Trees

Here, trees are:

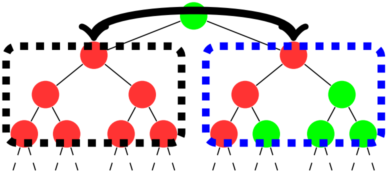
- rooted
- labeled by 0 or 1
- infinite
- Non-ordered
  (≠ Original definition
  for Sturmian Trees)



Infinite Labeled Non-Planar Trees

Here, trees are:

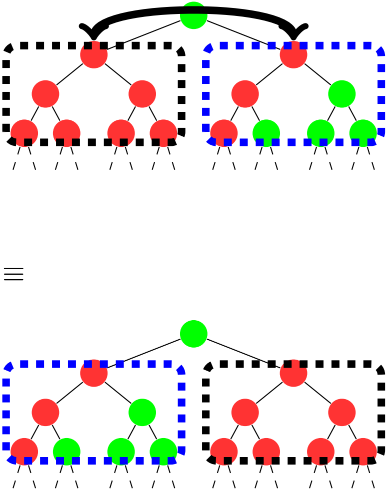
- rooted
- labeled by 0 or 1
- infinite
- Non-ordered
  ($\neq$ Original definition
  for Sturmian Trees)



Infinite Labeled Non-Planar Trees

Here, trees are:

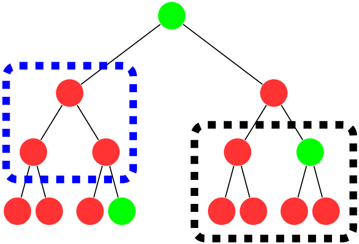
- rooted
- labeled by 0 or 1
- infinite
- Non-ordered
  (≠ Original definition
  for Sturmian Trees)



What are Subtrees and Density?

We define:

- Factor of height n (subtree).
- Factor of width k and height n
- Density of a factor = average number of 1.
- If d_n is the density of the factor of height n:
  - density = $\lim_n d_n$
  - average density = $\lim_n \frac{1}{n} \sum_{k=1}^{n} d_k$



First simple case

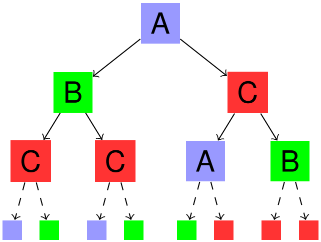
What is a non-planar Rational Tree?

Rational Trees: Definition

We call $P(n) =$ number of factors of size n.

Rational Trees: 3 equivalent definitions:

- $P(n)$ bounded.
- $\exists n / P(n) = P(n+1)$
- $\exists n / P(n) \leq n$.



Rational Tree: average Density

Theorem 1.

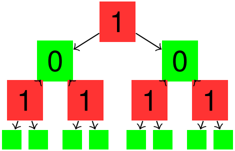
- A rational Tree has an average density α which is rational.

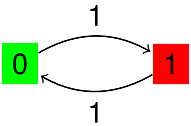
  α is not necessarily a density but:

- If the associated Markov chain is aperiodic then α is a density.

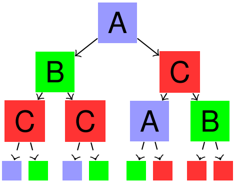
Example of density

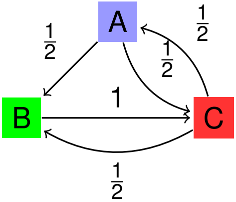
- Periodic = average density $d_{\text{average}} = \frac{1}{2}$





- Aperiodic : density $d = \frac{2}{9}\ell_A + \frac{1}{3}\ell_B + \frac{4}{9}\ell_C$





Second case

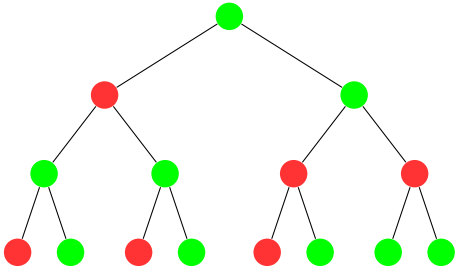
Balanced and Mechanical Trees

Balanced Trees and Strongly Balanced Trees

- Balanced tree: number of 1 in factors of height n only differ by 1.
- Strongly balanced tree: same property with factors of height n and width k.

Balanced Trees and Strongly Balanced Trees

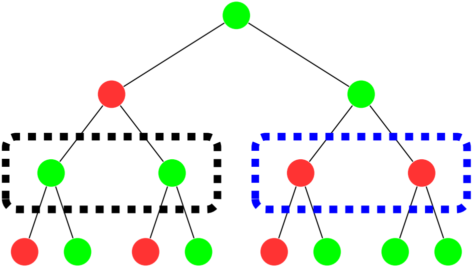
- Balanced tree: number of 1 in factors of height n only differ by 1.
- Strongly balanced tree: same property with factors of height n and width k.



Example: Balanced tree not strongly balanced

Balanced Trees and Strongly Balanced Trees

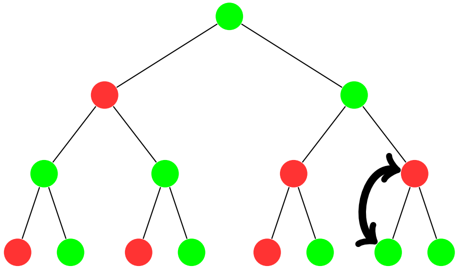
- Balanced tree: number of 1 in factors of height n only differ by 1.
- Strongly balanced tree: same property with factors of height n and width k.



Example: Balanced tree not strongly balanced

Balanced Trees and Strongly Balanced Trees

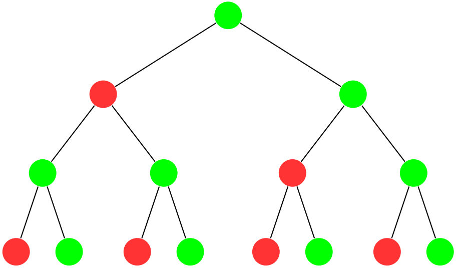
- Balanced tree: number of 1 in factors of height n only differ by 1.
- Strongly balanced tree: same property with factors of height n and width k.



Example: Balanced tree not strongly balanced

Balanced Trees and Strongly Balanced Trees

- Balanced tree: number of 1 in factors of height *n* only differ by 1.
- Strongly balanced tree: same property with factors of height *n* and width *k*.



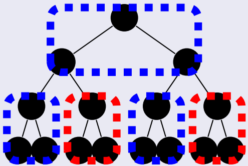
Example: Strongly balanced tree

Density of a Balanced Tree

Theorem 2.

- A balanced tree has a density.

Sketch of the proof.

1. A tree of size n has a density α_n or $\alpha_n + \frac{1}{2^n-1}$
2. 
   If blue has density α_2 and red $\alpha_2 + \frac{1}{3}$ then $\alpha_2 \leq \alpha_4 \leq \alpha_2 + \frac{1}{3}$
3. Take limit

□

Mechanical Trees

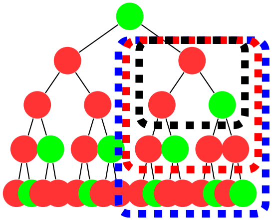
- Subtree of size n has $2^n - 1$ nodes.
- We want density α

Mechanical Trees

- Subtree of size n has $2^n - 1$ nodes.
- We want density α

Mechanical tree of density α:

- For all node i, there is a phase $\phi_i \in [0; 1)$ such that the number of 1 in a subtree of height n and root i is $\lfloor (2^n - 1)\alpha + \phi_i \rfloor$ (resp. for all i: $\lceil (2^n - 1)\alpha + \phi_i \rceil$)



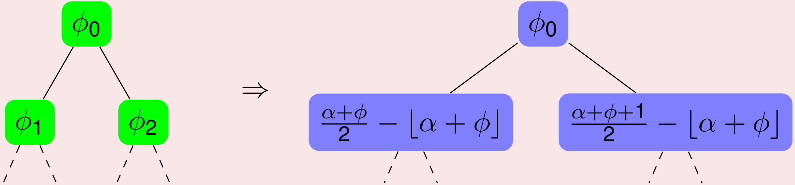
$\alpha = 0.3$, $\phi = 0.55$.

n	$(2^n - 1)\alpha + \phi$
1	0.85
2	1.45
3	2.65
4	5.05

Uniqueness of a mechanical Tree

Theorem 3.

- There exists a unique mechanical tree if (α, ϕ_0) is fixed.



- The phase of the root ϕ_0 is unique for almost all α.

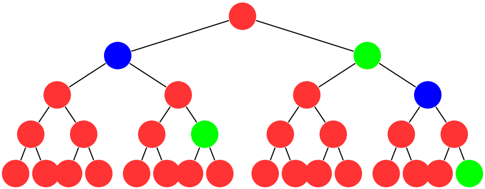
Equivalences?

What are the equivalences between definitions?

Equivalences between Definitions

Theorem 4: Mechanical $\sim$ strongly balanced.

- A mechanical tree is strongly balanced
- A strongly balanced tree with irrational density is mechanical
- A strongly balanced tree with rational density is ultimately mechanical.



Example: Ultimately mechanical tree

Sketch of Proof

Mechanical implies strongly balanced.

The number of 1 in a factor of size n and width k is bounded by $\lfloor (2^n - 2^k)\alpha \rfloor$ and $\lfloor (2^n - 2^k)\alpha \rfloor + 1$ □

Strongly Balanced implies mechanical.

$\forall \tau \in [0; 1)$, if h_n is the number of 1 in the subtree of size n, at least one of these properties is true:

1. for all n: $h_n \leq \lfloor (2^n - 1)\alpha + \tau \rfloor$,
2. for all n: $h_n \geq \lfloor (2^n - 1)\alpha + \tau \rfloor$.

Choose ϕ the maximal τ such that 1 is true. □

Theorem 5.

- An irrational mechanical tree is a Sturmian tree: it has $n+1$ subtrees of height n.

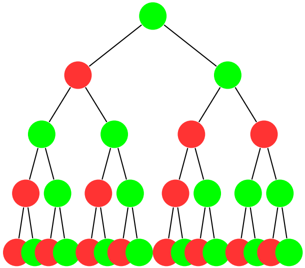
Proof.

- A subtree of size n depends only on its phase
- In fact, it depends on $((2^1-1)\alpha+\phi,\ldots,(2^n-1)\alpha+\phi)$ which takes $n+1$ values when $\phi \in [0;1)$.

□

Limit of the Equivalences

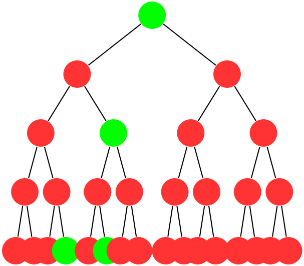
- Balanced $\not\Rightarrow$ strongly balanced (no matter whether the density is rational or not).
- Sturmian $\not\Rightarrow$ balanced.
- Irrational Balanced tree $\not\Rightarrow$ Sturmian.



Example: Balanced tree not str. bal.

Limit of the Equivalences

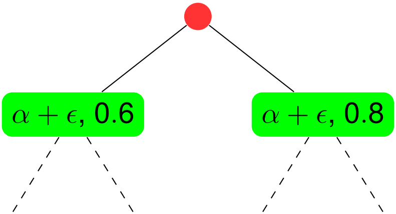
- Balanced $\not\Rightarrow$ strongly balanced (no matter whether the density is rational or not).
- Sturmian $\not\Rightarrow$ balanced.
- Irrational Balanced tree $\not\Rightarrow$ Sturmian.



Example: Dyck Tree

Limit of the Equivalences

- Balanced $\not\Rightarrow$ strongly balanced (no matter whether the density is rational or not).
- Sturmian $\not\Rightarrow$ balanced.
- Irrational Balanced tree $\not\Rightarrow$ Sturmian.



Example: Balanced tree non Sturmian

Optimization Issues

Let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be a convex function. For each node n and each height $k > 0$, we define a cost $C_{[n,k]}$:

$$C_{[n,k]} = g(d(\mathcal{A}_{n,k})).$$

cost of order k of the tree is:

$$C_k = \limsup_{\ell \to \infty} \frac{\sum_{n \in \mathcal{A}_{0,\ell}} C_{[n,k]}}{2^\ell - 1}.$$

Optimization Issues

Let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be a convex function. For each node n and each height $k > 0$, we define a cost $C_{[n,k]}$:

$$C_{[n,k]} = g(d(\mathcal{A}_{n,k})).$$

cost of order k of the tree is:

$$C_k = \limsup_{\ell \to \infty} \frac{\sum_{n \in \mathcal{A}_{0,\ell}} C_{[n,k]}}{2^\ell - 1}.$$

If g has a minimum in α, C_k is minimized when the number of 1 in a tree of height k is between $\lfloor \alpha(2^k - 1) \rfloor$ and $\lceil \alpha(2^k - 1) \rceil$. That means that a balanced tree will minimize any increasing function of all C_k.

Optimization Issues

Let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be a convex function. For each node n and each height $k > 0$, we define a cost $C_{[n,k]}$:

$$C_{[n,k]} = g(d(\mathcal{A}_{n,k})).$$

cost of order k of the tree is:

$$C_k = \limsup_{\ell \to \infty} \frac{\sum_{n \in \mathcal{A}_{0,\ell}} C_{[n,k]}}{2^\ell - 1}.$$

If g has a minimum in α, C_k is minimized when the number of 1 in a tree of height k is between $\lfloor \alpha(2^k - 1) \rfloor$ and $\lceil \alpha(2^k - 1) \rceil$. That means that a balanced tree will minimize any increasing function of all C_k.

This has potential applications in optimization problem in distributed systems with a binary causal structure.

Conclusion

- Non-planar definition better?
- Constructive definition
- Strong inclusions
- Good characterization

but:

- What are exactly balanced trees?
- How many balanced trees of size n?