Pattern avoidance in binary trees

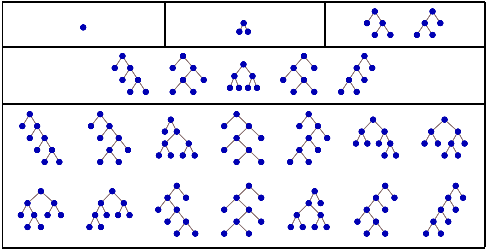
Eric Rowland
erowland@math.rutgers.edu

Department of Mathematics
Rutgers University, New Brunswick / Piscataway

January 5, 2009

Binary trees

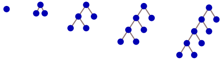
Binary trees with ≤ 5 leaves:

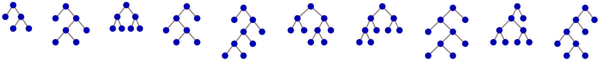


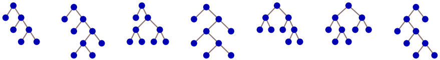
Pattern containment

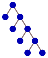
Patterns are contiguous. For example, let $t =$ .

Small binary trees containing...

0 copies of t: 

1 copy of t: 

2 copies of t: 

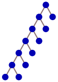
3 copies of t: 

Small patterns

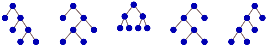
What is the number $a(n)$ of n-leaf binary trees avoiding t?

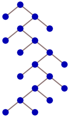
1-leaf tree patterns: $t = \bullet$.
$a(n) = 0$.

2-leaf tree patterns: $t =$ .
$a(1) = 1$; $a(n) = 0$ for $n \geq 2$.

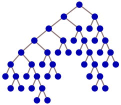
3-leaf tree patterns:  and . “Typical” tree avoiding : .
$a(n) = 1$.

4-leaf tree patterns

4-leaf tree patterns: .

- $t =$ . A “typical” tree avoiding t looks like .

  $a(1) = 1$; $a(n) = 2^{n-2}$ for $n \geq 2$.

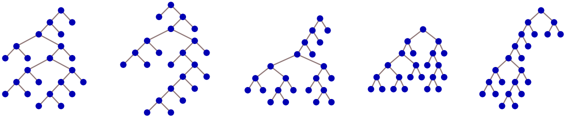
- $t =$ . A “typical” tree avoiding t looks like .

  $a(1) = 1$; $a(n) = 2^{n-2}$ for $n \geq 2$.

These two patterns are equivalent.

The remaining 4-leaf pattern

- $t =$ . Some trees avoiding t:



$a(n) = M_{n-1}$ (a Motzkin number[1]).

[1] Robert Donaghey and Louis Shapiro, Motzkin numbers, *Journal of Combinatorial Theory, Series A* **23** (1977) 291–301.

Systematic enumeration

Is there a systematic way to compute $a(n)$ for an arbitrary pattern t?

$$\text{Let } \mathrm{Av}_t(x) = \sum_{T \text{ avoids } t} x^{\text{number of leaves in } T} = \sum_{n=0}^{\infty} a(n) x^n.$$

Theorem

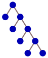
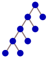
$\mathrm{Av}_t(x)$ *is algebraic.*

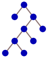
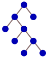
The proof is constructive.

5-leaf equivalence classes

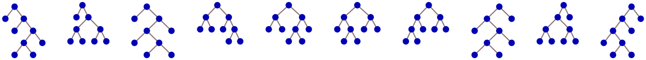
A computer implementation produces all equivalence classes for binary trees up to 8 leaves.

For 5-leaf tree patterns:

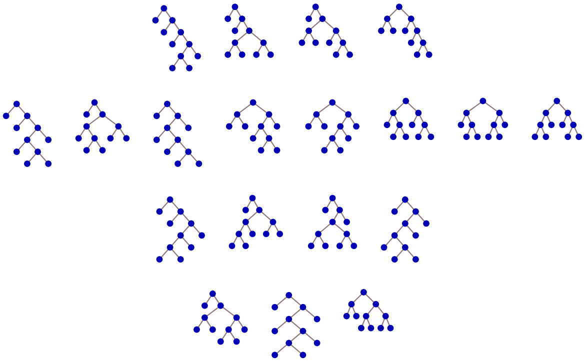
 and  form an equivalence class.

 and  form an equivalence class.

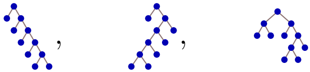
The other 10 tree patterns are equivalent:



6-leaf equivalence classes



Isolated trees:

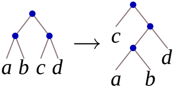


Bijective proofs

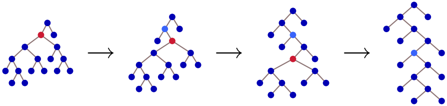
Given two equivalent tree patterns *s* and *t*, can we find a bijective proof of the equivalence?

For example,  and  are equivalent.

Let *T* avoid . Idea: Replace all instances of  with .
How?

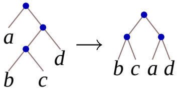


What order? top-down.
For example:



Inverse

The inverse map is a *bottom-up replacement* with the inverse replacement rule:



Conjecture

Not every pair of equivalent trees can be shown equivalent by such a bijection. However, the following appears to hold.

Conjecture

Two binary tree patterns s and t are equivalent if and only if there is a sequence of

- *top-down replacements,*
- *bottom-up replacements, and*
- *left–right reflections*

that produces a bijection from binary trees avoiding s to binary trees avoiding t.

The conjecture is true for tree patterns of ≤ 7 leaves.