LECTURE 5 – COMPOSITIONS, WORDS, TREES, AND MORE

JAY PANTONE

Compositions

A *composition* of size n is an ordered sequence of positive integers that sums to n. The length of a composition is the number of summands.

Example: $4+1+1$ and $1+1+4$ are distinct compositions of 6.

Let $\mathcal{C}$ be the combinatorial class of all compositions. By our definition, we see that

$$\mathcal{C} = \textsc{Seq}([\text{positive integers}]),$$

because the size of a sequence is by definition the sum of the sizes of its components. Therefore, to describe the class of all compositions using the symbolic method, we need to first describe the class of positive integers (where the size of an integer is its value).

Think of an integer as a sequence of identical dots of that length:

$$\begin{aligned} 1 &= \bullet \\ 2 &= \bullet\,\bullet \\ 3 &= \bullet\;\bullet\;\bullet \\ &\cdots \end{aligned}$$

Therefore, the class of positive integers can be described by

$$\mathcal{I}_{\geq 1} = \textsc{Seq}_{\geq 1}(\mathcal{Z}_\bullet).$$

From this we find

$$I_{\geq 1}(z) = \boxed{\frac{z}{1-z}} = z + z^2 + z^3 + \cdots.$$

Now, a composition is a (possibly empty) sequence of positive integers, so

$$\mathcal{C} = \textsc{Seq}(\mathcal{I}_{\geq 1}),$$

and thus

$$C(z) = \frac{1}{1 - \dfrac{z}{1-z}} = \boxed{\frac{1-z}{1-2z}} = 1 + z + 2z^2 + 4z^3 + \cdots.$$

This is easy enough that we can find a closed-form formula with what we already know.

$$\begin{aligned}[z^n]\frac{1-z}{1-2z} &= [z^n]\left(\frac{1}{1-2z} - \frac{z}{1-2z}\right)\\ &= [z^n]\frac{1}{1-2z} - [z^n]\frac{z}{1-2z}\\ &= 2^n - [z^{n-1}]\frac{1}{1-2z}\\ &= 2^n - 2^{n-1}\\ &= 2^{n-1}.\end{aligned}$$

This is typically proved directly in an undergraduate class using *stars-and-bars* or *balls-and-walls*.

The symbolic method allows for many variations without an increase in difficulty. We specialize $\mathcal{C}$ using a superscript in $(\cdot)$ to denote a restriction on the number of parts and in $\{\cdot\}$ to denote a restriction on the size of the parts

Example: Find the generating function for compositions of n into at least 3 parts. Well,

$$\mathcal{C}^{(\geq 3)} = \textsc{Seq}_{\geq 3}(\mathcal{I}_{\geq 1}),$$

and thus

$$C^{(\geq 3)}(z) = \frac{I_{\geq 1}(z)^3}{1 - I_{\geq 1}(z)} = \frac{\left(\dfrac{z}{1-z}\right)^3}{1 - \dfrac{z}{1-z}} = \boxed{\frac{z^3}{(1-z)^2(1-2z)}} = z^3 + 4z^4 + 11z^5 + \cdots .$$

Coefficient extraction will eventually give us $a_n = 2^{n-1} - n$.

Example: Find the generating function for compositions of n into parts of size at most r. The symbolic method gives

$$\mathcal{C}^{\{\leq r\}} = \textsc{Seq}(\mathcal{I}_{[1..r]})$$

and thus

$$C^{\{\leq r\}}(z) = \frac{1}{1 - (z + z^2 + \cdots + z^r)} = \frac{1}{1 - \dfrac{z(1-z^r)}{1-z}} = \boxed{\frac{1-z}{1-2z+z^{r+1}}.}$$

Coefficient extraction will give

$$c_n^{\{\leq r\}} = \sum_{j,k} (-1)^k \binom{j}{k} \binom{n-rk-1}{j-1},$$

but this doesn't give us any sense of asymptotic behavior (does it still grow like 2^n? Or slower? Is it exponential or polynomial?) Asymptotic analysis will tell us

$$c_n^{\{\leq r\}} \sim K_r \mu_r^n,$$

for $K_r \geq 0$ and $1 \leq \mu_r < 2$.

$r =$	1	2	3	4	5	10
$\mu_r \approx$	1	1.618	1.839	1.928	1.966	1.999

Example: Find the generating function for the class of compositions in which the parts alternate even/odd (can start with either). See if you can figure out why

$$\mathcal{C}^{\{e/o\}} = \mathcal{E} + \mathcal{I}_{\geq 1} + \textsc{Seq}_{\geq 1}(\mathcal{I}_{\text{even}} \times \mathcal{I}_{\text{odd}}) \times (\mathcal{E} + \mathcal{I}_{\text{even}}) + \textsc{Seq}_{\geq 1}(\mathcal{I}_{\text{odd}} \times \mathcal{I}_{\text{even}}) \times (\mathcal{E} + \mathcal{I}_{\text{odd}}).$$

We therefore derive

$$\begin{aligned} C^{\{e/o\}}(z) &= 1 + \frac{z}{1-z} + \frac{\frac{z^2}{1-z^2}\frac{z}{1-z^2}}{1 - \frac{z^2}{1-z^2}\frac{z}{1-z^2}}\left(1 + \frac{z^2}{1-z^2}\right) + \frac{\frac{z}{1-z^2}\frac{z^2}{1-z^2}}{1 - \frac{z}{1-z^2}\frac{z^2}{1-z^2}}\left(1 + \frac{z}{1-z^2}\right) \\ &= \boxed{\frac{1+z-z^2}{1-2z^2-z^3+z^4}} \\ &= 1 + z + z^2 + 3z^3 + 2z^4 + 6z^5 + \cdots. \end{aligned}$$

A closed form can be found by coefficient extraction but it's probably messy. The asymptotic form is:

$$c_n^{\{e/o\}} \sim K \cdot (1.490\ldots)^n$$

After all of this, you might ask what we can do with partitions? At this point, we run into the problem that compositions are ordered sums (i.e., sequences of integers) and partitions are unordered sums (i.e., multisets of integers), and we don't (yet!) have a multiset construction.

Words

An *alphabet* A is a finite set of symbols (called *letters*), such as $A = \{0, 1\}$ or $A = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$. A *word* over an alphabet A is an ordered sequence of letters from A. For example 100110 is a word of length 6 over the alphabet $\{0, 1\}$.

Let $\mathcal{B}$ be the class of binary words (those over $\{0, 1\}$). Then, we can express $\mathcal{B}$ as

$$\mathcal{B} = \textsc{Seq}(\mathcal{Z}_{\mathsf{a}} + \mathcal{Z}_{\mathsf{b}}),$$

which allows us to recover the known generating function

$$B(z) = \frac{1}{1-(z+z)} = \boxed{\frac{1}{1-2z}}.$$

Furthermore, for an alphabet $A = \{\mathsf{a}_1, \mathsf{a}_2, \ldots, \mathsf{a}_m\}$, the class of words over A is

$$\mathcal{W} = \textsc{Seq}(\mathcal{Z}_{\mathsf{a}_1} + \mathcal{Z}_{\mathsf{a}_2} + \cdots + \mathcal{Z}_{\mathsf{a}_m}),$$

and has generating function

$$W(z) = \boxed{\frac{1}{1-mz}},$$

yielding the already-known closed form

$$w_n = m^n.$$

As with compositions, we can use the symbolic method to describe classes of words with several types of restrictions. In each case, we have to be careful that we're only counting each word once. For example, the construction

$$\textsc{Seq}(1100 + 11 + 0)$$

generates the string 1100 two different ways, which causes a problem with the enumeration.

Every binary word can be uniquely decomposed as an initial sequences of 0's, followed by a sequence of blocks of the form $10 \cdots 0$. Thus,

$$\mathcal{B} = \textsc{Seq}(0)\ \textsc{Seq}(1\ \textsc{Seq}(0)),$$

(we use this shorter notation from now on) and so we recover

$$B(z) = \frac{1}{1-z} \cdot \frac{1}{1 - \dfrac{z}{1-z}} = \boxed{\frac{1}{1-2z}}.$$

This decomposition can be modified to count all binary words that avoid the pattern 11:

$$\mathcal{W}^{\{\text{no }11\}} = \textsc{Seq}(0)\ \textsc{Seq}(1\ \textsc{Seq}_{\geq 1}(0))(\epsilon + 1),$$

therefore

$$W^{\{\text{no }11\}}(z) = \frac{1}{1-z} \cdot \frac{1}{1 - \dfrac{z^2}{1-z}} \cdot (1+z) = \boxed{\frac{1+z}{1-z-z^2}},$$

giving the Fibonacci numbers. Actually, we can find an easier description:

$$\mathcal{W}^{\{\text{no }11\}} = (\epsilon + 1)\ \textsc{Seq}(0\ (\epsilon + 1)),$$

which gives the same $W(z)$. This extends naturally to class of words avoiding k consecutive 1s:

$$\mathcal{W}^{\{<k\ 1\text{'s}\}} = \textsc{Seq}_{<k}(1)\ \textsc{Seq}(0\ \textsc{Seq}_{<k}(1)),$$

giving

$$W^{\{<k\ 1\text{'s}\}}(z) = \frac{1-z^k}{1-z} \cdot \frac{1}{1 - \dfrac{z(1-z^k)}{1-z}} = \boxed{\frac{1-z^k}{1-2z+z^{k+1}}}.$$

Set Partitions

Give a finite set S, a *set partition* of S is a splitting of the set S into subsets called *blocks* of the partition. Every element of S must be in exactly one block, and neither the order of the blocks nor the order of the elements within each block matters. For example, one set partition of the set $S = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$ is

$$\{\{\clubsuit, \heartsuit\}, \{\diamondsuit\}, \{\spadesuit\}\},$$

and this is the same set partition as

$$\{\{\spadesuit\}, \{\heartsuit, \clubsuit\}, \{\diamondsuit\}\}.$$

Set partitions are often studied in introductory combinatorics courses, in which the *Stirling numbers of the second kind* are defined to be the number of set partitions of n distinct elements into r non-empty blocks, and are denoted $\left\{ n \atop r \right\}$.

We will now fix r and find a generating function for the sequence $s_n^{(r)} = \left\{ n \atop r \right\}$.

The first issue we address is that each set partition should be described in a unique way. To accomplish this, put an arbitrary total order on the set S. Define the canonical representation of a particular set partition to be that in which each block is listed in increasing order, and the blocks are then listed in increasing order by the minimum element.

For example, the canonical representation of the set partition $\{\{3,7\},\{1,5,2\},\{4,8,6\}\}$ is

$$\{\{1,2,5\},\{3,7\},\{4,6,8\}\}.$$

Now, each set partition of n into r parts can be bijectively associated to a word of length n over the letters $\{\ell_1, \ell_2, \ldots, \ell_r\}$ such that

(1) each letter is used at least once,

(2) for all i, the first occurrence of ℓ_i is before the first occurrence of ℓ_{i+1}.

If the ith letter of a word w is ℓ_j, this corresponds to putting the ith element in the set (now ordered by a total order) into the jth block. Rule (1) above ensures that every block is nonempty, and rule (2) requires that the blocks are indeed sorted in increasing order by their minimum entry.

The example $\{\{1,2,5\},\{3,7\},\{4,6,8\}\}$ is generated by the word 11231323.

The specification for the words over an r-letter alphabet with the 2 restrictions above is

$$\mathcal{S}^{(r)} = \ell_1 \,\mathrm{SEQ}(\ell_1)\, \ell_2 \,\mathrm{SEQ}(\ell_1+\ell_2)\, \ell_3 \,\mathrm{SEQ}(\ell_1+\ell_2+\ell_3) \,\cdots\, \ell_r \,\mathrm{SEQ}(\ell_1+\cdots+\ell_r),$$

from which we find the generating function

$$S^{(r)}(z) = \boxed{\frac{z^r}{(1-z)(1-2z)(1-3z)\cdots(1-rz)}}.$$

A simple coefficient extraction gives the closed form

$$s_n^{(r)} = \frac{1}{r!}\sum_{j=1}^{r}(-1)^{r-j}\binom{r}{j} j^n.$$

Dyck Paths

So far, all of our constructions have been *iterative* in nature; that is, the class on the left-hand side of the construction did not appear on the right-hand side as well. In fact, there is no reason we can't the class being defined as a building block itself on the right-hand side, so long as we are careful that our specification is well-defined. This type of definition is called *recursive*.

Remember that the positive integers $\mathcal{I}_{\geq 1}$ were defined by the iterative specification

$$\mathcal{I}_{\geq 1} = \textsc{Seq}(\mathcal{Z}_\bullet).$$

However, recognizing that every positive integer is either equal to 1 or is exactly one bigger than another positive integer, we could have used the recursive specification

$$\mathcal{I}_{\geq 1} = \mathcal{Z}_\bullet + \mathcal{Z}_\bullet \times \mathcal{I}_{\geq 1}.$$

On the level of generating functions, this translate to the functional equation

$$I_{\geq 1}(z) = z + zI_{\geq 1}(z),$$

which gives the solution

$$I_{\geq 1}(z) = \boxed{\frac{z}{1-z}}.$$

This is a well-defined recursive specification because the $\mathcal{I}_{\geq 1}$ on the right-hand side is in a cartesian product with something else, namely $\mathcal{Z}_\bullet$. Thus, we're defining the elements of $\mathcal{I}_{\geq 1}$ of size n in terms of the elements of size $n-1$. In this sense, the definition is not really circular. Specifications that are not well-defined will typically lead to non-sensical generating functions, such as

$$\mathcal{C} = \mathcal{E} + \mathcal{C} \times \mathcal{C}$$

leading to the functional equation

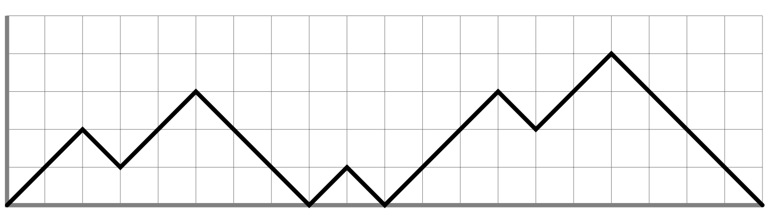
$$C(z) = 1 + C(z)^2$$

which has no formal power series solution.

Exercise: Explain why $\mathcal{C} = \mathcal{Z} + \mathcal{C} \times \mathcal{C}$ is a well-defined recursive construction.

Definition: A *Dyck path* of length n is a walk in the plane from $(0,0)$ to $(2n,0)$ that takes only steps of size $(1,1)$ and $(1,-1)$ and does not pass below the x-axis.

Example: Below is a Dyck path of length 20.



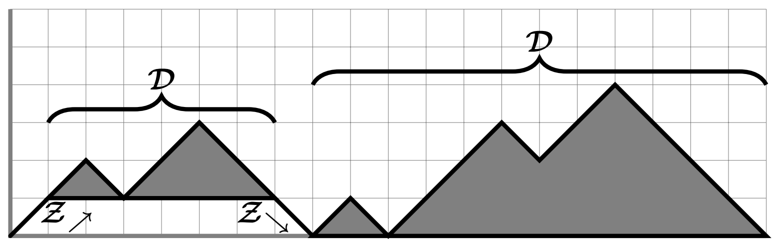
Exercise: Before we proceed further, consider the following issue. It is clear that every non-empty Dyck path can be described as a non-empty sequence of Dyck paths. So why does the construction

$$\mathcal{D} = \mathcal{E} + \textsc{Seq}_{\geq 1}(\mathcal{D})$$

fail?

There are two natural ways to decompose Dyck paths: *first return*, and *arch decomposition*.

(1) The *first return* decomposition splits every non-empty Dyck path around the first point where it returns to the x-axis. To the left of that point is a non-empty Dyck path that does not return to the origin and to the right lies a (possibly empty) arbitrary Dyck path.



As this uniquely describes every Dyck path, we find the recursive construction

$$\mathcal{D} = \mathcal{E} + (\mathcal{Z}_{\nearrow} \times \mathcal{D} \times \mathcal{Z}_{\searrow}) \times \mathcal{D}.$$

Therefore

$$D(z) = 1 + z^2(D(z))^2.$$

This implicitly defines $D(z)$ with minimal polynomial

$$0 = 1 - D(z) + z^2(D(z))^2.$$

We can solve explicitly using the quadratic formula

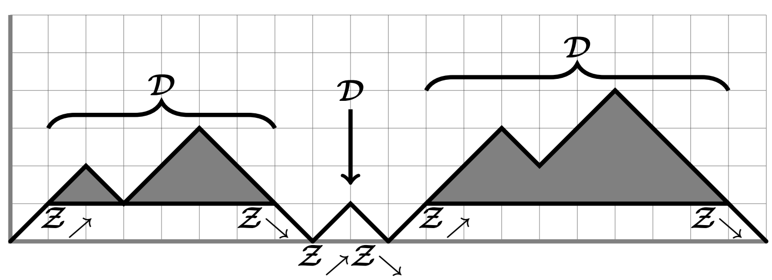
$$D(z) = \frac{1 \pm \sqrt{1-4z^2}}{2z^2}.$$

As the expression taken with the $+$ is not a formal power series[1], we determine that

$$D(z) = \boxed{\frac{1 - \sqrt{1-4z^2}}{2z^2}} = 1 + z^2 + 2z^4 + 5z^6 + \cdots .$$

Note that $d_{2n+1} = 0$ for all n and d_{2n} is the nth Catalan number.

(2) The *arch decomposition* splits every non-empty Dyck path into a non-empty sequence of non-empty Dyck paths that do not return to the x-axis until their final step. (This restriction corrects the ambiguity in the exercise above.)



[1]In fact it is a Laurent series given by $z^{-1} - 1 - z - 2z^2 - \cdots$. For now, one just needs to recognize that substituting $z = 0$ into the expression with the $+$ sign yields $2/0$, ruling out the required property that $\lim_{z\to 0} D(z) = 1$.

From this decomposition we find the alternative construction

$$\mathcal{D} = \textsc{Seq}(\mathcal{Z}_{\nearrow} \times \mathcal{D} \times \mathcal{Z}_{\searrow}),$$

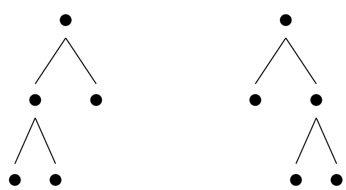
which of course yields the same generating function $D(z)$.

Dyck paths are generalizable in a number of ways that will be considered in the homework assignment (e.g., different step sets, less restrictive ending location, etc).

Trees

A *tree* is a connected graph without cycles. A *rooted* tree is one in which a particular vertex is chosen to be the *root*. The *children* of a node are those nodes that are adjacent to the node and further from the root than the node. A *leaf* is a node without children.[2]

A tree is called *planar* if the order of its children matters. For example, the two trees shown below are considered distinct as planar trees but equal as non-planar trees.



The size of a tree is considered to be either the total number of nodes, the number of internal nodes (non-leaves), or the number of external nodes (leaves).

First we set up a specification for the class $\mathcal{G}$ of all non-empty planar trees, where the size of a tree is the total number of nodes. Each non-empty tree can be decomposed as the root vertex and a possibly-empty ordered sequence of planar trees making up the children of the root. Therefore,

$$\mathcal{G} = \mathcal{Z} \times \textsc{Seq}(\mathcal{G}),$$

which leads to

$$G(z) = \frac{z}{1 - G(z)},$$

and so again we find the generating function of the Catalan numbers

$$G(z) = \boxed{\frac{1 - \sqrt{1 - 4z}}{2}} = z + z^2 + 2z^3 + \cdots .$$

Consider next the class $\mathcal{B}$ of planar trees such that each node has either 0 or 2 children. These are called *binary* trees. First, let the size of a tree be the number of internal nodes. Then,

$$\mathcal{B} = \mathcal{E} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}.$$

[2] Like all serious combinatorialists, we draw our trees with the root on top and the leaves on bottom.

Due to its recursive nature, the $\mathcal{E}$ in this construction does not just represent the case of a tree with a single node (which has size 0 by our definition), but also that any leaves will have size 0.

From the construction, we see that

$$B(z) = 1 + zB(z)^2,$$

and thus

$$B(z) = \boxed{\frac{1-\sqrt{1-4z}}{2z}} = 1 + z + 2z^2 + 5z^3 + \cdots .$$

What if instead we want the size of a tree to be the total number of nodes? Then,

$$\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B},$$

and we find

$$B(z) = \boxed{\frac{1-\sqrt{1-4z^2}}{2z}} = z + z^3 + 2z^5 + 5z^7 + \cdots .$$

What if the size is the number of leaves? Then,

$$\mathcal{B} = \mathcal{Z} + \mathcal{E} \times \mathcal{B} \times \mathcal{B},$$

and so

$$B(z) = \boxed{\frac{1-\sqrt{1-4z}}{2}} = z + z^2 + 2z^3 + 5z^4 + \cdots .$$

Let Ω be a subset of the positive integers that includes 0 and suppose we want to construct the class of all planar trees such that the number of children of any node is a number in Ω. If the size of a tree is the total number of nodes, then

$$\mathcal{T} = \mathcal{Z} \times \textsc{Seq}_{\Omega}(\mathcal{T}),$$

giving a functional equation

$$T(z) = z \sum_{\omega \in \Omega} T(z)^{\omega}.$$

For most Ω, this functional equation cannot be solved explicitly. However, we will learn a technique called *Lagrange inversion* that allows us to extract coefficients.

As with the other classes of combinatorial objects, we can use the principals described above to build more complicated objects.

Example: Let $\mathcal{T}$ be the class of planar trees where the nodes are colored either **red** or **blue**, the root is always **red**, all **red** nodes have an even number of children which must all be **blue**, and all **blue** nodes have an odd number of children which must all be **red**. Let the size of a tree be the total number of nodes. We derive the specification

$$\mathcal{T} = 1 + \mathcal{R},$$
$$\mathcal{B} = \mathcal{Z}_{\textbf{blue}} \times \textsc{Seq}_{\text{odd}}(\mathcal{R}),$$
$$\mathcal{R} = \mathcal{Z}_{\textbf{red}} \times \textsc{Seq}_{\text{even}}(\mathcal{B}).$$

Converting to functional equations, we see

$$T(z) = 1 + R(z),$$
$$B(z) = \frac{zR(z)}{1 - R(z)^2},$$
$$R(z) = \frac{z}{1 - B(z)^2}.$$

To solve, we see that

$$R(z) = \frac{z}{1 - B(z)^2} = \frac{z}{1 - \left(\dfrac{zR(z)}{1 - R(z)^2}\right)^2}$$

and so

$$0 = R(z)^5 - zR(z)^4 - (z^2+2)R(z)^3 + 2zR(z)^2 + R(z) - z.$$

Since $T(z) - 1 = R(z)$, we have

$$0 = (T(z)-1)^5 - z(T(z)-1)^4 - (z^2+2)(T(z)-1)^3 + 2z(T(z)-1)^2 + (T(z)-1) - z.$$

Expanding,

$$\boxed{0 = T(z)^5 - (z+5)T(z)^4 - (z^2 - 4z - 8)T(z)^3 + (3z+2)(z-2)T(z)^2 - 3z^2T(z) + z^2}.$$

Since this functional equation cannot be solved explicitly, let's take a quick detour and see how we can use Maple to calculate the series expansion of $T(z)$.

Maple

```
> solve(R = z/(1-(z*R/(1-R^2))^2),R);
         5       4      2        3         2
RootOf(_Z  - z _Z  + (-z  - 2) _Z  + 2 z _Z  + _Z - z),

             5       4      2        3         2
    RootOf(_Z  - z _Z  + (-z  - 2) _Z  + 2 z _Z  + _Z - z)

> R_expr := R^5 - z*R^4 - (z^2+2)*R^3 + 2*z*R^2 + R - z:

> T_expr := subs(R=T-1,R_expr);
                     5             4      2              3               2
T_expr := (T - 1)  - z (T - 1)  - (z  + 2) (T - 1)  + 2 z (T - 1)  + T - 1 - z

> T_expr := expand(subs(R=T-1,R_expr));
           5    4       3  2      4       3         2  2       3       2           2
T_expr := T  - T  z - T  z  - 5 T  + 4 T  z + 3 T  z  + 8 T  - 4 T  z - 3 T z

          2    2
     - 4 T  + z
```

Maple (continued)

```
> T_expr := collect(T_expr, T, factor);
T_expr :=

     5             4      2              3                       2      2      2
    T  + (-z - 5) T  + (-z  + 4 z + 8) T  + (3 z + 2) (z - 2) T  - 3 z  T + z

> root_of := RootOf(T_expr,T);
                    5             4      2              3
root_of := RootOf(_Z  + (-z - 5) _Z  + (-z  + 4 z + 8) _Z

          2              2      2       2
     + (3 z  - 4 z - 4) _Z  - 3 z  _Z + z )

> series(root_of,z,10);
                        2           3   /          17 \  4   /35          \  5
%1 z + (-1/2 %1 - 1/8) z  + 3/8 %1 z  + |-1/4 %1 + ---| z  + |--- %1 - 1/4| z
                                        \          128/      \128         /

       /  17      359 \  6   /1071         \  7   /  473      27077\  8
     + |- -- %1 + ----| z  + |---- %1 - 1/2| z  + |- --- %1 + -----| z  +
       \  32      1024/      \1024         /      \  256      32768/

    /102531         \  9      10
    |------ %1 - 3/2| z  + O(z  )
    \32768          /

                 2
%1 := RootOf(4 _Z  - 1)

> simplify(convert(%,radical));
             2        3          4   29   5    87   6    47   7   3195   8
1/2 z - 3/8 z  + 3/16 z  + 1/128 z  - --- z  + ---- z  + ---- z  - ----- z  +
                                      256      1024      2048      32768

    4227   9      10
    ----- z  + O(z  )
    65536

> root_of := RootOf(T_expr,T,index=2);
                    5             4      2              3
root_of := RootOf(_Z  + (-z - 5) _Z  + (-z  + 4 z + 8) _Z

          2              2      2       2
     + (3 z  - 4 z - 4) _Z  - 3 z  _Z + z , index = 2)

> simplify(convert(series(root_of,z,10),radical));
                2        3   33   4   99   5   631   6   2095  7   57349  8
 - 1/2 z + 1/8 z  - 3/16 z  + --- z  - --- z  + ---- z  - ---- z  + ----- z
                              128      256      1024      2048      32768

       200835  9      10
     - ------ z  + O(z  )
       65536

> root_of := RootOf(T_expr,T,index=3);
                    5             4      2              3
root_of := RootOf(_Z  + (-z - 5) _Z  + (-z  + 4 z + 8) _Z

          2              2      2       2
     + (3 z  - 4 z - 4) _Z  - 3 z  _Z + z , index = 3)
```

Maple (continued)

```
> simplify(convert(series(root_of,z,10),radical));
                               5      7      9      10
                     1 + z + z  + 2 z  + 6 z  + O(z  )

> coeff(simplify(convert(series(root_of,z,110),radical)),z,100);
                                    0

> coeff(simplify(convert(series(root_of,z,110),radical)),z,103);
                      17343442720761458606l351407294
```

Let us mention explicitly the Maple command `RootOf` used above. When you wish to find the series expansion of an generating function $f(z)$ defined *implicitly* by the minimal polynomial $P(z, f(z))$, you should not use the `solve` command. Instead, use

```
RootOf(P(z,f),f,index=N),
```

where index ranges from 1 to $\deg_f(P)$. One of these will be the right series, which you can confirm checking the initial terms. Sometimes the series command will produce terms with smaller `RootOf` expressions that themselves evaluate to real (or complex) numbers. To force this evaluate, wrap the series in

```
convert([series], radical)
```

and then `simplify`.

One last note about Maple: as a way to save space, Maple will use placeholder markers such as the `%1` seen in the code above which it them defines at the bottom of the output for the command.

The reader has likely noticed that every class of trees enumerated in this section was planar. This is because the order of the children of a given node does not matter in a non-planar tree, which means the sequence construction won't work. In order to build a non-planar tree, we must have a construction that builds *sets* of objects, rather than *sequences*. This is one construction, among others, that we define in the next lecture.