Math 343, Lecture 6

① Specifications

Def A combinatorial specification for the classes $\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}$ is a set of equations

$$\mathcal{A}^{(1)} = \Phi_1(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)})$$
$$\vdots$$
$$\mathcal{A}^{(r)} = \Phi_r(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)})$$

where the Φ_i are built from admissible operators along with $\mathcal{E}$ and $\mathcal{Z}$

eg We've seen many examples of combinatorial specifications already

Binary rooted trees

Dyck paths

also

Binary strings

eg None of those examples involved a system of equations, so consider the following

Consider a class of plane rooted trees with white and black vertices where

let $\mathcal{A}_{\circ}$ be

let $\mathcal{A}_{\bullet}$ be

let $\mathcal{A}$ be

Then

Def Let $\begin{cases} A^{(1)} = \Phi_1(A^{(1)}, \dots, A^{(r)}) \\ \vdots \\ A^{(r)} = \Phi_r(A^{(1)}, \dots, A^{(r)}) \end{cases}$ be a combinatorial specification.

The dependency digraph of the specification is eg

eg

eg

Def If a specification has

recursive

iterative

eg

Def An iterative specification that

regular
specification

This is similar to, but not the same as the notion of regular language from computer science

The problem is ambiguity.

For a regular language

eg $\{01, 0, 1\}^*$

So as a combinatorial class

$$Seq(z_0 \times z_1 + z_0 + z_1)$$

does not give the class of all binary strings

However

In fact this is true in general
every regular language has a regular combinatorial specification

but it takes some work to show the redundancy can always be removed see Flajolet + Sedgewick Appendix A.7

② Integer compositions

Another important class of examples of regular specifications is integer compositions

Def A composition of an integer n is

parts of the composition

eg Composition of 1 :

" 2 :

" 3 :

Let $\mathcal{C}$ be the combinatorial class of compositions where the size of a composition is what it is a composition of (eg $2+1+3$ is a composition of 6)

What is a specification for $\mathcal{C}$?

So $C_6 =$

so $C(x) =$

expanding $C(x) =$

Now we can restrict these.

eg let $\mathcal{C}$ be the class of compositions where the parts are all ≤ 5

Then $\mathcal{C} =$

so $C(x) =$

Now restrict the other way

eg let $\mathcal{C}$ be the class of compositions with at most 3 parts

$$\mathcal{C} =$$

so $C(x) =$

③ Computer explorations in Maple

Maple has a nice package combstruct

for working with combinatorial specifications

see the typed notes for examples and key commands

④ Next time

How to solve for the generating function

when its not quadratic – Lagrange inversion