1

Contents

2 Combinatorial Applications

2.1 Trees

Types of trees I

- A *free tree* is a connected graph with no cycles.
- A *rooted tree* is a free tree with a distinguished node called the *root*.
- An *ordered tree* is a rooted tree where the order of subtrees is important.
- A *binary tree* is an ordered tree where every node has 0 or 2 children.
- A *labelled tree* is a tree with n nodes such that each node is labelled by an element of $[n]$ and all labels are distinct.

Synonyms in literature: plane = ordered; oriented = rooted.

Types of trees II

- *Binary tree*: an *external node*, or an *internal node* n connected to an ordered pair L_n, R_n of binary trees.
- *Ordered tree*: a node connected to a sequence of ordered trees.
- *Binary search tree*: a binary tree with each internal node having a *key*, such that the key of each node is $\leq$ all keys in right subtree and $\geq$ all keys in left subtree.
- *Heap-ordered tree*: a binary tree such that the key of each node is $\geq$ the key of anything in its subtree.
- *Trie*:

Degree-restricted trees

- Let $0 \in \Omega \subseteq \mathbb{N}$. We consider the combinatorial class $\mathcal{T}_\Omega$ of ordered plane trees with the outdegree of each node restricted to belong to Ω.
- Examples: $\Omega = \{0, 1\}$ gives paths; $\Omega = \{0, 2\}$ gives binary trees; $\Omega = \{0, t\}$ gives t-ary trees; $\Omega = \mathbb{N}$ gives general ordered trees.

- Let $T_\Omega(z)$ be the enumerating GF of this class. The symbolic method immediately gives the equation

$$T_\Omega(z) = z\phi(T_\Omega(z))$$

where $\phi(x) = \sum_{\omega\in\Omega} x^\omega$.

- Lagrange inversion is tailor-made for this situation.

Ternary tree example

- Let $\mathcal{T}$ be the set of ordered ternary trees (similar definition to binary trees with 3 instead of 2). Then the counting (by external nodes) OGF $T(z)$ satisfies $T(z) = z(1 + T(z)^3)$.

- By Lagrange inversion we get

$$a_n = [z^n]T(z) = \frac{1}{n}[u^{n-1}]\left(1 + u^3\right)^n.$$

- By lookup we obtain

$$a_n = \begin{cases} \frac{1}{n}\binom{n}{k} & \text{if } n = 3k+1 \\ 0 & \text{otherwise.} \end{cases}$$

Path length in trees

- The bivariate generating function $F(z, u)$ enumerating binary trees by number of nodes and internal path length satisfies the equation

$$F(z, u) = 1 + zF(zu, u)^2.$$

- The mean and variance are given by a standard computation. Note that

$$F_u(z, u) = 2zF(zu, u)[F_u(zu, u) + zF_z(zu, u)]$$

and so $F_u(z, 1) = 2zF(z, 1)[F_u(z, 1) + zF_z(z, 1)]$. Thus

$$\mu_n := \frac{[z^n]\frac{zF_z(z,1)}{1-2zF(z,1)}}{[z^n]F(z, 1)}$$

- The mean μ_n is asymptotic to $\sqrt{\pi}n^{3/2}$, so the mean level of a node is of order $\sqrt{n}$. The variance is also of order $n^{3/2}$.

Labelled trees

- A properly labelled (unordered) tree is a connected acyclic graph with n vertices, each with one of the numbers $1, \ldots, n$.
- By symbolic method, the set $\mathcal{T}$ of rooted labelled unordered trees satisfies $\mathcal{T} = \{\bullet\} \times \mathit{set}(\mathcal{T})$ and so $T(z) = z \exp(T(z))$.
- Lagrange inversion gives

$$a_n = \frac{n!}{n}[y^{n-1}] \exp(ny) = \frac{n!}{n}[y^{n-1}] \sum_k (ny)^k/k! = n^{n-1}.$$

- Hence the number of labelled trees is n^{n-2}.

Binary search trees

- Suppose we insert n distinct keys into an initially empty BST. The uniform distribution on permutations of size n does not induce the uniform distribution on binary trees.
- The mean and variance (for a binary tree of size n taken uniformly at random) can be obtained by standard methods. Analysis is the same as for quicksort. The mean is of order $2n \log n$ and the variance of order $0.4203n^2$. So these trees tend to be shorter and fatter than general binary trees.
- The internal path length equals the construction cost of a binary search tree of size n; dividing by n gives the expected cost of a successful search.

Heap-ordered trees

- A heap-ordered tree (HOT) is a binary tree whose nodes are labelled in such a way that the labels along each path from the root are strictly increasing.
- There is a 1-1 correspondence between HOT's and permutations.

2.2 Strings

Hidden pattern occurrence

- The set of occurrences of a *subsequence* (hidden pattern) $-a-b-c-d-$ in an English text of length n corresponds to $\mathcal{A}^* a \mathcal{A}^* b \mathcal{A}^* c \mathcal{A}^* d \mathcal{A}^*$.
- The counting OGF of $\mathcal{A}^*$ is $1/(1-26z)$ so OGF for all pattern occurrences is $P(z) = z^4/(1-26z)^5$.

- Expected number of occurrences in random "word" of length n is $[z^n]P(z)/(26^n) = (26)^{-4}\binom{n}{4}$.
- The OGF for total occurrences of the *substring abcd* is $z^4/(1-26z)^2$ and a similar analysis applies.
- Note relevance to various conspiracy theories. *Every sufficiently long random text contains a given hidden message with high probability.*

Pattern avoidance

- We have already counted total *occurrences* of a given substring or pattern. Now we want to count *number of words* a_n not containing a given pattern (harder problem).
- A nice trick: let T be the position of the end of the first occurrence of the pattern, X_n the event that the first n bits of a random bitstring do not contain the pattern. Then $S(z) = \sum_{n\geq 0} a_n z^n$ implies that

$$S(1/2) = \sum_{n\geq 0} a_n/2^n = \sum_{n\geq 0} \Pr(X_n) = \sum_{n\geq 0} \Pr(T > n) = E[T].$$

A simple pattern

- Given substring $\sigma = 00\cdots 0$ of length k, let $S(z)$ be the counting OGF for bitstrings without σ as substring.
- Recursion/symbolic method gives

$$S(z) = \left(\sum_{i<k} z^i\right)(1 + zS(z))$$

so

$$S(z) = \frac{1 + z + z^2 + \cdots + z^{k-1}}{1 - z - z^2 - \cdots - z^k} = \frac{1 - z^k}{1 - 2z + z^{k+1}}.$$

- Asymptotics: $a_n \approx C\rho^{-n}$ where ρ is smallest modulus root of denominator.
- Note that $\rho = 1/2 + \rho^{k+1}/2$ and $0 < \rho < 1$. Thus $1/2 < \rho < 1/2 + 1/2^k$, etc, and we can compute ρ quickly by iteration.
- Note $S(1/2) = 2^{k+1} - 2$.

Substring patterns - autocorrelation polynomial

- Consider an arbitrary binary string $\sigma = \sigma_0\sigma_1 \cdots \sigma_{k-1}$ of length k.
- For $0 \leq j \leq 1$, shift σ right j places. Define $c_j = 1$ if the overlap matches the tail $\sigma^{(j)}$ of σ, $c_j = 0$ otherwise. The *autocorrelation polynomial* is $c(z) = \sum_j c_j z^j$.
- Let $\mathcal{S}$, (resp. $\mathcal{T}$) be the set of bitstrings not containing p (resp. containing it once at the end). Then

$$\mathcal{S} \cup \mathcal{T} \cong \{\epsilon\} \cup \mathcal{S} \times \{0, 1\}$$
$$\mathcal{S} \times \{\sigma\} \cong \mathcal{T} \times \cup_{\{j:c_j \neq 0\}} \sigma^{(j)}$$

  and the symbolic method gives $S(z) + T(z) = 1 + 2zS(z)$ and $S(z)z^k = T(z)c(z)$.
- Thus
$$S(z) = \frac{c(z)}{z^k + (1 - 2z)c(z)}.$$

Regular languages

- Rational GFs always arise from the *transfer matrix method*.
- Special case: the counting GF of an unambiguous regular language is rational (Chomsky-Schützenberger, 1963).
- Recall that every regular language can be defined by an unambiguous regular expression.
- Thus if we construct a combinatorial class iteratively using only disjoint union, cartesian product, and sequence, the counting GF is rational.

Regular expression example

- Consider language (over alphabet $\{a, b\}$) defined by $(bb \mid a(bb)^*aa \mid a(bb)^*(ab \mid ba)(bb)^*(ab \mid ba))^*$ (number of b's is even, number of a's divisible by 3).
- The symbolic method gives
$$S(z) = \frac{(1 - z^2)^2}{1 - 3z^2 - z^3 + 3z^4 - 3z^5 + z^6}.$$

  Hence $a_n \approx CA^n$, $A \cong 1.7998$.
- Need to check that the expression is unambiguous.

Tries

- Each binary tree corresponds to a set of binary strings (0 encodes left branch, 1 encodes right branch, string is given by labels on path to external node). This set of strings is *prefix-free*.
- Conversely a finite prefix-free set of strings corresponds to a unique binary tree, a *full trie*.
- More generally, we may stop branching as soon as the strings are all distinguished. This gives a *trie*, a binary tree such that all children of leaves are nonempty. Each string is stored in an external node but not all external nodes have strings. Can be described by symbolic method.
- A *Patricia trie* saves space, by collapsing one-way branches to a single node.
- Relevant parameters: number of internal nodes I_n; external path length L_n; height H_n.

Trie recurrences

We assume that a trie is built from n infinite random bitstrings. Each bit of each string is independently either 0 or 1.

- $$L_n = n + \frac{1}{2^n} \sum_k \binom{n}{k} (L_k + L_{n-k}),$$
  L the mean external path length.
- $$I_n = 1 + \frac{1}{2^n} \sum_k \binom{n}{k} (I_k + I_{n-k}),$$
  I the number of internal nodes.
- Let $L(z) = \sum_n L_n z^n/n!$, etc. Then
  $$L(z) = 2L(z/2)e^{z/2} + ze^z - z$$
  $$I(z) = 2I(z/2)e^{z/2} + e^z - z - 1.$$

Solving the trie recurrences, I

- If $\phi(z) = 2e^{z/2}\phi(z/2) + a(z)$, then by iteration we obtain
  $$\phi(z) = \sum_{j\geq 0} 2^j e^{z(1-2^{-j})} a(2^{-j}z).$$

- Thus we obtain

$$L_n = n \sum_{j \ge 0} \left(1 - \left(1 - 2^{-j}\right)^{n-1}\right)$$

$$I_n = \sum_{j \ge 0} 2^j \left[1 - \left(1 - 2^{-j}\right)^n - \frac{n}{2^j}\left(1 - 2^{-j}\right)^{n-1}\right].$$

- How to derive an asymptotic approximation? See Flaj-Sedg p211, p402 for elementary arguments. Answers: $L_n \sim n \lg n, I_n \sim n/\lg 2$. More precise answers are obtained by complex methods (*Mellin transform*).

Solving the trie recurrences, II

- Define $\hat{\phi}(z) = e^{-z}\phi(z)$, etc (this is the *Poisson transform*). Then we have

$$\hat{\phi}(z) = 2\hat{\phi}(z/2) + \hat{a}(z).$$

- Iteration yields

$$\hat{\phi}(z) = \sum_{j \ge 0} 2^j \hat{a}(2^{-j} z).$$

- This gives, on inverting the transform,

$$L_n = \sum_{k \ge 2} (-1)^k \binom{n}{k} \frac{k 2^{k-1}}{2^{k-1} - 1}.$$

- Asymptotics for such alternating sums can be obtained by *Rice's method*.

Summary: tries

- A useful data structure for dictionary and pattern matching. Also a mathematical model for many algorithms.
- Asymptotically optimal ($\lg n$) expected search cost.
- Space wastage: about 44% extra nodes ($1/\lg 2 - 1$).
- Recurrences under the infinite random bitstring model yield GF equations that are tricky. Solution involves infinite sums of functions.
- Explicit formulae for solutions are infinite sums. Mellin transforms or Rice's integrals give precise asymptotics; elementary methods can also be used.

2.3 Permutations

Basic definitions

- A *permutation* is a bijection $\pi : [n] \to [n]$ for some $n \in \mathbb{N}$. We write $n = |\pi|$. We can think of a permutation as a labelled sequence of atoms, so the counting EGF is $1/(1-z)$.
- Let Π be the set of all permutations, $\Pi_n := \{\pi \in \Pi \mid |\pi| = n\}$, $\Pi = \bigcup_n \pi_n$.
- For $n \geq 0$, adding the element $n+1$ in all possible places gives a bijection between $\Pi_n \times [n+1]$ and Π_{n+1}.
- An *inversion* in π is a pair $i < j$ with $\pi(i) > \pi(j)$. Let $\iota(p)$ be the number of inversions of π. Relevant to inversion sort performance.
- A *left-to-right minimum* in π is an index i with $\pi(j) > \pi(i)$ for all $j < i$. Let $\lambda(\pi)$ be the number of left-to-right minima in π. Relevant to selection sort performance.

Left-to-right minima

- Let $F(z,u) = \sum_{\pi\in\Pi} \frac{z^{|\pi|}}{|\pi|!} u^{\lambda(\pi)}$. We obtain the recurrence

$$\begin{aligned} F(z,u) &= u + \sum_{\pi'\in\Pi} u^{1+\lambda(\pi')} \frac{z^{|\pi'|+1}}{(|\pi'|+1)!} \\ &+ \sum_{\pi'\in\Pi, 1<k\leq n(\pi')+1} |\pi'| u^{\lambda(\pi')} \frac{z^{|\pi'|+1}}{(|\pi'|+1)!} \\ &= u + u\int F + \int zF_z \end{aligned}$$

  where integration is with respect to z. Thus $F_z(z,u) = uF(z,u) + zF_z(z,u)$.

- Solve this first order differential equation to get $F(z,u) = (1-z)^{-u}$. It follows that the expected number of left-to-right minima in a uniformly chosen permutation of size n is $H_n \approx \log n$. The variance is also of order $\log n$.

Inversions

- Let $F(z,u) = \sum_\pi u^{\iota(\pi)} z^{|\pi|}/|\pi|!$. We obtain the recurrence

$$\begin{aligned}
F(z,u) &= 1 + \sum_\pi \sum_{0 \le k \le |\pi|} u^{\iota(\pi)+|\pi|-k} \frac{z^{|\pi|+1}}{(|\pi|+1)!} \\
&= 1 + \sum_\pi \frac{1-u^{|\pi|+1}}{1-u} u^{\iota(\pi)} \frac{z^{|\pi|+1}}{(|\pi|+1)!} \\
&= 1 + \frac{A(z,u) - A(zu,u)}{1-u}
\end{aligned}$$

  where A is a z-antiderivative of F.

- Differentiate with respect to u, take limit as $u \to 1$ by L'Hopital. Get

$$F_u(z,1) = \frac{z^2}{2(1-z)^3}$$

  so mean number of involutions is $n(n-1)/4$. Similarly, the variance is of order n^3.

Random generation of permutations

- Given n, construct a permutation of $[n]$ uniformly at random.
- Standard method: Fisher-Yates (1938). Start with identity permutation. Iterate i from n down to 2; at each step choose j uniformly at random from $[i]$ and exchange the elements in positions i and j. Closely related to the *** method for enumeration.
- Example:
- It is easily shown that this works.