Probabilistic Proofs of Hooklength Formulas

Bruce Sagan
Department of Mathematics
Michigan State University
East Lansing, MI 48824-1027
sagan@math.msu.edu
www.math.msu.edu/~sagan

October 26, 2009

Outline

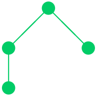
Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set.

Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L : T \rightarrow \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

$$f^T = \text{number of increasing labelings of } T.$$

Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L: T \to \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

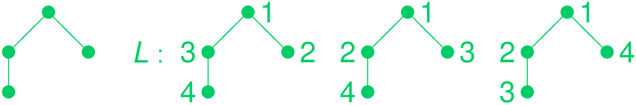
$$f^T = \text{number of increasing labelings of } T.$$

Ex.
$T =$ 

Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L : T \to \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

$$f^T = \text{number of increasing labelings of } T.$$

Ex. $T =$, $L:$

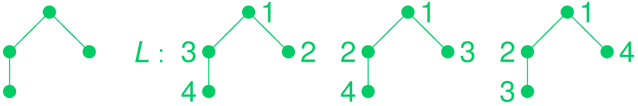


$f^T = 3$

Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L : T \to \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

$$f^T = \text{number of increasing labelings of } T.$$

Ex. $T =$



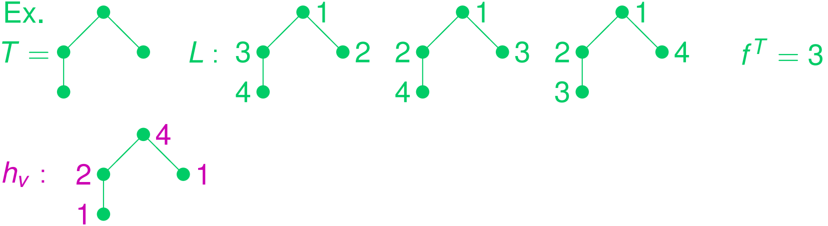
$f^T = 3$

The *hooklength* of a vertex v is

$$h_v = \text{ number of descendents of } v \text{ (including } v\text{)}.$$

Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L : T \to \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

$$f^T = \text{number of increasing labelings of } T.$$

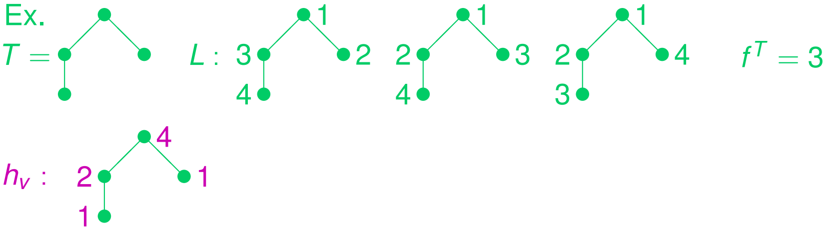


The *hooklength* of a vertex v is

$$h_v = \text{ number of descendents of } v \text{ (including } v\text{)}.$$

Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L : T \to \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

$$f^T = \text{number of increasing labelings of } T.$$



The *hooklength* of a vertex v is

$$h_v = \text{ number of descendents of } v \text{ (including } v\text{).}$$

Theorem (Hooklength Formula)

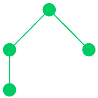
If T has n vertices, then

$$f^T = \frac{n!}{\prod_{v \in T} h_v}.$$

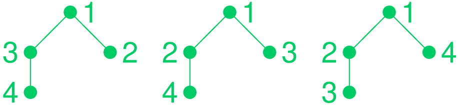
Let T be a rooted tree with n distinguishable vertices. We also use T for its vertex set. An *increasing labeling* of T is a bijection $L : T \to \{1, 2, \ldots, n\}$ such that if vertex v has a child w then $L(v) < L(w)$. Let

$$f^T = \text{number of increasing labelings of } T.$$

Ex. $T =$

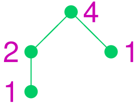


L :



$f^T = 3$

h_v :



$$f^T = \frac{4!}{4 \cdot 2 \cdot 1^2} = 3.$$

The *hooklength* of a vertex v is

$$h_v = \text{ number of descendents of } v \text{ (including } v).$$

Theorem (Hooklength Formula)

If T has n vertices, then
$$f^T = \frac{n!}{\prod_{v \in T} h_v}.$$

History.

1. There are also hooklength formulas for
   1.1 ordinary Young tableaux (Frame-Robinson-Thrall),

History.

1. There are also hooklength formulas for
   1.1 ordinary Young tableaux (Frame-Robinson-Thrall),
   1.2 shifted Young tableaux (Knuth), and

History.

1. There are also hooklength formulas for
   1.1 ordinary Young tableaux (Frame-Robinson-Thrall),
   1.2 shifted Young tableaux (Knuth), and
   1.3 d-complete posets (Proctor).

History.

1. There are also hooklength formulas for
    1.1 ordinary Young tableaux (Frame-Robinson-Thrall),
    1.2 shifted Young tableaux (Knuth), and
    1.3 d-complete posets (Proctor).
2. Probabilistic proofs of these formulas were given by
    2.1 Greene-Nijenhuis-Wilf (ordinary tableaux),

History.

1. There are also hooklength formulas for
   1.1 ordinary Young tableaux (Frame-Robinson-Thrall),
   1.2 shifted Young tableaux (Knuth), and
   1.3 *d*-complete posets (Proctor).
2. Probabilistic proofs of these formulas were given by
   2.1 Greene-Nijenhuis-Wilf (ordinary tableaux),
   2.2 S (shifted tableaux),

History.

1. There are also hooklength formulas for
   1.1 ordinary Young tableaux (Frame-Robinson-Thrall),
   1.2 shifted Young tableaux (Knuth), and
   1.3 d-complete posets (Proctor).
2. Probabilistic proofs of these formulas were given by
   2.1 Greene-Nijenhuis-Wilf (ordinary tableaux),
   2.2 S (shifted tableaux),
   2.3 S-Yeh (trees),

History.

1. There are also hooklength formulas for
   1.1 ordinary Young tableaux (Frame-Robinson-Thrall),
   1.2 shifted Young tableaux (Knuth), and
   1.3 d-complete posets (Proctor).
2. Probabilistic proofs of these formulas were given by
   2.1 Greene-Nijenhuis-Wilf (ordinary tableaux),
   2.2 S (shifted tableaux),
   2.3 S-Yeh (trees),
   2.4 Okamura (d-complete).

Let

$\mathcal{B}(n) =$ set of binary trees, T, on n vertices.

Let

$$\mathcal{B}(n) = \text{ set of binary trees, } T\text{, on } n \text{ vertices.}$$

Let

$$\mathcal{L}(n) = \text{ set of all increasing labelings, } L\text{, of trees in } \mathcal{B}(n).$$

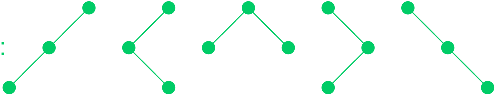
Let

$$\mathcal{B}(n) = \text{ set of binary trees, } T\text{, on } n \text{ vertices.}$$

Let

$$\mathcal{L}(n) = \text{ set of all increasing labelings, } L\text{, of trees in } \mathcal{B}(n).$$

Ex.
$\mathcal{B}(3)$:



Let

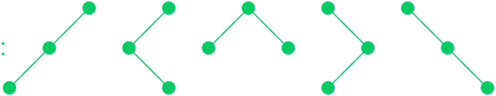
$$\mathcal{B}(n) = \text{ set of binary trees, } T\text{, on } n \text{ vertices.}$$

Let

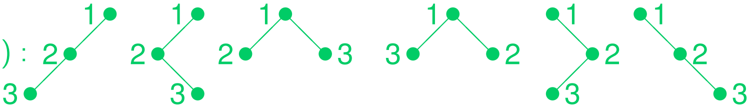
$$\mathcal{L}(n) = \text{ set of all increasing labelings, } L\text{, of trees in } \mathcal{B}(n).$$

Ex.

$\mathcal{B}(3)$:



$\mathcal{L}(3)$:

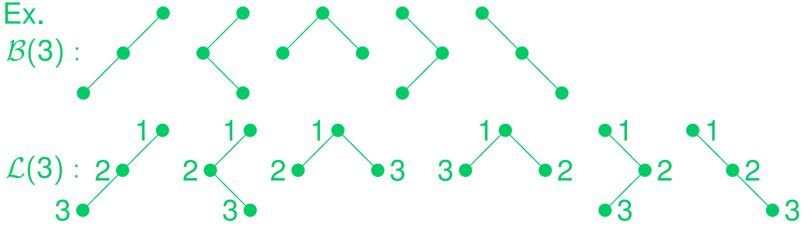


Let

$$\mathcal{B}(n) = \text{ set of binary trees, } T\text{, on } n \text{ vertices.}$$

Let

$$\mathcal{L}(n) = \text{ set of all increasing labelings, } L\text{, of trees in } \mathcal{B}(n).$$



Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

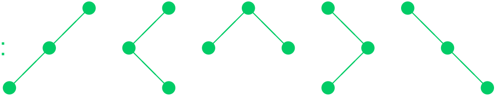
Let

$$\mathcal{B}(n) = \text{ set of binary trees, } T\text{, on } n \text{ vertices.}$$

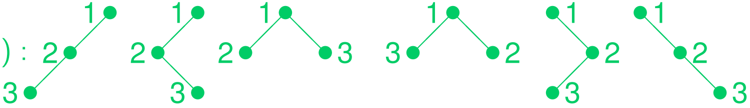
Let

$$\mathcal{L}(n) = \text{ set of all increasing labelings, } L\text{, of trees in } \mathcal{B}(n).$$

Ex.
$\mathcal{B}(3)$:



$\mathcal{L}(3)$:



Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

Notes.

1. The hooklengths appear as exponents.

Let

$$\mathcal{B}(n) = \text{ set of binary trees, } T\text{, on } n \text{ vertices.}$$

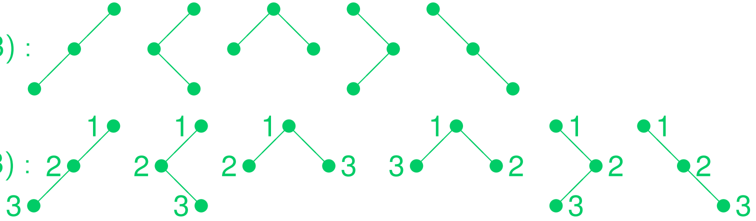
Let

$$\mathcal{L}(n) = \text{ set of all increasing labelings, } L\text{, of trees in } \mathcal{B}(n).$$

Ex.

$\mathcal{B}(3)$:

$\mathcal{L}(3)$:



Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

Notes.

1. The hooklengths appear as exponents.
2. Han's proof is algebraic. Our proof is probabilistic.

Outline

Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

Proof Multiplying the above equation by $n!$ and using the Hooklength Formula, it suffices to show

$$\sum_{T \in \mathcal{B}(n)} f^T \prod_{v \in T} \frac{1}{2^{h_v - 1}} = 1.$$

Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

Proof Multiplying the above equation by $n!$ and using the Hooklength Formula, it suffices to show

$$\sum_{T \in \mathcal{B}(n)} f^T \prod_{v \in T} \frac{1}{2^{h_v - 1}} = 1.$$

So it suffices to find an algorithm generating each $L \in \mathcal{L}(n)$ such that

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

Theorem (Han, 2008)

For any $n \geq 0$,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{h_v 2^{h_v - 1}} = \frac{1}{n!}.$$

Proof Multiplying the above equation by $n!$ and using the Hooklength Formula, it suffices to show

$$\sum_{T \in \mathcal{B}(n)} f^T \prod_{v \in T} \frac{1}{2^{h_v - 1}} = 1.$$

So it suffices to find an algorithm generating each $L \in \mathcal{L}(n)$ such that

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)} \text{prob}\, L = 1$.

(I) $\text{prob}\, L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)} \text{prob}\, L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-} w \text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

(I) $\text{prob}\, L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)} \text{prob}\, L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and $\text{prob}\, w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

L :

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

L :

prob $L = 1$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

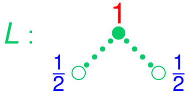
(II) $\sum_{L \in \mathcal{L}(n)} \text{prob } L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$



prob $L = 1$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

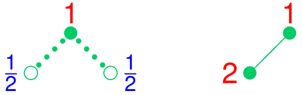
For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

L :



prob $L = 1$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)} \text{prob}\, L = 1$.

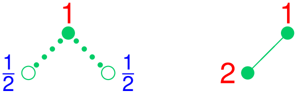
For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

L :



prob $L = 1 \quad \cdot \quad \frac{1}{2}$

(I) prob $L = \prod_{v\in T} 1/2^{h_v - 1}$ if L labels T, and

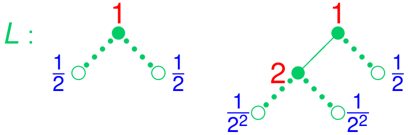
(II) $\sum_{L\in\mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$



prob $L = 1 \cdot \frac{1}{2}$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

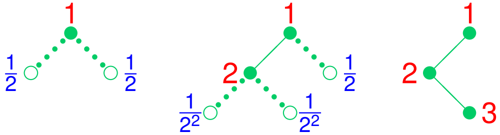
For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

L :



prob $L = 1 \cdot \frac{1}{2}$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

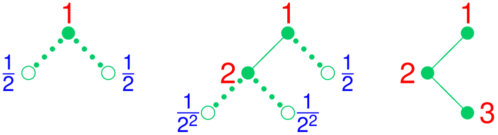
For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

L :



prob $L = 1 \quad \cdot \quad \frac{1}{2} \quad \cdot \quad \frac{1}{2^2}$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

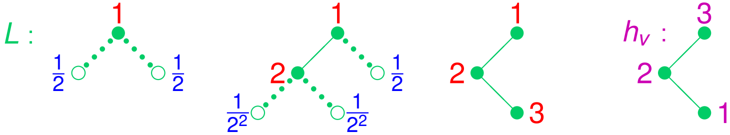
(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$



prob $L = 1 \quad \cdot \quad \frac{1}{2} \quad \cdot \quad \frac{1}{2^2}$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T, and

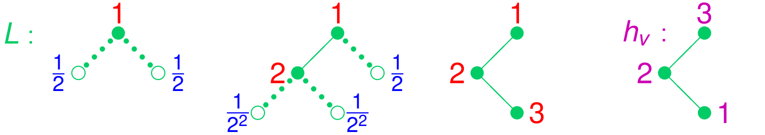
(II) $\sum_{L \in \mathcal{L}(n)}$ prob $L = 1$.

For $w \in T$, consider the *depth* of w:

$$d_w = \text{ length of the unique root-to-}w\text{ path.}$$

Algorithm. (a) Let L consist of a root labeled 1.
(b) While $|L| < n$, pick a leaf w to be added to L with label $|L| + 1$ and prob $w = 1/2^{d_w}$.
(c) Output L.

Ex. $n = 3$

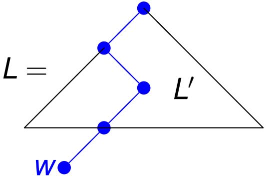


$$\text{prob } L = 1 \quad \cdot \quad \frac{1}{2} \quad \cdot \quad \frac{1}{2^2} \quad = \quad \prod_{v \in T} \frac{1}{2^{h_v - 1}}.$$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

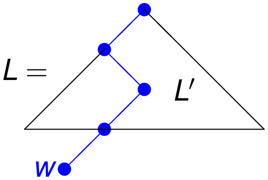
(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

Proof Let w be the node labeled n in L and let $L' = L - w$.



(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

Proof Let w be the node labeled n in L and let $L' = L - w$.

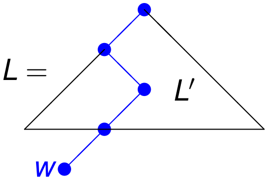


The hooklengths in L and L' are related by

$$h_v = \begin{cases} h'_v + 1 & \text{if } v \text{ is on the unique root-to-}w\text{ path } P, \\ h'_v & \text{else.} \end{cases}$$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

Proof Let w be the node labeled n in L and let $L' = L - w$.



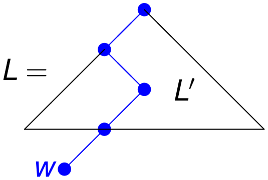
The hooklengths in L and L' are related by

$$h_v = \begin{cases} h'_v + 1 & \text{if } v \text{ is on the unique root-to-}w\text{ path } P, \\ h'_v & \text{else.} \end{cases}$$

Note that there are d_w vertices on $P \cap L'$.

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

Proof Let w be the node labeled n in L and let $L' = L - w$.



The hooklengths in L and L' are related by

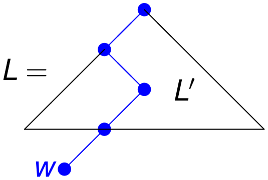
$$h_v = \begin{cases} h'_v + 1 & \text{if } v \text{ is on the unique root-to-}w \text{ path } P, \\ h'_v & \text{else.} \end{cases}$$

Note that there are d_w vertices on $P \cap L'$. So

$$\text{prob}\, L = \text{prob}\, w \cdot \text{prob}\, L'$$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

Proof Let w be the node labeled n in L and let $L' = L - w$.



The hooklengths in L and L' are related by

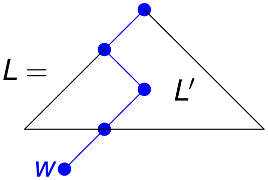
$$h_v = \begin{cases} h'_v + 1 & \text{if } v \text{ is on the unique root-to-}w \text{ path } P, \\ h'_v & \text{else.} \end{cases}$$

Note that there are d_w vertices on $P \cap L'$. So

$$\text{prob } L = \text{prob } w \cdot \text{prob } L' = \frac{1}{2^{d_w}} \prod_{v \in L'} \frac{1}{2^{h'_v - 1}}$$

(I) prob $L = \prod_{v \in T} 1/2^{h_v - 1}$ if L labels T.

Proof Let w be the node labeled n in L and let $L' = L - w$.



The hooklengths in L and L' are related by

$$h_v = \begin{cases} h'_v + 1 & \text{if } v \text{ is on the unique root-to-} w \text{ path } P, \\ h'_v & \text{else.} \end{cases}$$

Note that there are d_w vertices on $P \cap L'$. So

$$\text{prob}\, L = \text{prob}\, w \cdot \text{prob}\, L' = \frac{1}{2^{d_w}} \prod_{v \in L'} \frac{1}{2^{h'_v - 1}} = \prod_{v \in L} \frac{1}{2^{h_v - 1}}. \qquad \square$$

Outline

(a) Yang generalized Han’s formula to weighted ordered trees and a similar probabilistic proof works.

(a) Yang generalized Han's formula to weighted ordered trees and a similar probabilistic proof works. Let

$$\mathcal{O}(n) = \text{ set of ordered trees on } n \text{ vertices.}$$

(a) Yang generalized Han's formula to weighted ordered trees and a similar probabilistic proof works. Let

$$\mathcal{O}(n) = \text{ set of ordered trees on } n \text{ vertices.}$$

If m is a variable and c_v is the number of children of v in T, let

$$\mathrm{wt}(T) = \prod_{v \in T} \binom{m}{c_v}.$$

(a) Yang generalized Han's formula to weighted ordered trees and a similar probabilistic proof works. Let

$$\mathcal{O}(n) = \text{ set of ordered trees on } n \text{ vertices.}$$

If m is a variable and c_v is the number of children of v in T, let

$$\mathrm{wt}(T) = \prod_{v \in T} \binom{m}{c_v}.$$

Theorem (Yang, 2008)

For any n

$$\sum_{T \in \mathcal{O}(n)} \mathrm{wt}(T) \prod_{v \in T} \frac{1}{h_v m^{h_v - 1}} = \frac{1}{n!}.$$

□

(a) Yang generalized Han's formula to weighted ordered trees and a similar probabilistic proof works. Let

$$\mathcal{O}(n) = \text{ set of ordered trees on } n \text{ vertices.}$$

If m is a variable and c_v is the number of children of v in T, let

$$\mathrm{wt}(T) = \prod_{v \in T} \binom{m}{c_v}.$$

Theorem (Yang, 2008)

For any n

$$\sum_{T \in \mathcal{O}(n)} \mathrm{wt}(T) \prod_{v \in T} \frac{1}{h_v m^{h_v - 1}} = \frac{1}{n!}. \qquad \square$$

Note that if $m = 2$ then

$$\binom{m}{c_v} = \binom{2}{c_v} = \text{ \# of ways to make the children of } v \text{ binary.}$$

(a) Yang generalized Han's formula to weighted ordered trees and a similar probabilistic proof works. Let

$$\mathcal{O}(n) = \text{ set of ordered trees on } n \text{ vertices.}$$

If m is a variable and c_v is the number of children of v in T, let

$$\mathrm{wt}(T) = \prod_{v \in T} \binom{m}{c_v}.$$

Theorem (Yang, 2008)

For any n

$$\sum_{T \in \mathcal{O}(n)} \mathrm{wt}(T) \prod_{v \in T} \frac{1}{h_v m^{h_v - 1}} = \frac{1}{n!}. \qquad \square$$

Note that if $m = 2$ then

$$\binom{m}{c_v} = \binom{2}{c_v} = \text{ \# of ways to make the children of } v \text{ binary.}$$

So $\mathrm{wt}(T)$ becomes the number of ways to make T binary and Yang's result implies Han's.

(b) One can also generalize Han's formula and the probabilitic proof by considering n-vertex subtrees of a given infinite tree.

(b) One can also generalize Han's formula and the probabilitic proof by considering n-vertex subtrees of a given infinite tree.

(c) With Carla Savage, we are considering probabilistic proofs of q-hooklength formulas of Björner and Wachs and q, t-analogues of Novelli and Thibon.

(b) One can also generalize Han's formula and the probabilitic proof by considering n-vertex subtrees of a given infinite tree.

(c) With Carla Savage, we are considering probabilistic proofs of q-hooklength formulas of Björner and Wachs and q, t-analogues of Novelli and Thibon.

(d) Han also proved the following result.

Theorem (Han, 2008)

For any n,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{(2h_v + 1)2^{2h_v - 1}} = \frac{1}{(2n+1)!}. \qquad \square$$

(b) One can also generalize Han's formula and the probabilitic proof by considering n-vertex subtrees of a given infinite tree.

(c) With Carla Savage, we are considering probabilistic proofs of q-hooklength formulas of Björner and Wachs and q, t-analogues of Novelli and Thibon.

(d) Han also proved the following result.

Theorem (Han, 2008)

For any n,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{(2h_v + 1)2^{2h_v - 1}} = \frac{1}{(2n+1)!}.$$

□

Is there a probabilistic proof?

(b) One can also generalize Han's formula and the probabilitic proof by considering n-vertex subtrees of a given infinite tree.

(c) With Carla Savage, we are considering probabilistic proofs of q-hooklength formulas of Björner and Wachs and q, t-analogues of Novelli and Thibon.

(d) Han also proved the following result.

Theorem (Han, 2008)

For any n,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{(2h_v + 1)2^{2h_v - 1}} = \frac{1}{(2n+1)!}.$$

□

Is there a probabilistic proof? Note that if $\hat{T}$ is the completion of T, i.e., T with all possible leaves added, then

$$f^{\hat{T}} = \frac{(2n+1)!}{\prod_{v \in T}(2h_v + 1)}.$$

(b) One can also generalize Han's formula and the probabilitic proof by considering n-vertex subtrees of a given infinite tree.

(c) With Carla Savage, we are considering probabilistic proofs of q-hooklength formulas of Björner and Wachs and q, t-analogues of Novelli and Thibon.

(d) Han also proved the following result.

Theorem (Han, 2008)

For any n,

$$\sum_{T \in \mathcal{B}(n)} \prod_{v \in T} \frac{1}{(2h_v + 1)2^{2h_v - 1}} = \frac{1}{(2n+1)!}. \qquad \square$$

Is there a probabilistic proof? Note that if $\hat{T}$ is the completion of T, i.e., T with all possible leaves added, then

$$f^{\hat{T}} = \frac{(2n+1)!}{\prod_{v \in T}(2h_v + 1)}.$$

(e) What is the analogue for tableaux of Han's formulas?