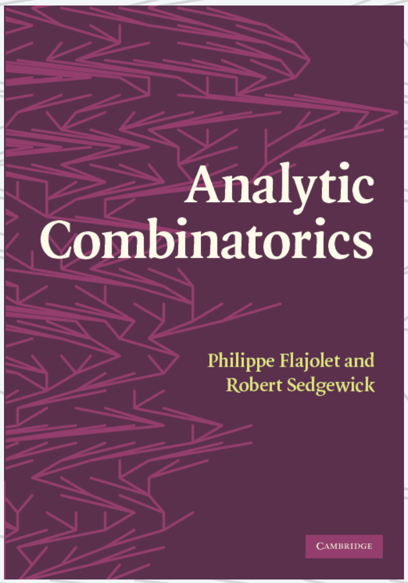
ANALYTIC COMBINATORICS

PART TWO



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

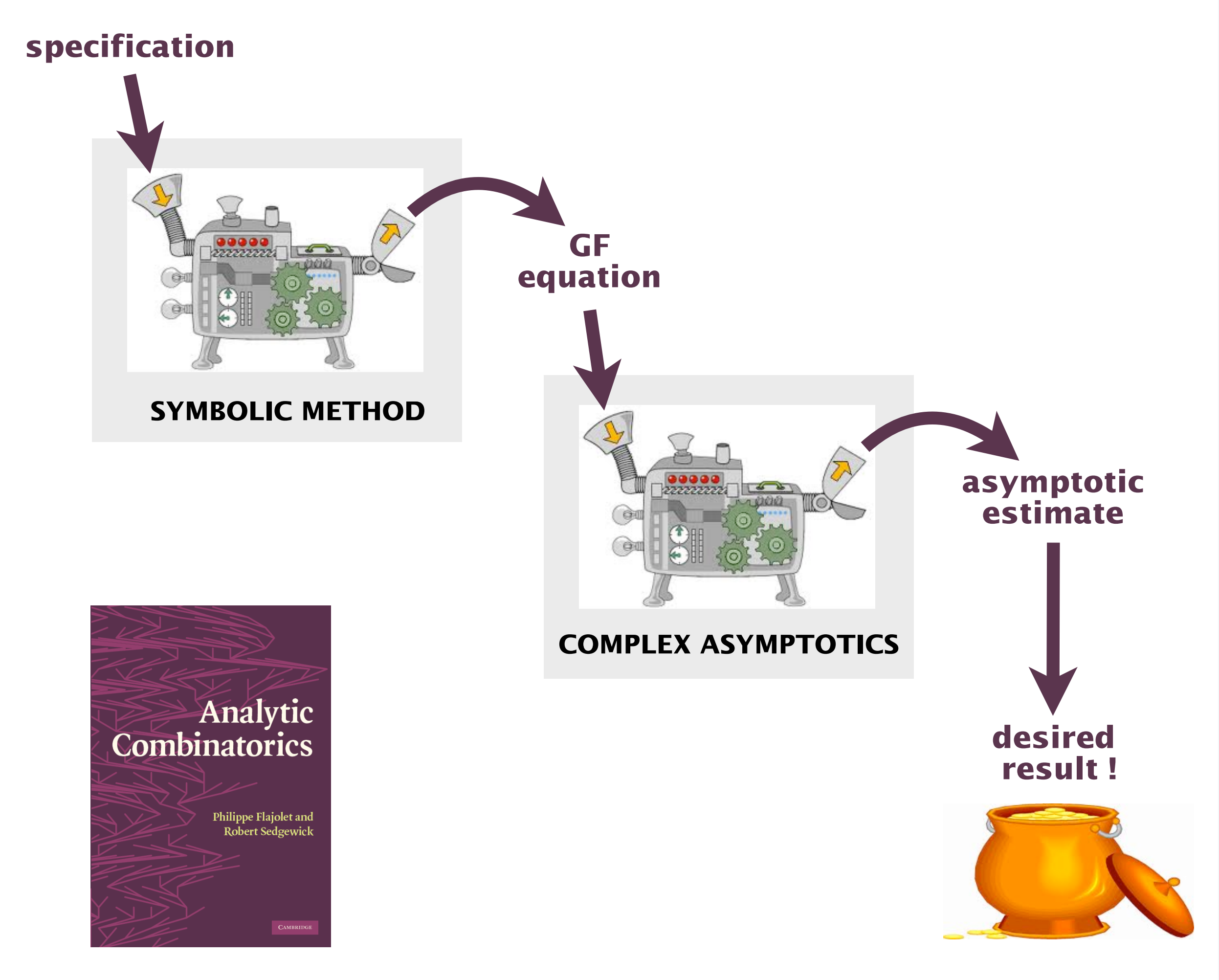
Analytic combinatorics overview

A. SYMBOLIC METHOD

1. OGFs
2. EGFs
3. MGFs

B. COMPLEX ASYMPTOTICS

4. Rational & Meromorphic
5. Applications of R&M
6. Singularity Analysis
7. Applications of SA
8. Saddle point



Attention: Much of this lecture is a *quick review* of material in *Analytic Combinatorics, Part I*

One consequence: it is a bit longer than usual

To: Students who took *Analytic Combinatorics, Part I*

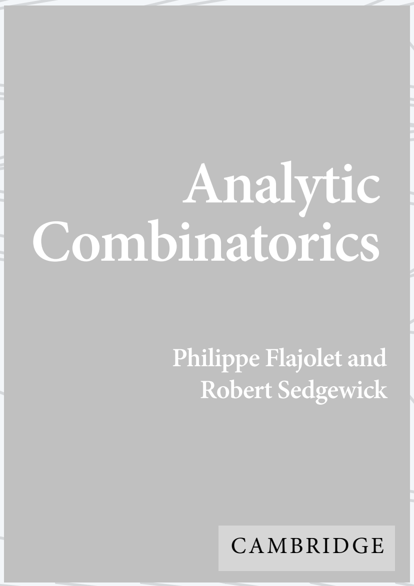
Bored because you understand it all?

GREAT! Skip to the section on labelled trees and do the exercises.

To: Students starting with *Analytic Combinatorics, Part II*

Moving too fast? Want to see details and motivating applications?

No problem, watch Lectures 5, 7, and 9 in Part I.



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

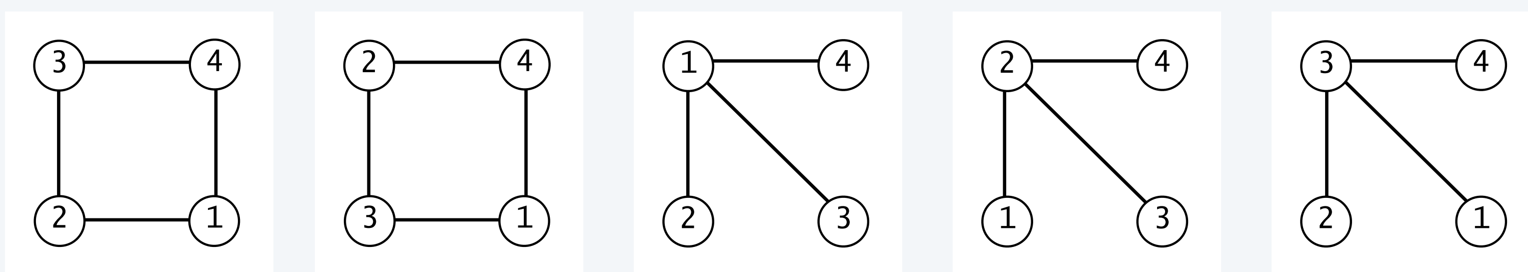
- **Basics**
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- Mappings

Labelled combinatorial classes

have objects composed of N atoms, labelled with the integers 1 through N.

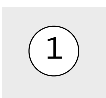
Ex. Different unlabelled objects

Ex. Different labelled objects

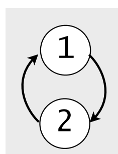


Labelled class example: cycles

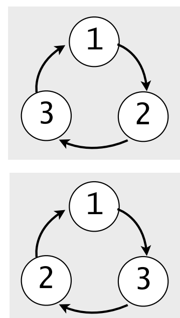
Q. How many *cycles* of labelled atoms?



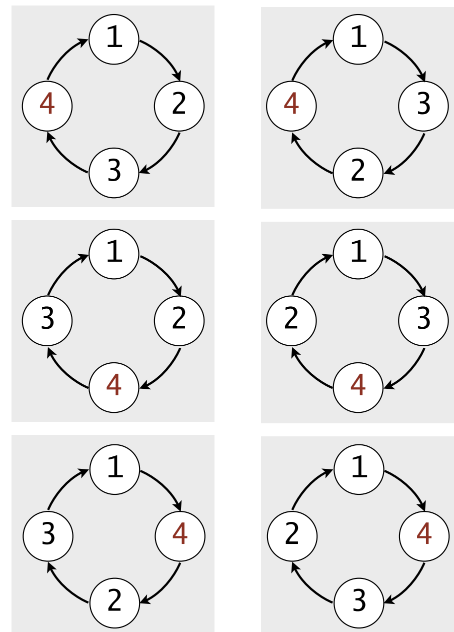
$Y_1 = 1$



$Y_2 = 1$



$Y_3 = 2$

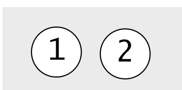


$Y_4 = 6$

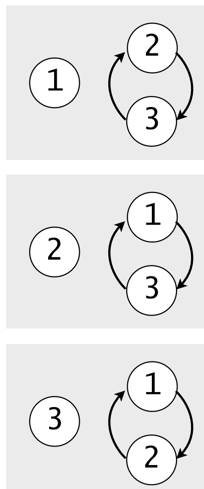
A. $(N-1)!$

Labelled class example 2: pairs of cycles

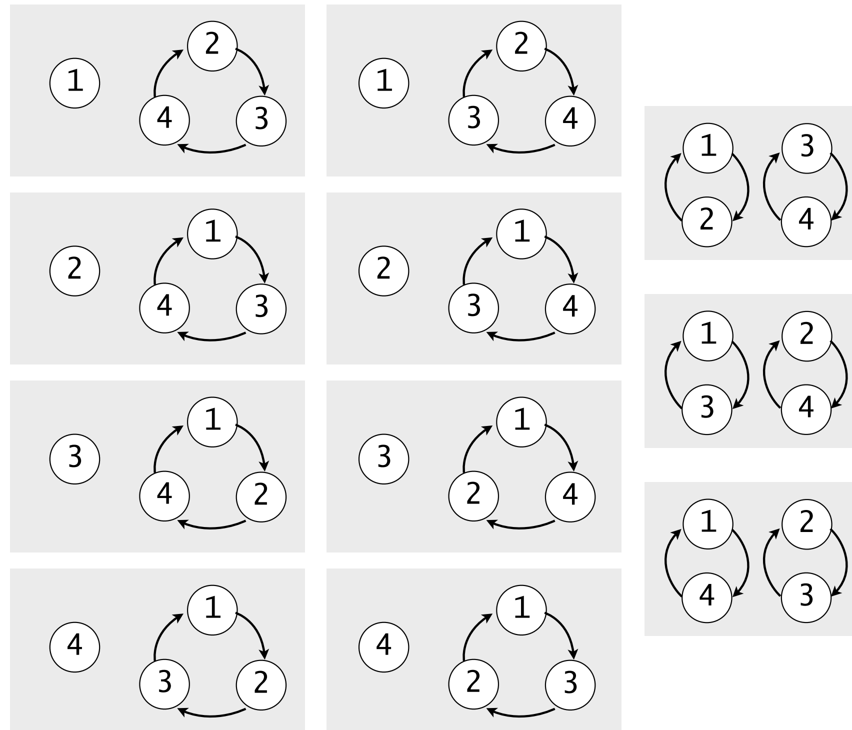
Q. How many *unordered pairs* of labeled cycles of size N?



$X_2 = 1$



$X_3 = 3$



$X_4 = 11$

A. $\left[{N \atop 2}\right]$ (Stirling numbers of the first kind.)

stay tuned (next lecture)

Basic definitions (labelled classes)

Def. A set of N atoms is said to be *labelled* if they can be distinguished from one another. Wlog, we use labels 1 through N to refer to them.

Def. A *labelled combinatorial class* is a set of combinatorial objects built from labelled atoms and an associated *size* function.

Def. The *exponential generating function* (EGF) associated with a labelled class is the formal power series $A(z) = \sum_{a \in A} \frac{z^{|a|}}{|a|!}$

size function

object name

class name

Fundamental (elementary) identity

$$A(z) \equiv \sum_{a \in A} \frac{z^{|a|}}{|a|!} = \sum_{N \geq 0} A_N \frac{z^N}{N!}$$

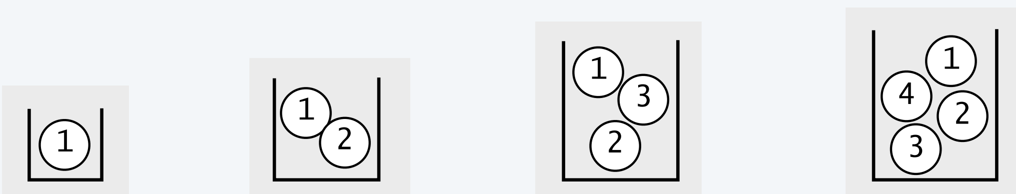
Q. How many objects of size N ?

A. $A_N = N![z^N]A(z)$

With the symbolic method, we specify the class *and at the same time* characterize the EGF

Basic labelled class 1: urns

Def. An *urn* is a set of labelled atoms.



$U_1 = 1$ $U_2 = 1$ $U_3 = 1$ $U_4 = 1$

counting sequence	*EGF*
$U_N = 1$	e^z

$$\sum_{N \geq 0} \frac{z^N}{N!} = e^z$$

Basic labelled class 2: permutations

Def. A *permutation* is a sequence of labelled atoms.

1

$P_1 = 1$

1	2
2	1

$P_2 = 2$

1	2	3
2	1	3
3	1	2
1	3	2
2	3	1
3	2	1

$P_3 = 6$

1	2	3	4	1	2	4	3
2	1	3	4	2	1	4	3
3	1	2	4	3	1	4	2
4	1	2	3	4	1	3	2
1	3	2	4	1	3	4	2
2	3	1	4	2	3	4	1
3	2	1	4	3	2	4	1
4	2	1	3	4	2	3	1
1	4	2	3	1	4	3	2
2	4	1	3	2	4	3	1
3	4	1	2	3	4	2	1
4	3	1	2	4	3	2	1

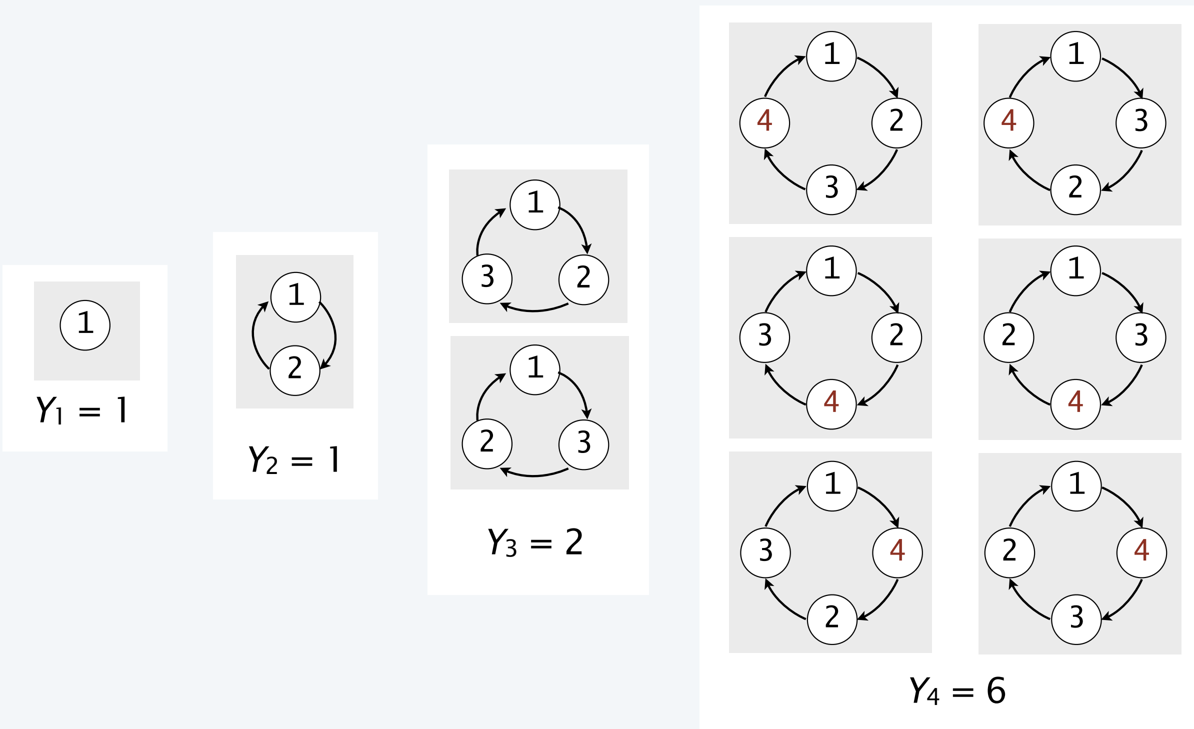
$P_4 = 24$

counting sequence	*EGF*
$P_N = N!$	$\dfrac{1}{1-z}$

$$\sum_{N\geq 0} \frac{N!z^N}{N!} = \sum_{N\geq 0} z^N = \frac{1}{1-z}$$

Basic labelled class 3: cycles

Def. A *cycle* is a cyclic sequence of labelled atoms



counting sequence	*EGF*
$Y_N = (N-1)!$	$\ln \frac{1}{1-z}$

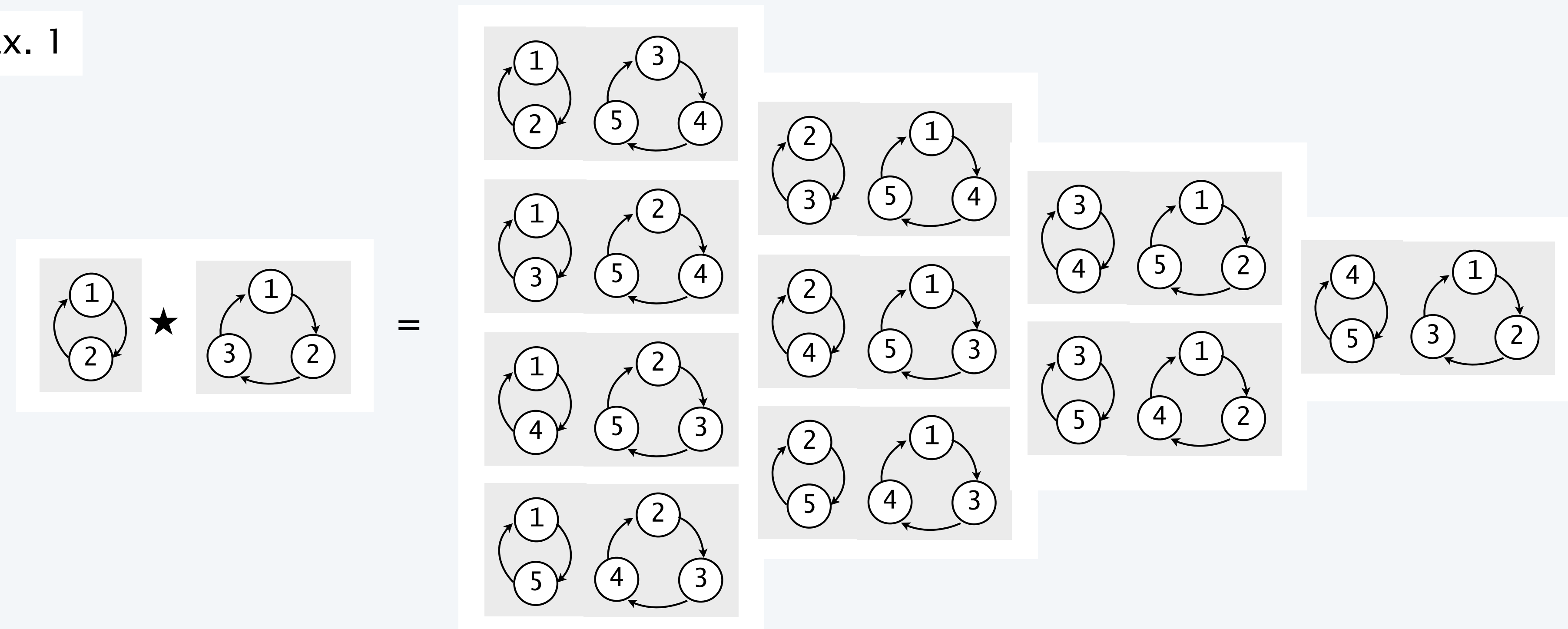
$$\sum_{N\geq 1} \frac{(N-1)!z^N}{N!} = \sum_{N\geq 1} \frac{z^N}{N} = \ln \frac{1}{1-z}$$

Labelled ("star") product operation for labelled classes

is the analog to the Cartesian product for unlabelled classes

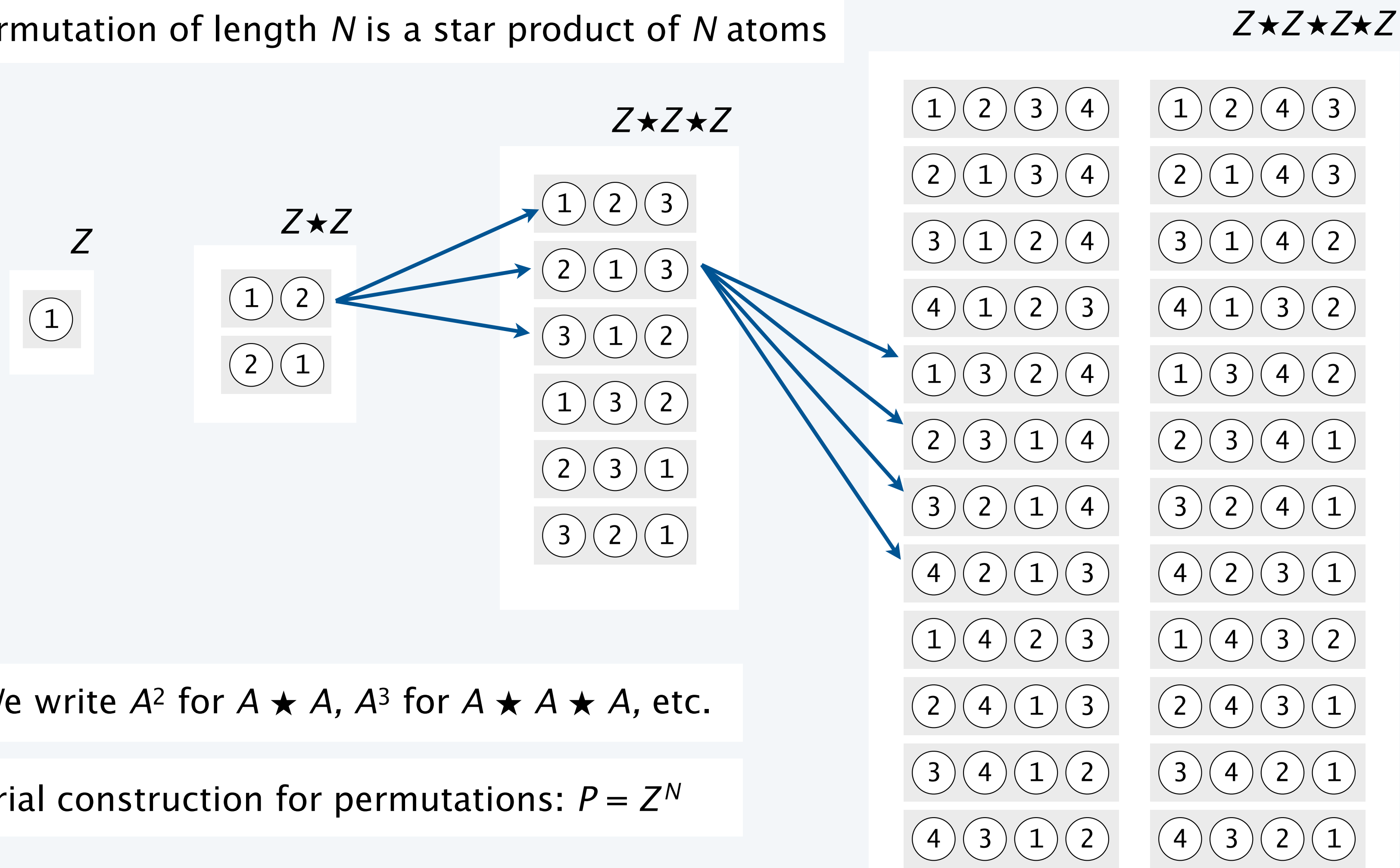
Def. Given two labelled combinatorial classes A and B, their *labelled product* $A \star B$ is a set of ordered pairs of copies of objects, one from A and one from B, *relabelled in all consistent ways*.

Ex. 1



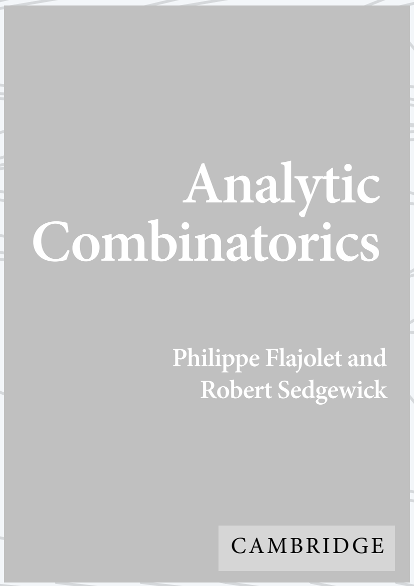
Labelled ("star") product operation for labelled classes

Ex. 2. A permutation of length N is a star product of N atoms



Notation. We write A^2 for $A \star A$, A^3 for $A \star A \star A$, etc.

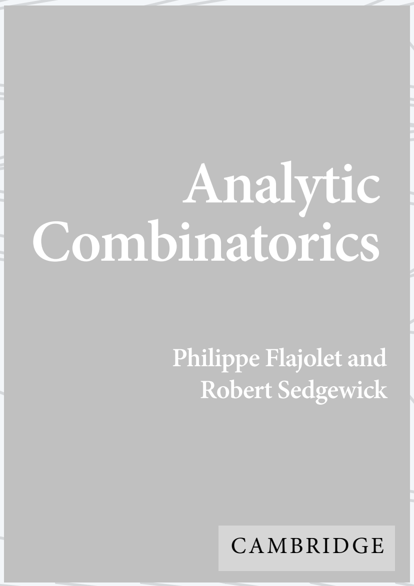
Combinatorial construction for permutations: $P = Z^N$



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- **Basics**
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- Mappings



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- **Symbolic method for labelled classes**
- Words and strings
- Labelled trees
- Mappings

Combinatorial constructions for labelled classes

construction	*notation*	*semantics*
disjoint union	$A + B$	disjoint copies of objects from A and B
labelled product	$A \star B$	ordered pairs of copies of objects, one from A and one from B *relabelled in all consistent ways*
sequence	$SEQ(A)$	sequences of objects from A
set	$SET(A)$	sets of objects from A
cycle	$CYC(A)$	cyclic sequences of objects from A

A and B are combinatorial classes of labelled objects

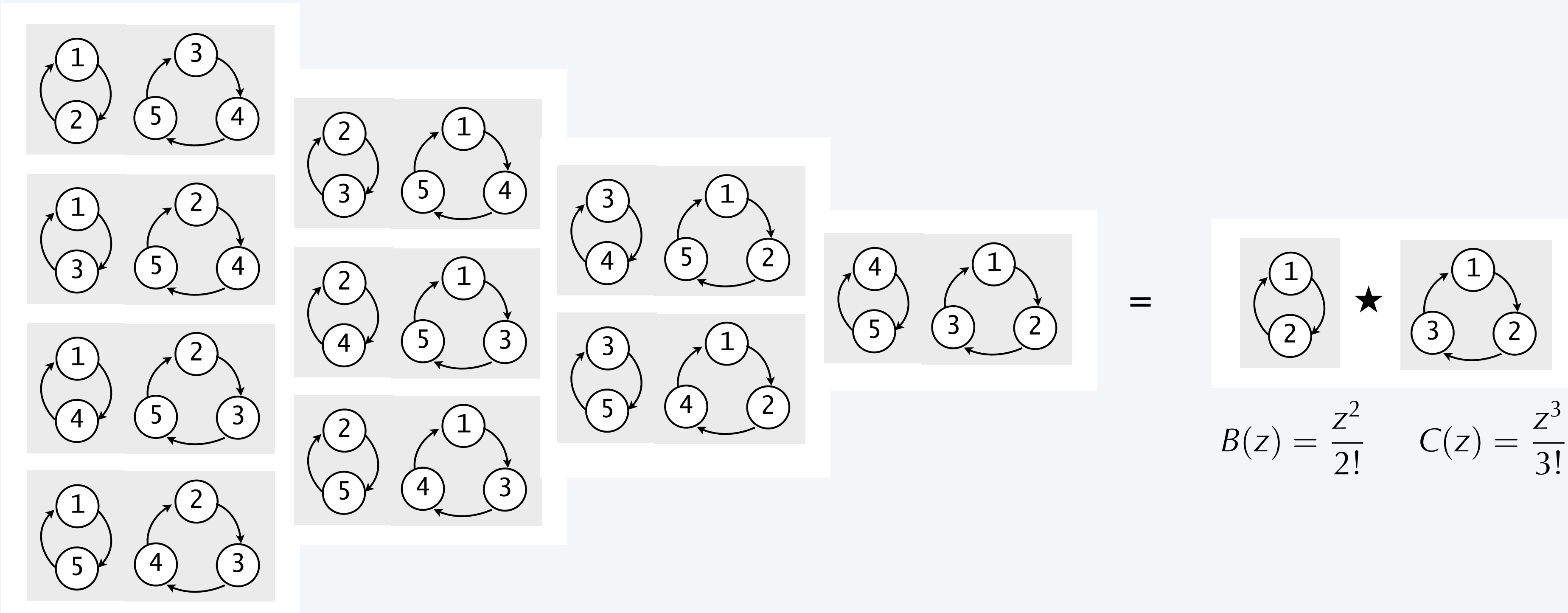
The symbolic method for labelled classes (transfer theorem)

Theorem. Let A and B be combinatorial classes of labelled objects with EGFs $A(z)$ and $B(z)$. Then

construction	*notation*	*semantics*	*EGF*
disjoint union	$A + B$	disjoint copies of objects from A and B	$A(z) + B(z)$
labelled product	$A \star B$	ordered pairs of copies of objects, one from A and one from B	$A(z)B(z)$
sequence	$SEQ_k(A)$ or A^k	k- sequences of objects from A	$A(z)^k$
	$SEQ(A)$	sequences of objects from A	$\dfrac{1}{1 - A(z)}$
set	$SET_k(A)$	k-sets of objects from A	$A(z)^k/k!$
	$SET(A)$	sets of objects from A	$e^{A(z)}$
cycle	$CYC_k(A)$	k-cycles of objects from A	$A(z)^k/k$
	$CYC(A)$	cycles of objects from A	$\ln \dfrac{1}{1 - A(z)}$

In-class exercise

Check the star-product transfer theorem for a small example.

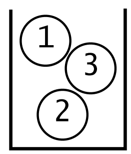
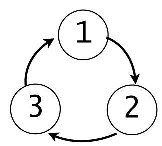


$$A(z) = 10\frac{z^5}{5!} = \frac{z^5}{12}$$

$$B(z) = \frac{z^2}{2!} \qquad C(z) = \frac{z^3}{3!}$$

$$= B(z)C(z) \quad ✓$$

The symbolic method for labelled classes: basic constructions

	urns	*cycles*	*permutations*
construction	$U = SET(Z)$	$Y = CYC(Z)$	$P = SEQ(Z)$
example			1 2 3 4
EGF	$U(z) = e^z$	$Y(z) = \ln \frac{1}{1-z}$	$P(z) = \frac{1}{1-z}$
counting sequence	$U_N = 1$	$Y_N = (N-1)!$	$P_N = N!$

construction	*notation*	*EGF*
disjoint union	$A + B$	$A(z) + B(z)$
labelled product	$A \star B$	$A(z)B(z)$
sequence	$SEQ_k(A)$	$A(z)^k$
	$SEQ(A)$	$\frac{1}{1-A(z)}$
set	$SET_k(A)$	$A(z)^k/k!$
	$SET(A)$	$e^{A(z)}$
cycle	$CYC_k(A)$	$A(z)^k/k$
	$CYC(A)$	$\ln \frac{1}{1-A(z)}$

Proofs of transfers

are immediate from GF counting

$A + B$

$$\sum_{\gamma \in A+B} \frac{z^{|\gamma|}}{|\gamma|!} = \sum_{\alpha \in A} \frac{z^{|\alpha|}}{|\alpha|!} + \sum_{\beta \in B} \frac{z^{|\beta|}}{|\beta|!} = A(z) + B(z)$$

$A \star B$

$$\sum_{\gamma \in \mathcal{A} \times \mathcal{B}} \frac{z^{|\gamma|}}{|\gamma|!} = \sum_{\alpha \in \mathcal{A}} \sum_{\beta \in \mathcal{B}} \binom{|\alpha| + |\beta|}{|\alpha|} \frac{z^{|\alpha|+|\beta|}}{(|\alpha| + |\beta|)!} = \Big(\sum_{\alpha \in \mathcal{A}} \frac{z^{|\alpha|}}{|\alpha|!}\Big)\Big(\sum_{\beta \in \mathcal{B}} \frac{z^{|\beta|}}{|\beta|!}\Big) = A(z)B(z)$$

Proofs of transfers

are immediate from GF counting

$$A(z)^k = \sum_{N\geq 0}\{\#k\text{-sequences of size } N\}\frac{z^N}{N!} = \sum_{N\geq 0} k\{\#k\text{-cycles of size } N\}\frac{z^N}{N!} = \sum_{N\geq 0} k!\{\#k\text{-sets of size } N\}\frac{z^N}{N!}$$

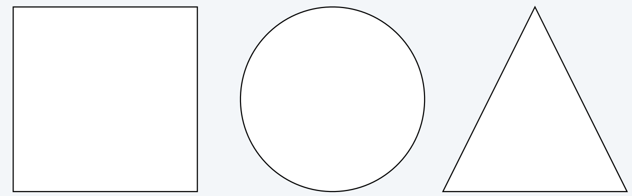
$$\frac{A(z)^k}{k} = \sum_{N\geq 0}\{\#k\text{-cycles of size } N\}\frac{z^N}{N!}$$

$$\frac{A(z)^k}{k!} = \sum_{N\geq 0}\{\#k\text{-sets of size } N\}\frac{z^N}{N!}$$

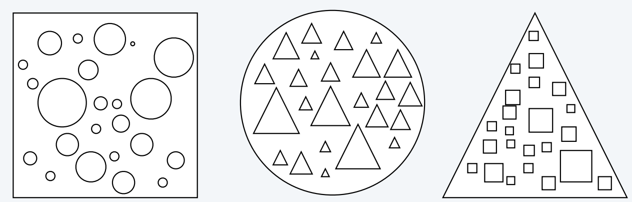
class	*construction*	*EGF*
k-sequence	$SEQ_k(A)$	$A(z)^k$
sequence	$SEQ_k(A) = SEQ_0(A) + SEQ_1(A) + SEQ_2(A) + \ldots$	$1 + A(z) + A(z)^2 + A(z)^3 + \ldots = \dfrac{1}{1-A(z)}$
k-cycle	$CYC_k(A)$	$\dfrac{A(z)^k}{k}$
cycle	$CYC_k(A) = CYC_0(A) + CYC_1(A) + CYC_2(A) + \ldots$	$1 + \dfrac{A(z)}{1} + \dfrac{A(z)^2}{2} + \dfrac{A(z)^3}{3} + \ldots = \ln\dfrac{1}{1-A(z)}$
k-set	$SET_k(A)$	$\dfrac{A(z)^k}{k!}$
set	$SET_k(A) = SET_0(A) + SET_1(A) + SET_2(A) + \ldots$	$1 + \dfrac{A(z)}{1!} + \dfrac{A(z)^2}{2!} + \dfrac{A(z)^3}{3!} + \ldots = e^{A(z)}$

A standard paradigm for analytic combinatorics

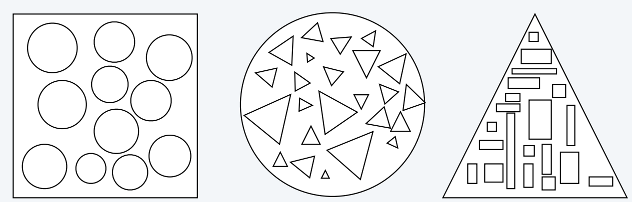
Fundamental constructs
- elementary or trivial
- confirm intuition



Compound constructs
- many possibilities
- classical combinatorial objects
- expose underlying structure



Variations
- unlimited possibilities
- *not* easily analyzed otherwise

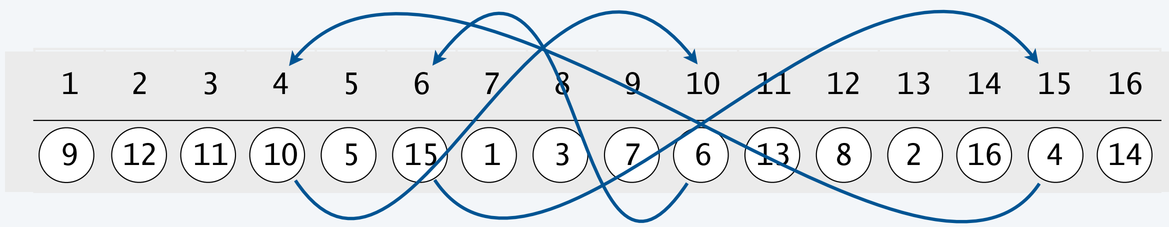


A combinatorial bijection

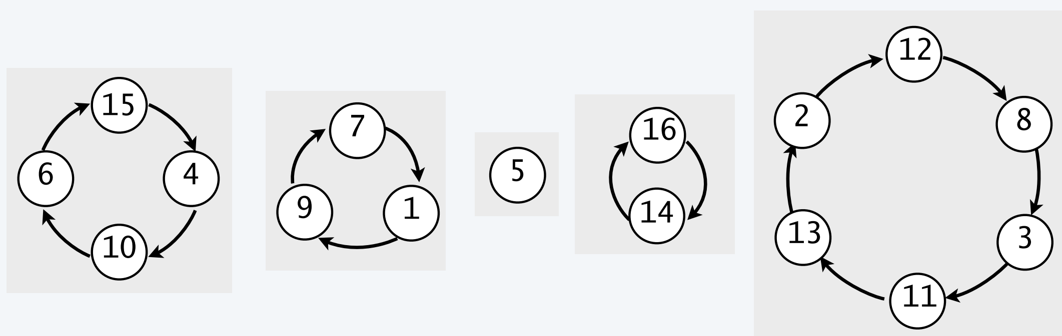
[from AC Part I Lecture 5]

A permutation is a set of cycles.

Standard representation



Set of cycles representation



Enumerating permutations

[from AC Part I Lecture 5]

How many permutations of length N?

Construction $P = SEQ(Z)$ “A permutation is a sequence of labelled atoms"

EGF equation $$P(z) = \frac{1}{1-z}$$

Counting sequence $P_N = N![z^N]P(z) = N!$

How many sets of cycles of length N?

Construction $P^* = SET(CYC(Z))$ “A permutation is a set of cycles"

EGF equation $$P^*(z) = \exp\left(\ln\frac{1}{1-z}\right) = \frac{1}{1-z}$$

Counting sequence $P^*_N = N![z^N]P^*(z) = N!$

Derangements

[from AC Part I Lecture 5]

A group of N graduating seniors each throw their hats in the air and each catch a random hat. akroom.

Q. *What is the probability that nobody gets their own hat back*?

Definition. A derangement is a permutation with no singleton cycles

Enumerating derangements

[from AC Part I Lecture 5]

How many permutations of length N ?

Construction	$P^* = SET(CYC(Z))$	"A permutation is a set of cycles"
EGF equation	$P^*(z) = \exp\big(\ln \frac{1}{1-z}\big) = \frac{1}{1-z}$	
Counting sequence	$P^*_N = N![z^N]P^*(z) = N!$	

How many derangements of length N ?

Construction	$D = SET(CYC_{>1}(Z))$	"Derangements are permutations with no singleton cycles"
EGF equation	$D(z) = e^{z^2/2+z^3/3+z^4/4+\ldots} = \exp\big(\ln \frac{1}{1-z} - z\big) = \frac{e^{-z}}{1-z}$	
Expansion	$[z^N]D(z) \equiv \frac{D_N}{N!} = \sum_{0\le k\le N} \frac{(-1)^k}{k!} \sim \frac{1}{e}$	

Derangements

[from AC Part I Lecture 5]

A group of N graduating seniors each throw their hats in the air and each catch a random hat.

Q. *What is the probability that nobody gets their own hat back?*

A. $\frac{1}{e} \doteq 0.36788$

More variations on the theme

[from AC Part I Lectures 5 and 7]

How many permutations of length N have no cycles of length $\leq M$ (*generalized derangements*)?

Construction $D_M = SET(CYC_{>M}(Z))$

"Derangements are permutations whose cycle lengths are all > M"

OGF equation $D_M(z) = e^{\frac{z^{M+1}}{M+1} + \frac{z^{M+2}}{M+2} + \cdots} = \exp\left(\ln \frac{1}{1-z} - z - z^2/2 - \ldots - z^M/M\right)$

$$= \frac{e^{-z - \frac{z^2}{2} - \frac{z^3}{3} - \cdots \frac{z^M}{M}}}{1-z}$$

How many permutations of length N have no cycles of length > 2 (*involutions*)?

Construction $I = SET(CYC_{1,2}(Z))$

"Involutions are permutations whose cycle lengths are all 1 or 2"

OGF equation $I(z) = e^{z+z^2/2}$

Standard paradigm example: permutations

DERANGEMENTS
(no singleton cycle)

$D = SET(CYC_{>1}(Z))$

$$D(z) = \frac{e^{-z}}{1-z}$$

PERMUTATIONS
with M cycles

$P_M = SET_M(CYC(Z))$

$$P_M(z) = \frac{1}{M!}\left(\ln\frac{1}{1-z}\right)^M$$

INVOLUTIONS
(cycle lengths 1 or 2)

$I = SET(CYC_{1,2}(Z))$

$$I(z) = e^{z+z^2/2}$$

PERMUTATIONS

$P = SET(CYC(Z))$

$$P(z) = e^{\ln\frac{1}{1-z}} = \frac{1}{1-z}$$

GENERALIZED INVOLUTIONS
(no cycle length > r)

$I_{\le r} = SET(CYC_{\le r}(Z))$

$$I_{\le r}(z) = e^{z+z^2/2+\ldots+z^r/r}$$

GENERALIZED DERANGEMENTS
(all cycle lengths > r)

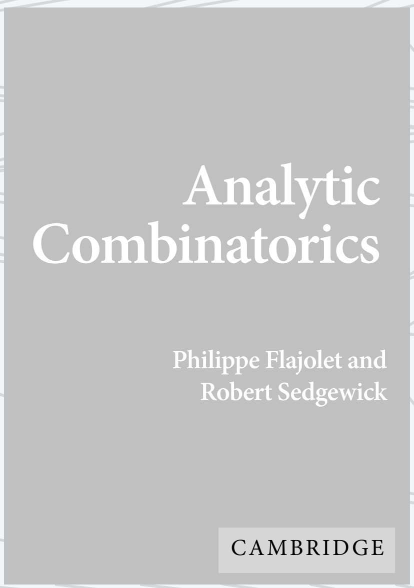
$D_{>r} = SET(CYC_{\le r}(Z))$

$$D_{>r}(z) = \frac{e^{-z-z^2/2-\ldots-z^r/r}}{1-z}$$

PERMUTATIONS
with *arbitrary*
cycle length constraints

$P_\Omega = SET_\Omega(CYC(Z))$

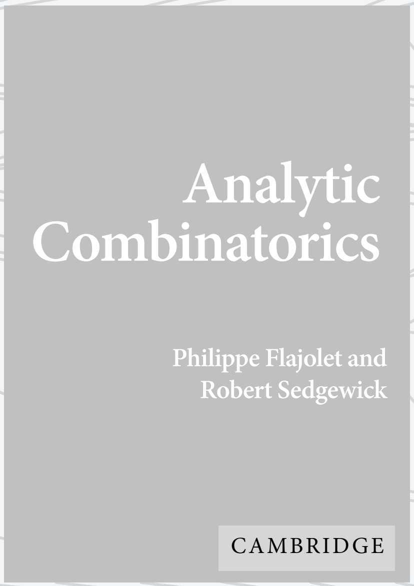
$$P_\Omega(z) = e^{\sum_{k\in\Omega} z^k/k}$$



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- **Symbolic method for labelled classes**
- Words and strings
- Labelled trees
- Mappings



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

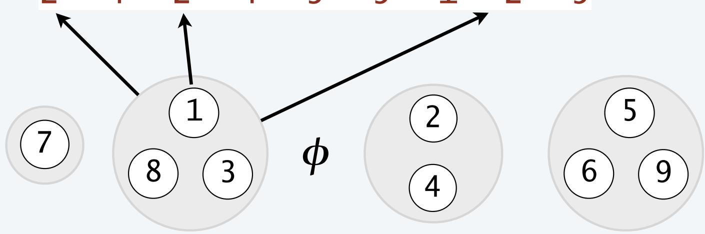
- Basics
- Symbolic method for labelled classes
- **Words and strings**
- Labelled trees
- Mappings

Words and strings

A string is a sequence of N characters (from an M-char alphabet). There are M^N strings.

A word is a sequence of M labelled sets (having N objects in total). There are M^N words.

1	2	3	4	5	6	7	8	9
2	4	2	4	5	5	1	2	5



Typical string

2 4 2 4 5 5 1 2 5

Typical word

{ 7 } { 1 8 3 } { } { 2 4 } { 5 6 9 }

Correspondence

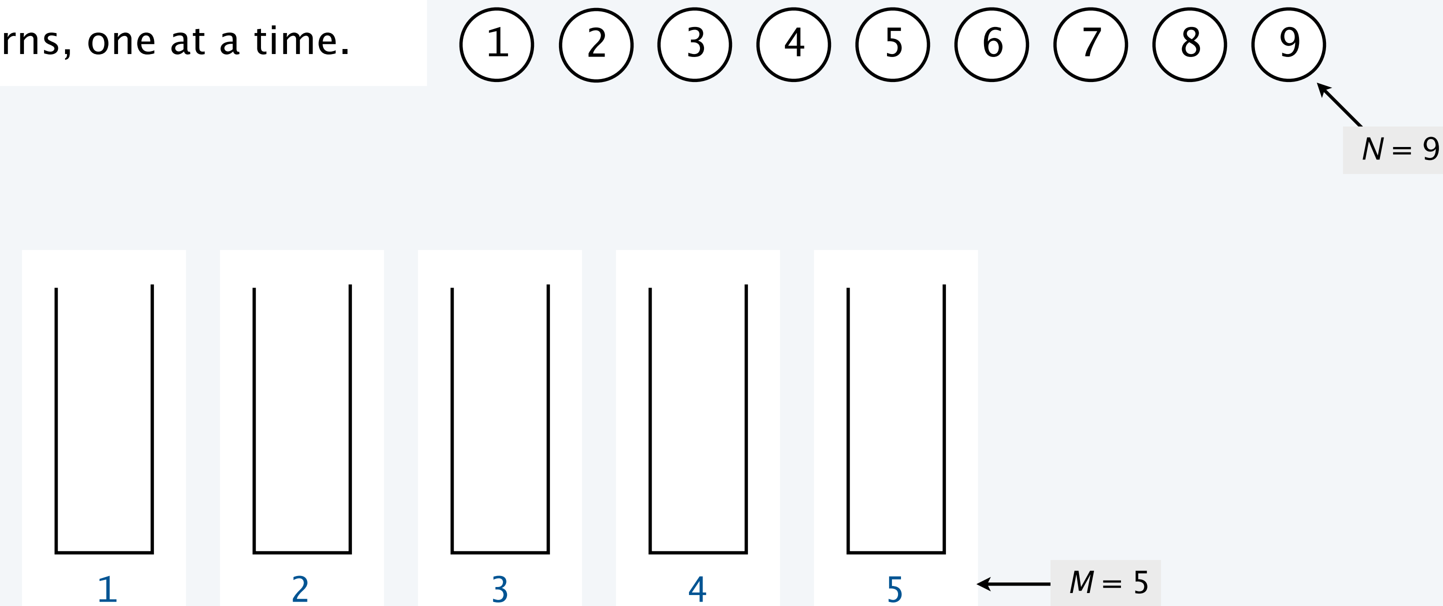
- For each i in the kth set in the word set the ith char in the string to k.
- If the ith char in the string is k, put i into the kth set in the word.

Q. What is the difference between strings and words?

A. Only the point of view (sequence of characters vs. sets of indices).

Balls and urns

Throw *N* balls into *M* urns, one at a time.



Balls-and-urns sequences are equivalent to strings and words

Corresponding string 2 5 1 5 1 1 4 4 3

Corresponding word { 3 5 6 } { 1 } { 9 } { 7 8 } { 2 4 }

Words

Def. A *word* is a sequence of M urns holding N objects in total.

"throw N balls into M urns"

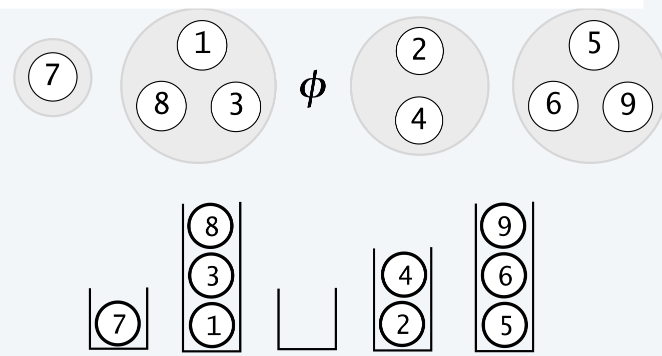
Q. How many words ?

Class	W_M, the class of M-sequences of urns
Size	$\|w\|$, the number of objects in w
EGF	$W_M(z) = \sum_{w \in W_M} \frac{z^{\|w\|}}{\|w\|!} = \sum_{N \ge 0} W_{MN} \frac{z^N}{N!}$

Atom

type	class	size	GF
labelled atom	Z	1	z

Example { 7 } { 1 8 3 } { } { 2 4 } { 5 6 9 }



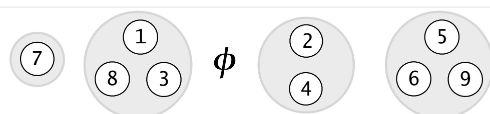
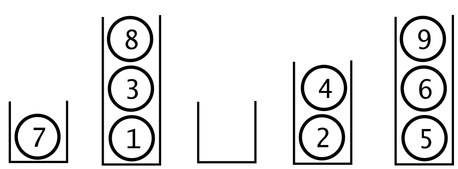
2 4 2 4 5 5 1 2 5

Construction $$W_M = SEQ_M(SET(Z))$$

OGF equation $$W_M(z) = (e^z)^M = e^{Mz}$$

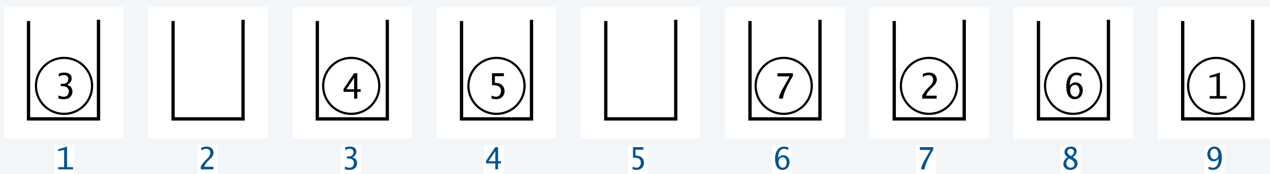
Counting sequence $$N![z^N]W_M(z) = M^N$$

Strings and Words (summary)

class	*type*	*GF type*	*example*	*AC enumeration*	*prototypical application*
STRING	unlabelled	OGF	2 4 2 4 5 5 1 2 5	$S = \text{SEQ}(Z_1 + \ldots + Z_M)$ $S(z) = \dfrac{1}{1 - Mz}$ $S_{MN} = M^N$	string search
WORD	labelled	EGF	 { 7 } { 1 8 3 } { } { 2 4 } { 5 6 9 } 2 4 2 4 5 5 1 2 5 	$W_M = \text{SEQ}_M(\text{SET}(Z))$ $W_M(z) = e^{Mz}$ $W_{MN} = M^N$	hashing

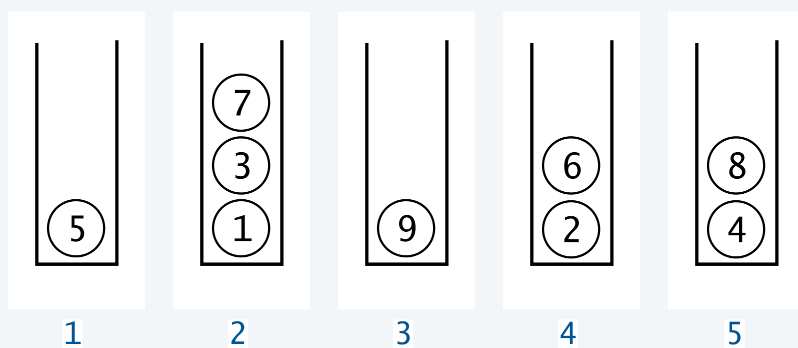
Variations on words: occupancy restrictions

Def. A *birthday sequence* is a word where no letter appears twice.



9 7 1 3 4 8 6

Def. A *coupon collector sequence* is a word where every letter appears at least once.



2 4 2 5 1 4 2 5

Birthday sequences (M-words with no duplicates)

Def. A *birthday sequence* is a word where no set has more than one element.

a string with no duplicate letters

Q. How many birthday sequences?

Class	B_M, the class of birthday sequences
EGF	$B_M(z) = \sum_{w \in B_M} \frac{z^{\lvert w \rvert}}{\lvert w \rvert !} = \sum_{N \geq 0} B_{MN} \frac{z^N}{N!}$

Example

{ 3 } { } { 5 } { 1 } { } { } { 4 } { 2 } { }

4 8 1 7 3

Construction $B_M = SEQ_M(E + Z)$

EGF equation $B_M(z) = (1+z)^M$

Counting sequence

$$N![z^N]B_M(z) = N!\binom{M}{N} = \frac{M!}{(M-N)!}$$
$$= M(M-1)\ldots(M-N+1)$$

Coupon collector sequences (M-words with no empty sets)

Def. A *coupon collector sequence* is an M-word with no empty set.

a string that uses all the letters in the alphabet

Q. How many coupon collector sequences?

Class	R_M, the class of coupon collector sequences
EGF	$R_M(z) = \sum_{w \in R_M} \frac{z^{\|w\|}}{\|w\|!} = \sum_{N \geq 0} R_{MN} \frac{z^N}{N!}$

Example ($M = 26$)

```
the quick brown fox jumps over the lazy dog
```

Example ($M = 5$)

```
2 4 2 4 5 5 1 5 3
{ 7 } { 1 3 } { 9 } { 2 4 } { 5 6 8 }
```

Construction $$R_M = SEQ_M(SET_{>0}(Z))$$

EGF equation $$R_M(z) = (e^z - 1)^M$$

Surjections

Def. An *M-surjection* is an M-word with no empty set. ← Alt name for "coupon collector sequence"

Def. A *surjection* is a word that is an M-surjection for some M.

Q. How many surjections of length N?

Class	R_M, the class of M-surjections

Construction

$$R_M = SEQ_M(SET_{>0}(Z))$$

EGF equation

$$R_M(z) = (e^z - 1)^M$$

Coefficients

$$R_{MN} \sim M^N$$

Class	R, the class of surjections

Construction

$$R = SEQ(SET_{>0}(Z))$$

EGF equation

$$R(z) = \frac{1}{1-(e^z-1)} = \frac{1}{2-e^z}$$

Coefficients

$$N![z^N]R(z) \sim \frac{N!}{2(\ln 2)^{N+1}}$$

← Best handled with complex asymptotics (stay tuned)

1

$R_1 = 1$

1 1
1 2
2 1

$R_2 = 3$

1 1 1
1 1 2
1 2 1
1 2 2
1 2 3
1 3 2
2 1 1
2 1 2
2 1 3
2 2 1
2 3 1
3 1 2
3 2 1

$R_3 = 13$

Some variations on words

M-SURJECTIONS
(*M*-word, all letters used)

$R_M = SEQ_M(SET_{>0}(Z))$

$R_M(z) = (e^z - 1)^M$

SURJECTIONS
(*M*-word for some *M*, all letters used)

$R = SEQ(SET_{>0}(Z))$

$$R(z) = \frac{1}{2 - e^z}$$

M-WORD

$W_M = SEQ_M(SET(Z))$

$W_M(z) = (e^z)^M = e^{Mz}$

Generalized Coupon Collector

MIN occupancy *M*-WORDS
(all letter counts > *b*)

$W_M^{>b} = SEQ_M(SET_{>b}(Z))$

$W_M^{>b}(z) = \left(\sum_{k>b} z^k/k!\right)^M$

Generalized Birthday
MAX occupancy *M*-WORDS
(all letter counts ≤ *b*)

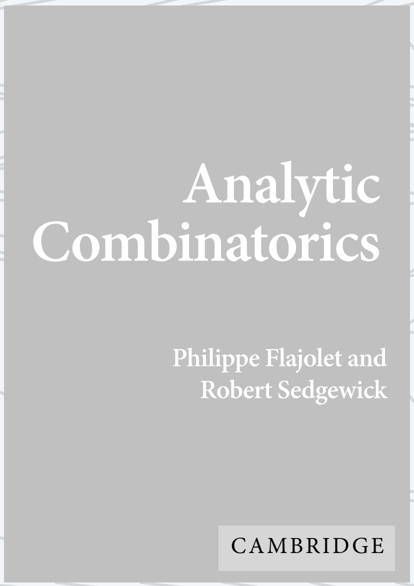
$W_M^{\le b} = SEQ_M(SET_{\le b}(Z))$

$W_M^{\le b}(z) = \left(\sum_{k\le b} z^k/k!\right)^M$

OCCUPANCY CONSTRAINED *M*-WORDS
(arbitrary letter count constraints)

$W_{M\Omega} = SEQ_M(SET_\Omega(Z))$

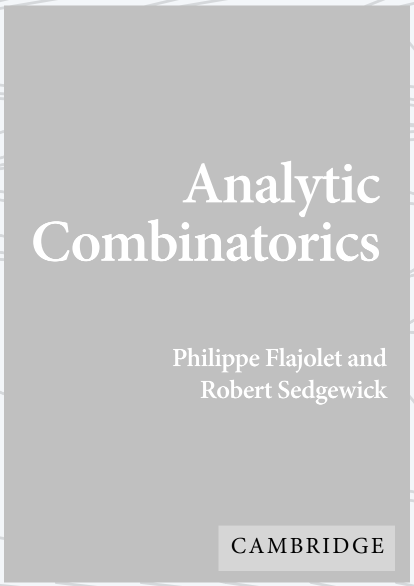
$W_{M\Omega}(z) = \left(\sum_{k\in\Omega} z^k/k!\right)^M$



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- **Words and strings**
- Labelled trees
- Mappings



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- **Labelled trees**
- Mappings

Labelled trees

Def. A *labelled tree* with N nodes is a tree whose nodes are labelled with the integers 1 to N.

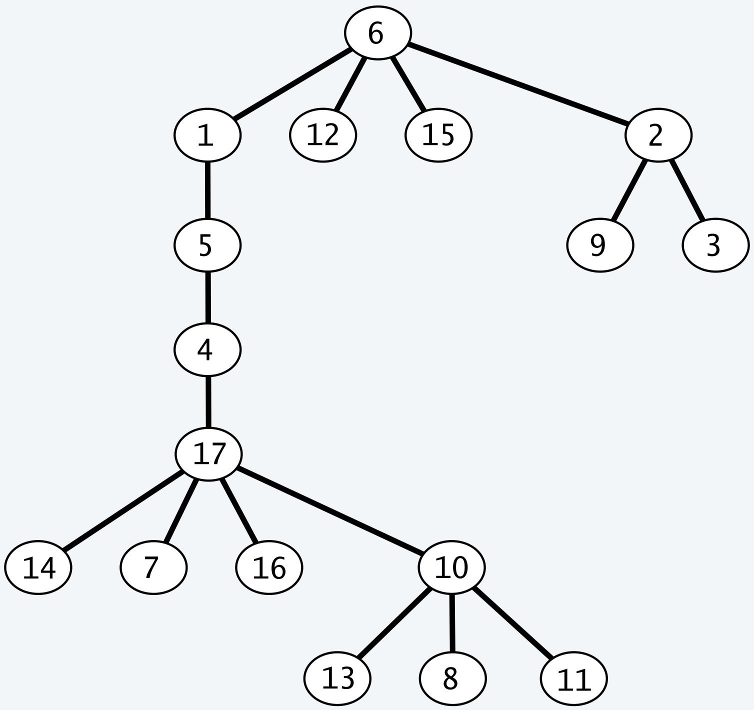
Q. How many different labelled trees of size N?

Q. Order of subtrees significant?

Q. Rooted?

Q. Binary?

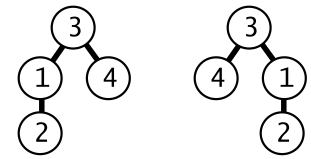
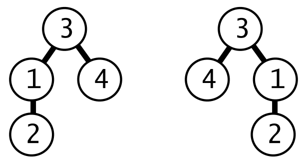
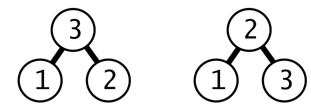
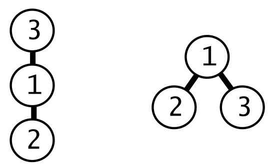
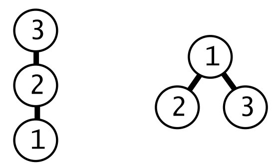
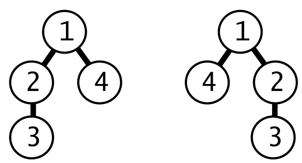
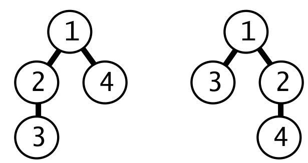
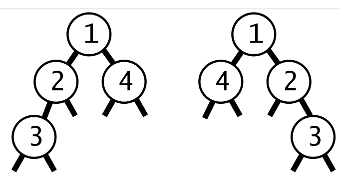
Q. Labels increase along paths?



Some of these questions are trivial; others are classic.

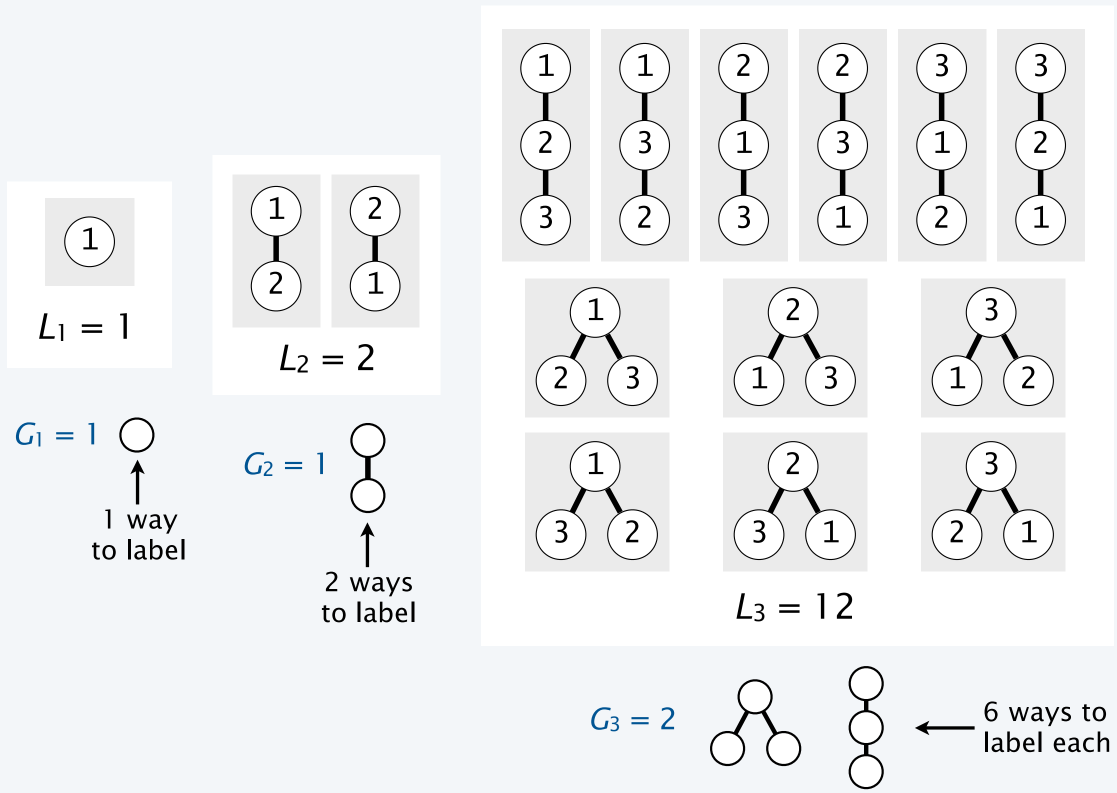
All of them are easily answered with analytic combinatorics.

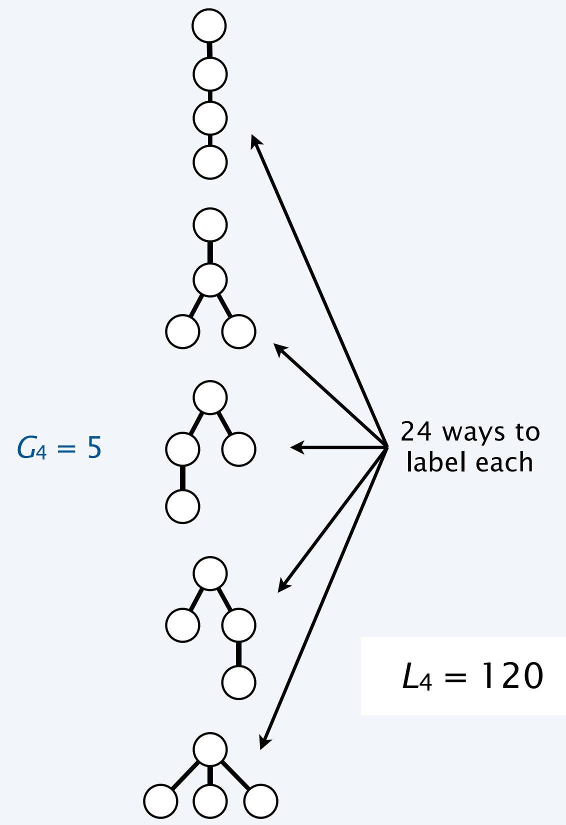
Counting labelled trees

class	*same trees*	*reason*	*different trees*	*reason*
rooted ordered				order of subtrees is significant
rooted unordered (Cayley)		order of subtrees is *not* significant		root label
unrooted unordered		same labels on middle node		different labels on middle node
increasing Cayley				different labels on paths
increasing binary				order of subtrees is significant

Labelled trees

Q. How many different labelled rooted ordered trees of size N?





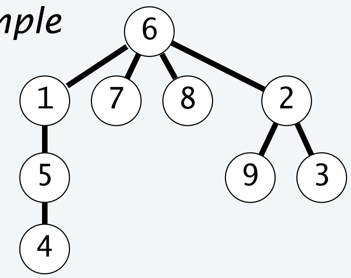
A. $N!\,G_N$. Proof. Label any canonical walk of every unlabelled tree $N!$ different ways

Labeled rooted ordered trees

Q. How many different labelled rooted ordered trees of size N?

Class	L, the class of labelled rooted ordered trees
EGF	$L(z) = \sum_{l \in L} \frac{z^{\vert l \vert}}{\vert l \vert !} = \sum_{N \geq 0} L_N \frac{z^N}{N!}$

Example



Construction $\qquad L = Z \star SEQ(L)$ $\qquad$ "A tree is a root and a sequence of trees"

EGF equation $\qquad L(z) = \dfrac{z}{1 - L(z)}$ $\qquad$ Same as **OGF** for *unlabelled* trees

Counting sequence $\qquad L_N = N![z^N]L(z) = N![z^N]G(z) = N!G_N$ $\qquad$ $N!$ ways to label a tree walk

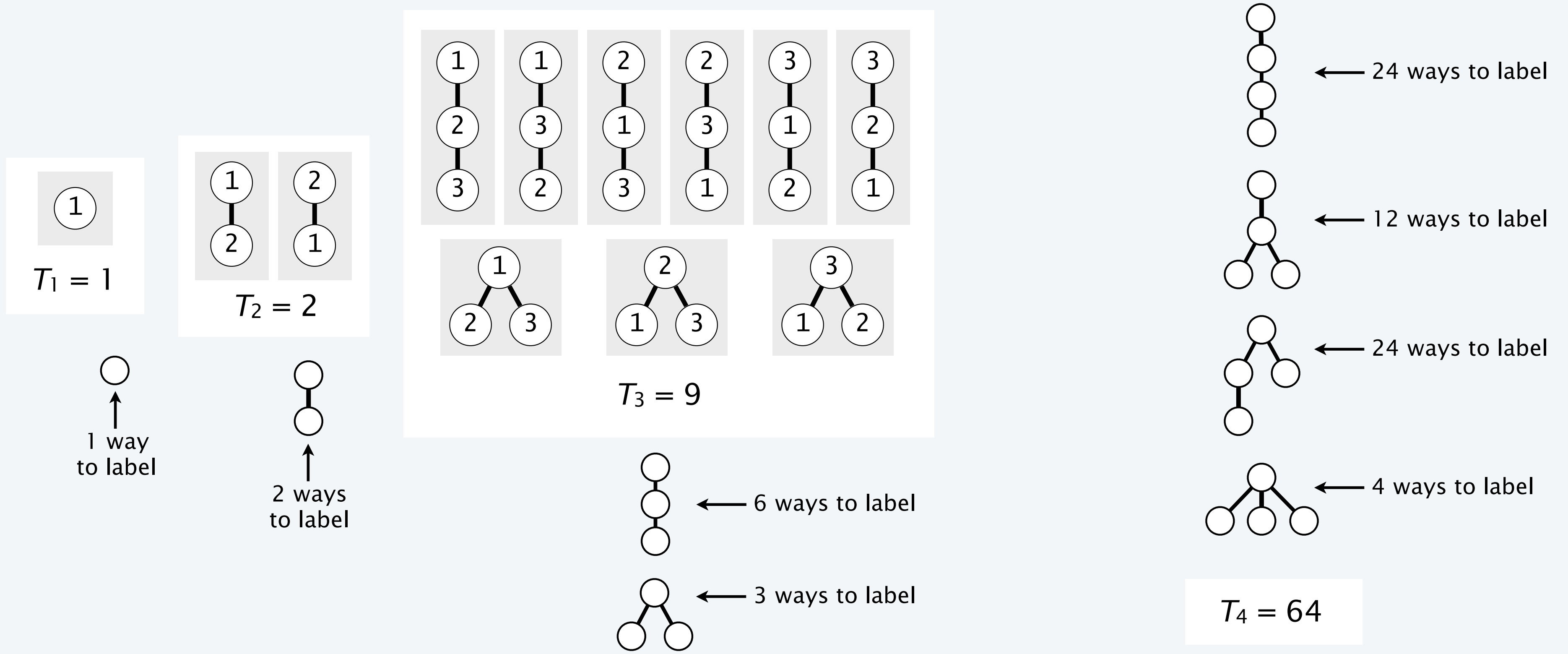
$$= N! \frac{1}{N} \binom{2N-2}{N-1} = \frac{(2N-2)!}{(N-1)!}$$

$$\sim \frac{(4/e)^N}{2\sqrt{2}} N^{N-1}$$

Stirling's approximation

Cayley trees

Q. How many different labelled rooted *unordered* trees of size N?

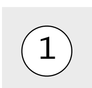


A. N^{N-1}. Proof. Stay tuned: Cayley trees are special cases of *mappings* (next section)

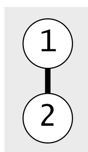
Increasing Cayley trees

Q. How many different Cayley trees of size N *with increasing labels on every path* ?

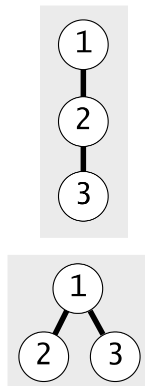
"Cayley" = "rooted, labelled, unordered"



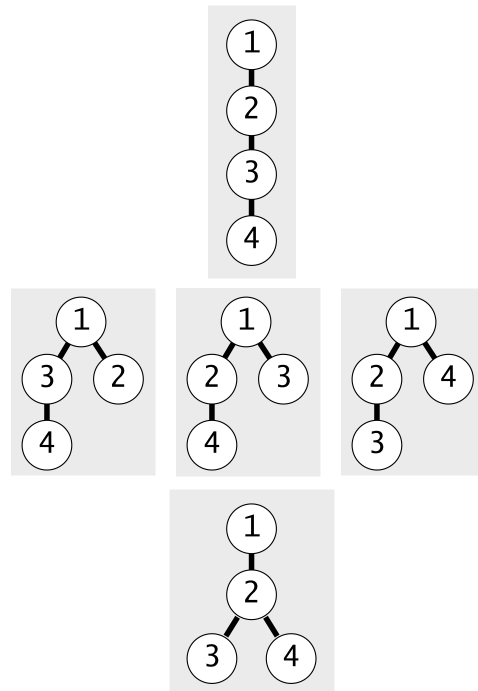
$Q_1 = 1$

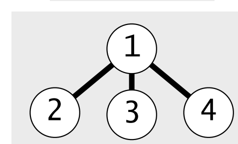


$Q_2 = 1$



$Q_3 = 2$





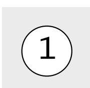
$Q_4 = 6$

A. $(N-1)!$. Proof. Stay tuned.

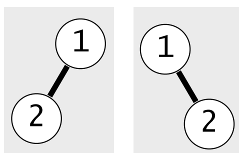
Increasing binary trees

Q. How many different *binary* trees of size N with increasing labels on every path ?

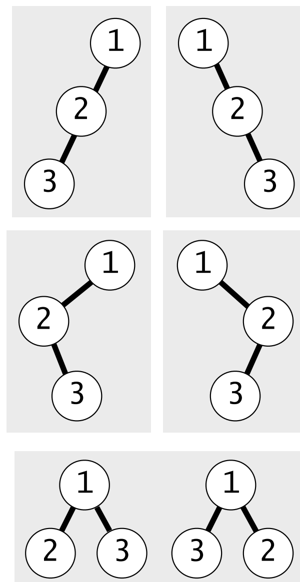
"binary" = "ordered, each node with 0 or 2 children"



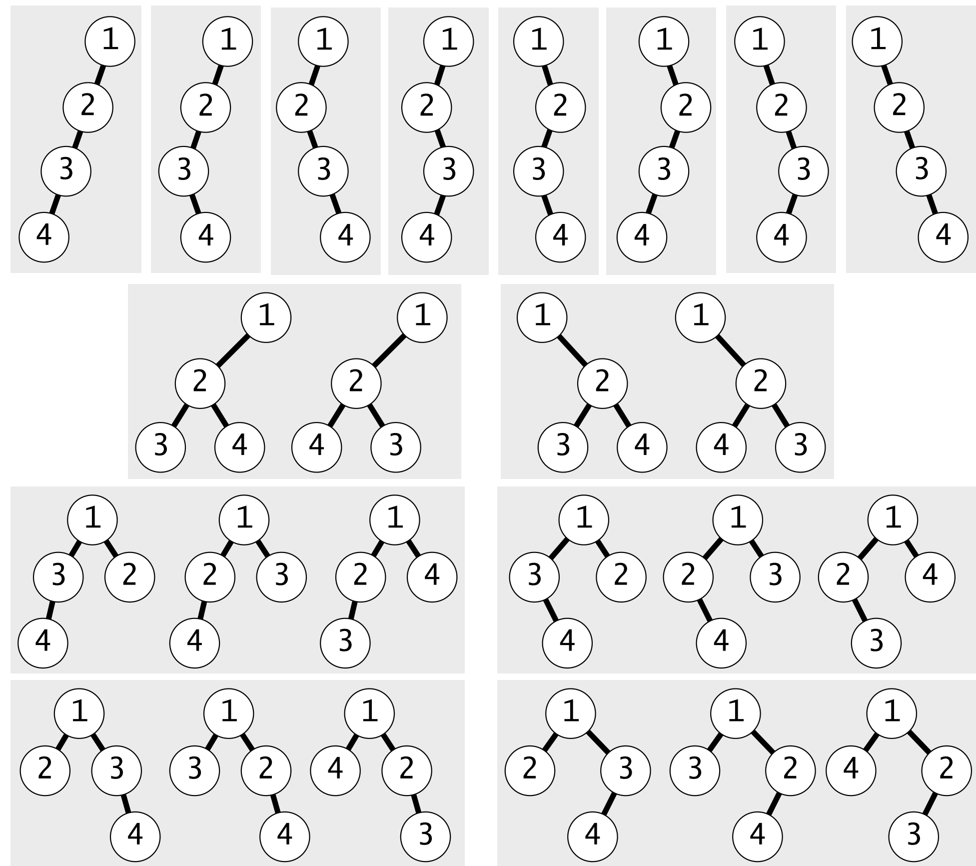
$R_1 = 1$



$R_2 = 2$



$R_3 = 6$



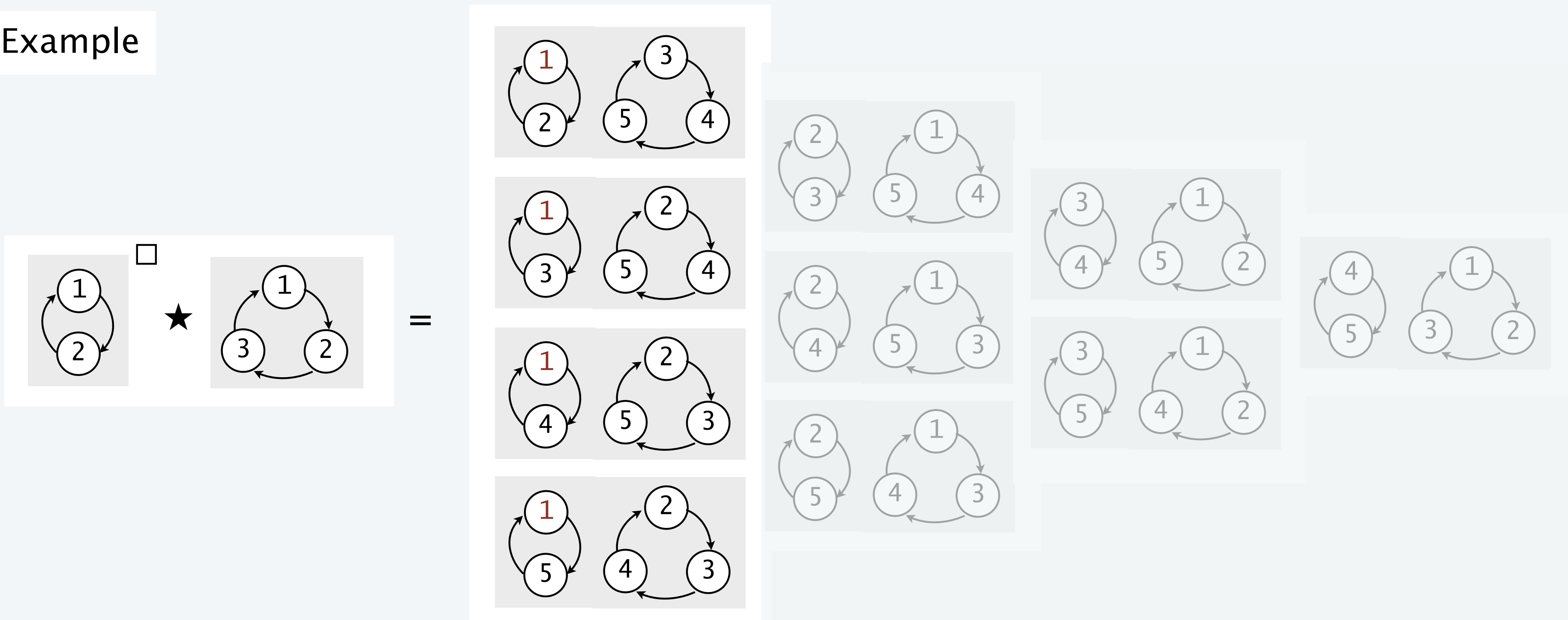
$R_4 = 24$

A. $N!$. Proof. Stay tuned.

Boxed product construction for labelled classes

construction	*notation*	*semantics*
boxed product	$A = B^{\square} \star C$	subset of B★C where *smallest* labelled element is from *B*

Example



Transfer theorem for the boxed product

construction	*notation*	*semantics*	*EGF*
boxed product	$A = B^{\square} \star C$	subset of B★C where *smallest* labelled element is from *B*	$A'(z) = B'(z)C(z)$

Proof.

$$A_N = \sum_{1\le k\le N} \binom{N-1}{k-1} B_k C_{N-k}$$

$$\frac{A_N}{(N-1)!} = \sum_{1\le k\le N} \frac{B_k}{(k-1)!}\frac{C_{N-k}}{(N-k)!}$$

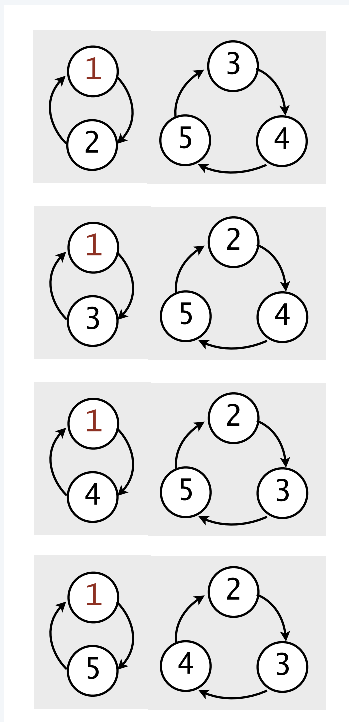
$$A'(z) = \sum_{N\ge 1} \frac{A_N}{(N-1)!} z^{N-1} = \sum_{N\ge 1}\sum_{1\le k\le N} \frac{B_k}{(k-1)!}\frac{C_{N-k}}{(N-k)!} z^{N-1} = \sum_{k\ge 1}\sum_{N\ge k} \frac{B_k}{(k-1)!}\frac{C_{N-k}}{(N-k)!} z^{N-1}$$

$$= \sum_{k\ge 1}\sum_{N\ge 0} \frac{B_k}{(k-1)!}\frac{C_N}{N!} z^{N+k-1} = \sum_{k\ge 1} \frac{B_k}{(k-1)!} z^{k-1} \sum_{N\ge 0} \frac{C_N}{N!} z^N$$

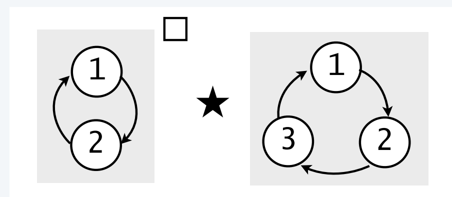
$$= B'(z)C(z)$$

In-class exercise

Check the boxed-product transfer theorem for a small example.



=



$$A(z) = 4\frac{z^5}{5!}$$

$$B(z) = \frac{z^2}{2!} \qquad C(z) = \frac{z^3}{3!}$$

$$B'(z) = z$$

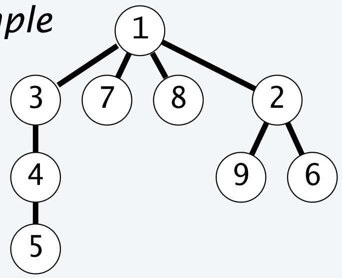
$$A'(z) = \frac{z^4}{3!}$$

$$= B'(z)C(z) \quad \checkmark$$

Increasing trees

Class	Q, the class of Cayley trees *whose labels increase on every path*
Construction	$Q = Z^{\Box} \star SET(Q)$
EGF equation	$Q'(z) = e^{Q(z)}$
Solution	$Q(z) = \ln \frac{1}{1-z}$
Counting sequence	$Q_N = N![z^N]Q(z) = (N-1)!$

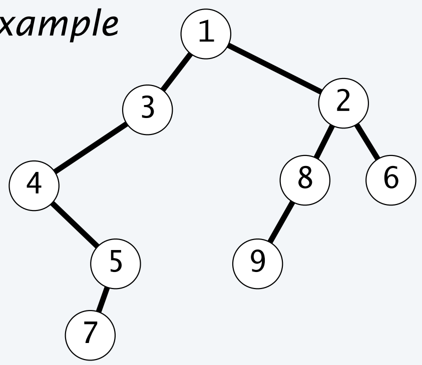
Example



"Cayley" = "rooted, labelled, unordered"

Class	B, the class of binary trees *whose labels increase on every path*
Construction	$B = E + Z^{\Box} \star B \star B$
EGF equation	$B'(z) = B(z)^2$
Solution	$B(z) = \frac{1}{1-z}$
Counting sequence	$B_N = N![z^N]B(z) = N!$

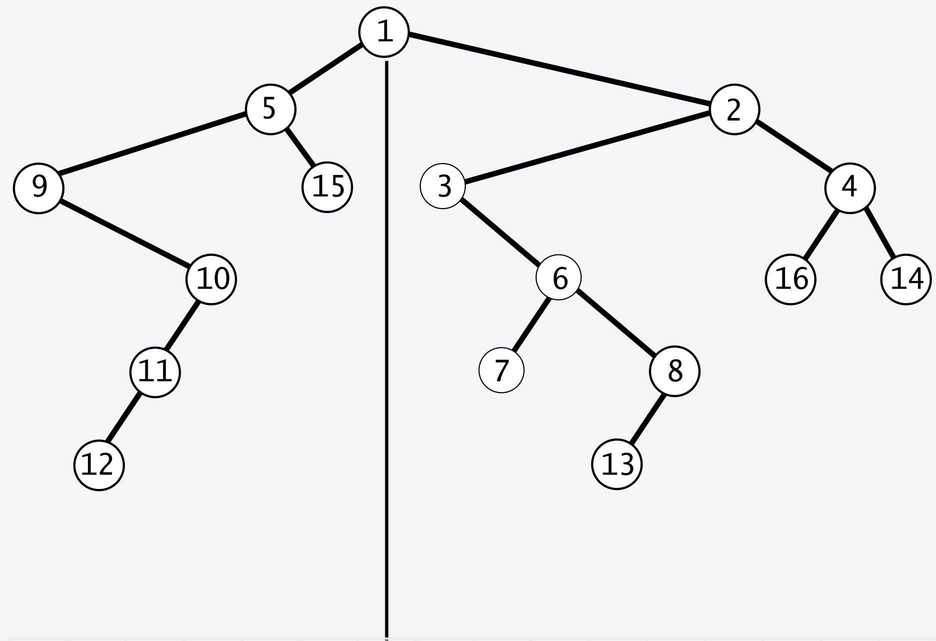
Example



"binary" = "ordered, each node with 0 or 2 children"

A permutation is an increasing binary tree

Increasing binary tree representation



Standard representation

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
9	12	11	10	5	15	1	3	7	6	13	8	2	16	4	14

Some variations on labelled trees

BINARY

M-ARY

ROOTED ORDERED

$L = Z \star SEQ(L)$

$$L(z) = \frac{z}{1 - L(z)}$$

ROOTED UNORDERED
(Cayley)

$C = Z \star SET(C)$

$$C(z) = ze^{C(z)}$$

INCREASING CAYLEY

$Q = Z^{\square} \star SET(Q)$

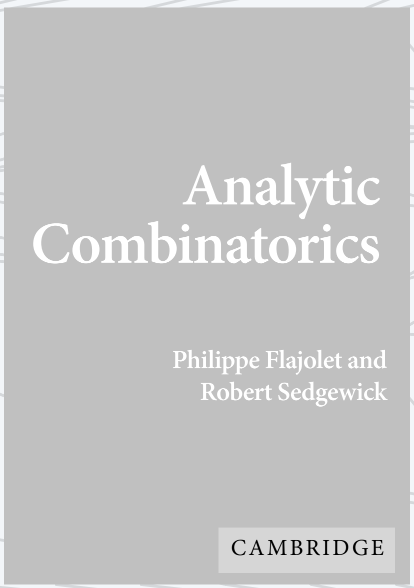
$$Q(z) = \ln \frac{1}{1 - z}$$

INCREASING BINARY

$B = E + Z^{\square} \star B \star B$

$$B(z) = \frac{1}{1 - z}$$

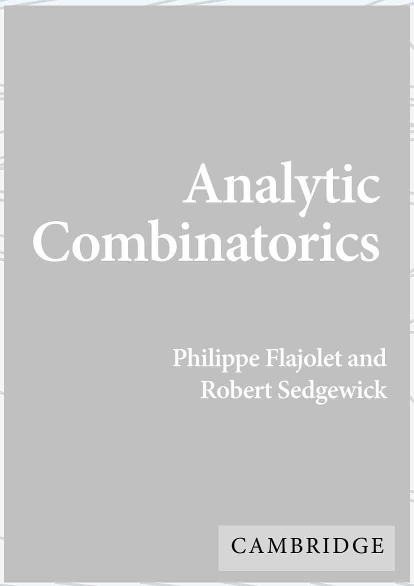
UNROOTED UNORDERED



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- **Labelled trees**
- Mappings



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- **Mappings**

Mappings

Q. How many *N*-words of length *N* ?

1

$M_1 = 1$

1 1
1 2
2 1
2 2

$M_2 = 4$

1 1 1	2 1 1	3 1 1
1 1 2	2 1 2	3 1 2
1 1 3	2 1 3	3 1 3
1 2 1	2 2 1	3 2 1
1 2 2	2 2 2	3 2 2
1 2 3	2 2 3	3 2 3
1 3 1	2 3 1	3 3 1
1 3 2	2 3 2	3 3 2
1 3 3	2 3 3	3 3 3

$M_3 = 27$

1 1 1 1	2 1 1 1	3 1 1 1	4 1 1 1
1 1 1 2	2 1 1 2	3 1 1 2	4 1 1 2
1 1 1 3	2 1 1 3	3 1 1 3	4 1 1 3
1 1 1 4	2 1 1 4	3 1 1 4	4 1 1 4
1 1 2 1	2 1 2 1	3 1 2 1	4 1 2 1
1 1 2 2	2 1 2 2	3 1 2 2	4 1 2 2
1 1 2 3	2 1 2 3	3 1 2 3	4 1 2 3
1 1 2 4	2 1 2 4	3 1 2 4	4 1 2 4
1 1 3 1	2 1 3 1	3 1 3 1	4 1 3 1
1 1 3 2	2 1 3 2	3 1 3 2	4 1 3 2
1 1 3 3	2 1 3 3	3 1 3 3	4 1 3 3
1 1 3 4	2 1 3 4	3 1 3 4	4 1 3 4
1 1 4 1	2 1 4 1	3 1 4 1	4 1 4 1
1 1 4 2	2 1 4 2	3 1 4 2	4 1 4 2
1 1 4 3	2 1 4 3	3 1 4 3	4 1 4 3
1 1 4 4	2 1 4 4	3 1 4 4	4 1 4 4
1 2 1 1	2 2 1 1	3 2 1 1	4 2 1 1
. . .	. . .	. . .	. . .

$M_4 = 64$

A. N^N

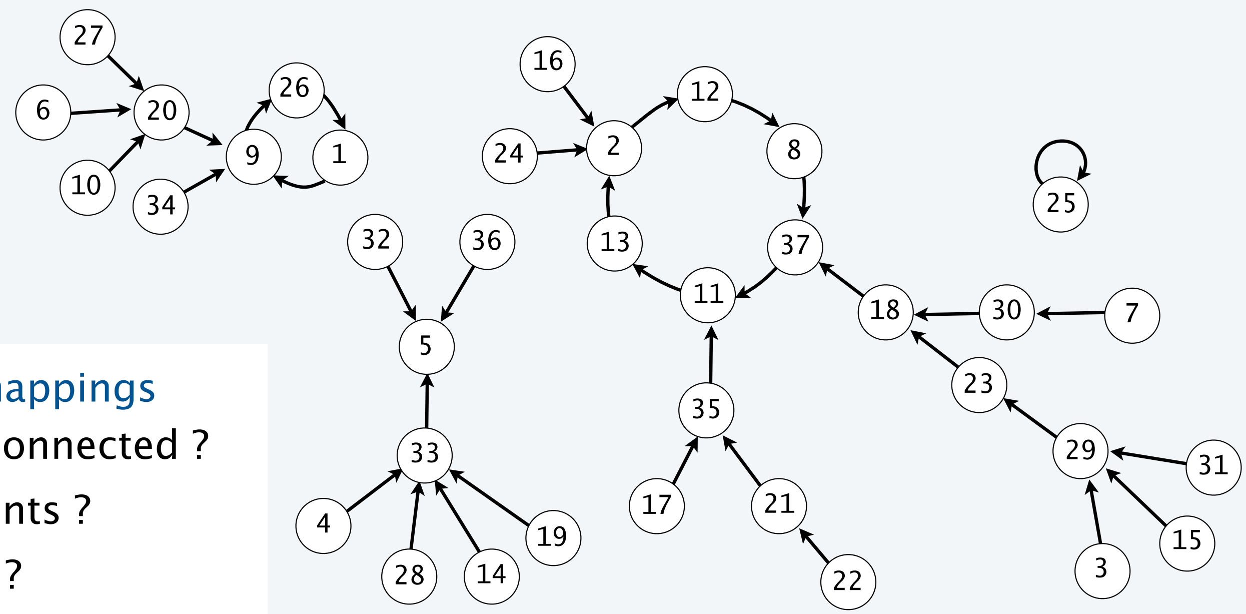
Mappings

Def. A *mapping* is a function from the set of integers from 1 to N onto itself.

Example

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
9	12	29	33	5	20	30	37	26	20	13	8	2	33	29	2	35	37	33	9	35	21	18	2	25	1	20	33	23	18	29	5	5	9	11	5	11

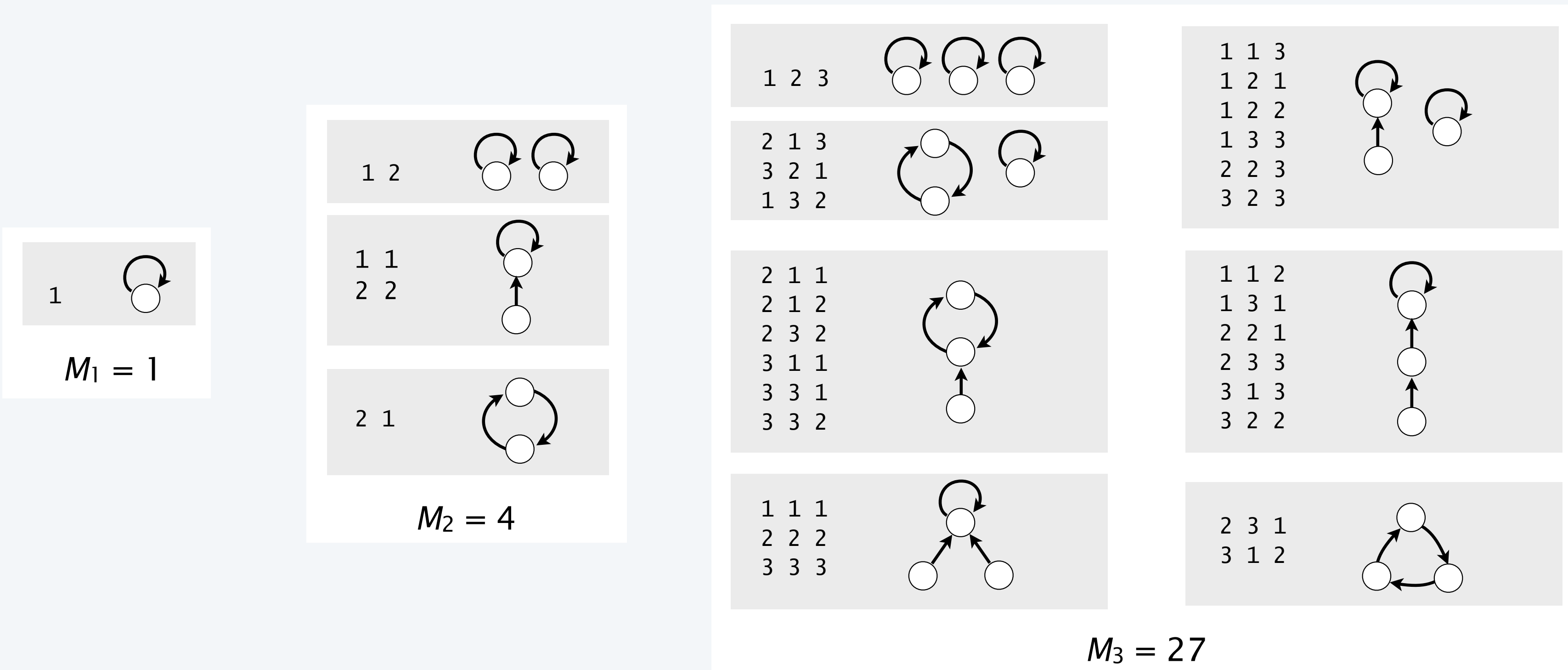
Every mapping corresponds to a digraph
- N vertices, N edges
- Outdegrees: all 1
- Indegrees: between 0 and N



Natural questions about random mappings
- Probability that the digraph is connected ?
- How many connected components ?
- How many nodes are on cycles ?

Mappings

Q. How many *mappings* of length N ?



A. N^N, by correspondence with N-words, but *internal structure is of interest*.

Lagrange inversion

is a classic method for computing a *functional inverse*.

Def. The *inverse* of a function $f(u) = z$ is the function $u = g(z)$.

Ex. $f(u) = \dfrac{u}{1-u}$ $\quad g(z) = \dfrac{z}{1+z}$

Lagrange Inversion Theorem.

If a GF $g(z) = \sum_{n\geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$

with $f(0) = 0$ and $f'(0) \neq 0$ then $g_n = \dfrac{1}{n}[u^{n-1}]\left(\dfrac{u}{f(u)}\right)^n$.

Proof. Omitted (best understood via complex analysis).

Ex. $f(u) = \dfrac{u}{1-u}$ $\quad g_n = \dfrac{1}{n}[u^{n-1}](1-u)^n = (-1)^{n-1}$ $\quad \sum_{n\geq 1}(-1)^n z^n = \dfrac{z}{1+z}$ ✓

Analytic combinatorics context: A widely applicable analytic transfer theorem

Lagrange-Bürmann inversion

A more general (and more useful) formulation:

Lagrange Inversion Theorem (Bürmann form).

If a GF $g(z) = \sum_{n\geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$

with $f(0) = 0$ and $f'(0) \neq 0$ then, for any function $H(u)$,

$$[z^n]H(g(z)) = \frac{1}{n}[u^{n-1}]H'(u)\Big(\frac{u}{f(u)}\Big)^n$$

$H(u) = u$ gives the basic theorem

One important application: enumerating mappings

Lagrange inversion: classic application

How many binary trees with N external nodes?

Class	T, the class of all binary trees
Size	The number of external nodes

Construction $$T = Z + T \times T$$

OGF equation $$T(z) = z + T(z)^2$$

$$z = T(z) - T(z)^2$$

Extract coefficients by Lagrange inversion with $f(u) = u - u^2$

$$[z^N]T(z) = \frac{1}{N}[u^{N-1}]\Big(\frac{1}{1-u}\Big)^N$$

$$= \frac{1}{N}\binom{2N-2}{N-1} \quad \checkmark$$

Lagrange Inversion Theorem.

If a GF $g(z) = \sum_{n\geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$ with $f(0) = 0$ and $f'(0) \neq 0$ then $g_n = \frac{1}{n}[u^{n-1}]\Big(\frac{u}{f(u)}\Big)^n$.

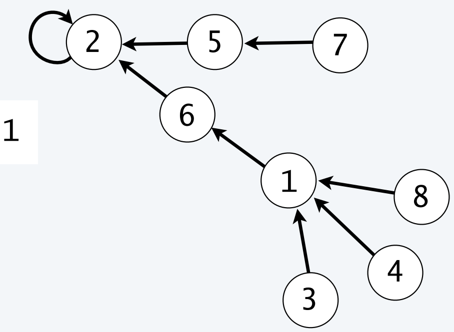
Take $M = N$ and $k = N - 1$ in
$$\frac{1}{(1-z)^M} = \sum_{k\geq 0}\binom{k+M-1}{M-1}z^k$$

Cayley trees

Class	$\mathcal{C}$, the class of labelled rooted unordered trees
EGF	$C(z) = \sum_{c \in \mathcal{C}} \frac{z^{\lvert c \rvert}}{\lvert c \rvert !} \equiv \sum_{N \geq 0} C_N \frac{z^N}{N!}$

Example

6 2 1 1 2 2 5 1



Construction $\quad C = Z \star (SET(C))$ ← "a tree is a root connected to a set of trees"

EGF equation $\quad C(z) = ze^{C(z)}$

Extract coefficients by Lagrange inversion with $f(u) = u/e^u$

$$[z^N]C(z) = \frac{1}{N}[u^{N-1}]\Big(\frac{u}{u/e^u}\Big)^N$$

$$= \frac{1}{N}[u^{N-1}]e^{uN} = \frac{N^{N-1}}{N!}$$

$$C_N = N![z^N]C(z) = N^{N-1} \quad \checkmark$$

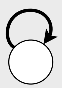
Lagrange Inversion Theorem.

If a GF $g(z) = \sum_{n \geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$ with $f(0) = 0$ and $f'(0) \neq 0$ then $g_n = \frac{1}{n}[u^{n-1}]\Big(\frac{u}{f(u)}\Big)^n$.

Connected components in mappings

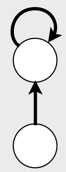
Q. How many different cycles of Cayley trees of size N ?

1

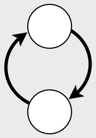


$Y_1 = 1$

1 1
2 2

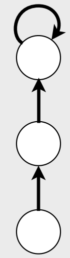


2 1

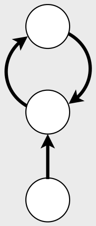


$Y_2 = 3$

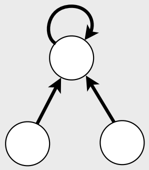
1 1 2
1 3 1
2 2 1
2 3 3
3 1 3
3 2 2



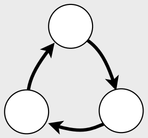
2 1 1
2 1 2
2 3 2
3 1 1
3 3 1
3 3 2



1 1 1
2 2 2
3 3 3



2 3 1
3 1 2



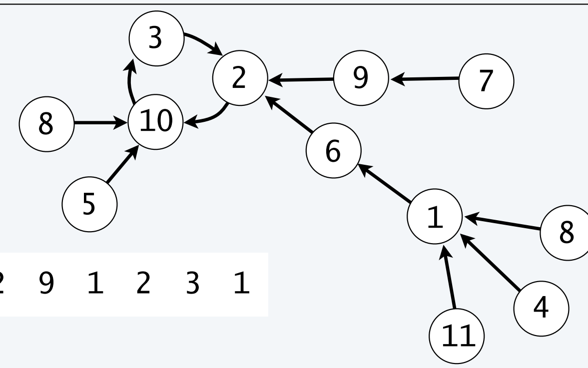
$Y_3 = 17$

A. $\sim \dfrac{N^N\sqrt{\pi}}{\sqrt{2N}}$ (see next slide)

Connected components in mappings

Class	Y, the class of cycles of Cayley trees
EGF	$Y(z) = \sum_{y \in Y} \frac{z^{\lvert y \rvert}}{\lvert y \rvert !} \equiv \sum_{N \geq 0} Y_N \frac{z^N}{N!}$

Example



1 10 2 1 10 2 9 1 2 3 1

Construction $\quad Y = CYC(C) \longleftarrow$ "a component is a cycle of trees"

EGF equation

$$Y(z) = \ln \frac{1}{1 - C(z)}$$

Lagrange Inversion Theorem (Bürmann form).

If a GF $g(z) = \sum_{n \geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$ with $f(0) = 0$ and $f'(0) \neq 0$ then, for any function $H(u)$,

$$[z^n]H(g(z)) = \frac{1}{n}[u^{n-1}]H'(u)\Big(\frac{u}{f(u)}\Big)^n$$

Extract coefficients by Lagrange inversion with $f(u) = u/e^u$ and $H(u) = \ln(1/(1-u))$

$$[z^N]Y(z) = \frac{1}{N}[u^{N-1}]\frac{1}{1-u}e^{uN}$$

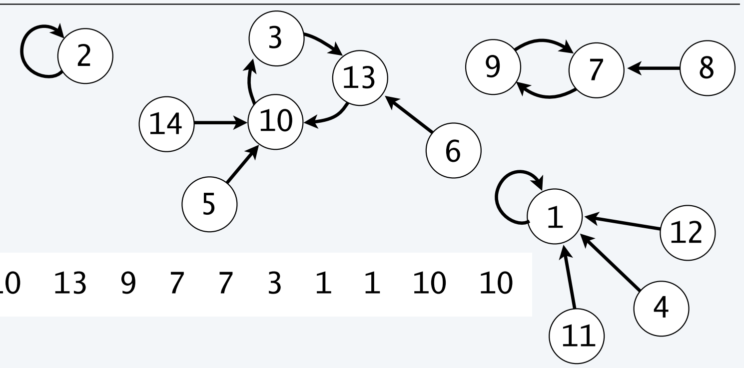
$$= \sum_{0 \leq k < N} \frac{N^{k-1}}{k!} = \sum_{1 \leq k \leq N} \frac{N^{N-k-1}}{(N-k)!}$$

$$Y_N = N![z^N]Y(z) = N^{N-1} \sum_{1 \leq k \leq N} \frac{N!}{N^k (N-k)!} = N^{N-1}Q(N) \sim \frac{N^N\sqrt{\pi}}{\sqrt{2N}}$$

Mappings

Class	*M*, the class of mappings
EGF	$M(z) = \sum_{m \in M} \frac{z^{\lvert m \rvert}}{\lvert m \rvert !} \equiv \sum_{N \geq 0} M_N \frac{z^N}{N!}$

Example



1 2 13 1 10 13 9 7 7 3 1 1 10 10

Construction $M = SET(CYC(C))$ "a mapping is a set of cycles of trees"

EGF equation

$$M(z) = \exp\Big(\ln \frac{1}{1 - C(z)}\Big) = \frac{1}{1 - C(z)}$$

Lagrange Inversion Theorem (Bürmann form).

If a GF $g(z) = \sum_{n \geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$ with $f(0) = 0$ and $f'(0) \neq 0$ then, for any function $H(u)$,

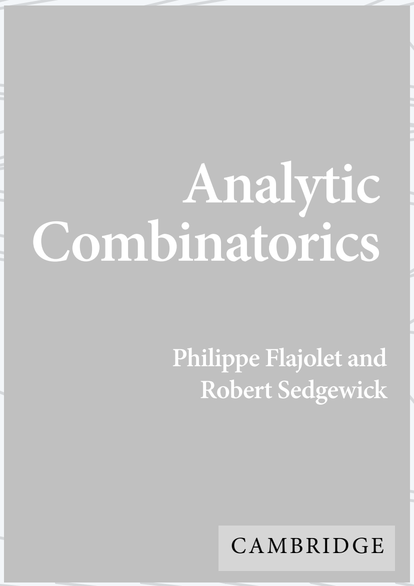
$$[z^n]H(g(z)) = \frac{1}{n}[u^{n-1}]H'(u)\Big(\frac{u}{f(u)}\Big)^n$$

Extract coefficients by Lagrange-Bürmann with $f(u) = u/e^u$ and $H(u) = 1/(1-u)$

$$[z^N]M(z) = \frac{1}{N}[u^{N-1}]\frac{1}{(1-u)^2}e^{uN}$$

$$= \sum_{0 \leq k \leq N} (N-k)\frac{N^{k-1}}{k!} = \sum_{0 \leq k \leq N} \frac{N^k}{k!} - \sum_{1 \leq k \leq N} \frac{N^{k-1}}{(k-1)!} = \frac{N^N}{N!}$$

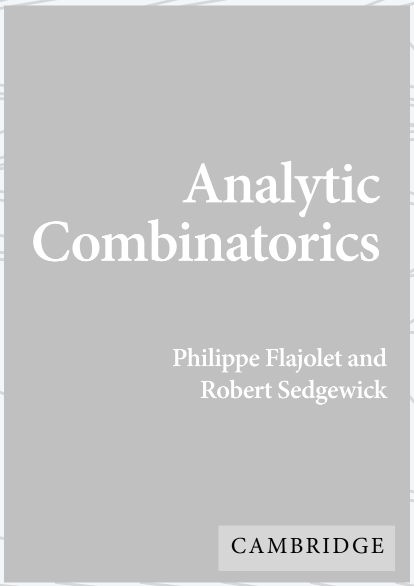
$$M_N = N^N$$



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- **Mappings**



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- Mappings
- **Summary**

The symbolic method for labelled classes (transfer theorem)

Theorem. Let A and B be combinatorial classes of labelled objects with EGFs $A(z)$ and $B(z)$. Then

construction	*notation*	*semantics*	*EGF*
disjoint union	$A + B$	disjoint copies of objects from A and B	$A(z)+B(z)$
labelled product	$A \star B$	ordered pairs of copies of objects, one from A and one from B	$A(z)B(z)$
sequence	$SEQ_k(A)$ or A^k	k- sequences of objects from A	$A(z)^k$
	$SEQ(A)$	sequences of objects from A	$\dfrac{1}{1-A(z)}$
set	$SET_k(A)$	k-sets of objects from A	$A(z)^k/k!$
	$SET(A)$	sets of objects from A	$e^{A(z)}$
cycle	$CYC_k(A)$	k-cycles of objects from A	$A(z)^k/k$
	$CYC(A)$	cycles of objects from A	$\ln\dfrac{1}{1-A(z)}$
boxed product	$A = B^{\square} \star C$	subset of B★C where *smallest* labelled element is from B	$A'(z)=B'(z)C(z)$

Constructions for labelled objects (summary)

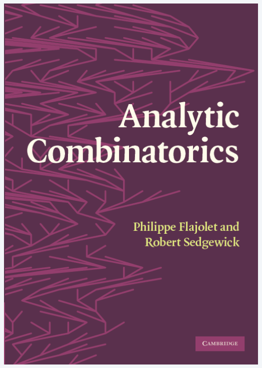
class	*construction*	*EGF*
urns	$U = SET(Z)$	$U(z) = e^z$
cycles	$Y = CYC(Z)$	$Y(z) = \ln \frac{1}{1-z}$
permutations	$P = SEQ(Z)$	$P(z) = \frac{1}{1-z}$
derangements	$D = SET(CYC_{>1}(Z))$	$D(z) = \frac{e^{-z}}{1-z}$
involutions	$I = SET(CYC_{1,2}(Z))$	$I(z) = e^{z+z^2/2}$
words	$W_M = SEQ_M(SET(Z))$	$W_M(z) = e^{Mz}$
surjections	$R = SEQ(SET_{>0}(Z))$	$R(z) = \frac{1}{2-e^z}$
trees	$L = Z \star SEQ(L)$	$L(z) = \ln \frac{1}{1-z}$
Cayley trees	$C = Z \star SET(C)$	$C(z) = ze^{C(z)}$
increasing Cayley trees	$Q = Z^{\square} \star SET(Q)$	$Q'(z) = e^{Q(z)}$
mappings	$M = SET(CYC(C))$	$M(z) = \frac{1}{1-C(z)}$

Analytic combinatorics overview

To analyze properties of a large combinatorial structure:

1. Use the symbolic method
   - Define a *class* of combinatorial objects.
   - Define a notion of *size* (and associated generating function).
   - Use standard operations to develop a *specification* of the structure.

Result: A direct derivation of a GF equation (implicit or explicit).



Important note: GF equations vary widely in nature

$$D(z) = \frac{e^{-z}}{1-z} \qquad I_{\le r}(z) = e^{z+z^2/2+\ldots+z^r/r}$$

$$U(z) = e^z \qquad Q'(z) = e^{Q(z)} \qquad R(z) = \frac{1}{2-e^z} \qquad I(z) = e^{z+z^2/2}$$

$$Y(z) = \ln\frac{1}{1-z} \qquad W_M^{\le b}(z) = (1+z+z^2/2!+\ldots+z^b/b!)^M \qquad L(z) = \ln\frac{1}{1-z}$$

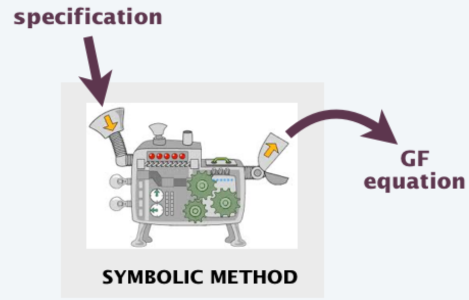
$$P(z) = \frac{1}{1-z} \qquad M(z) = \frac{1}{1-C(z)} \qquad W_M(z) = e^{Mz} \qquad D_{>r}(z) = \frac{e^{-z-z^2/2-\ldots-z^r/r}}{1-z} \qquad C(z) = ze^{C(z)}$$

2. Use complex asymptotics to estimate growth of coefficients (stay tuned).

Direct advantages of the symbolic method

We can *automate* the transfer from specifications to GFs.

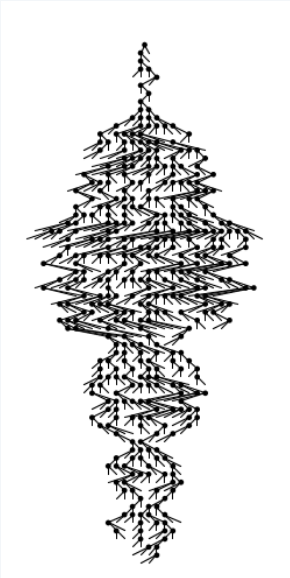
Ref: *Automatic average-case analysis of algorithms.*
by Philippe Flajolet, Bruno Salvy, and Paul Zimmerman (*TCS 1991*).



We can use specifications to *generate random structures.*

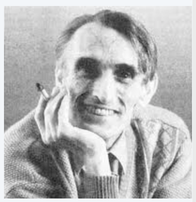
Approach 1: Use a recursive program based on the specification.
Drawback: Requires quadratic time (not useful for large structures).

Approach 2: Use a probabilistic recursive program based on the specification.
Need to settle for *approximate* size *N*.
Can generate large structures in linear time.



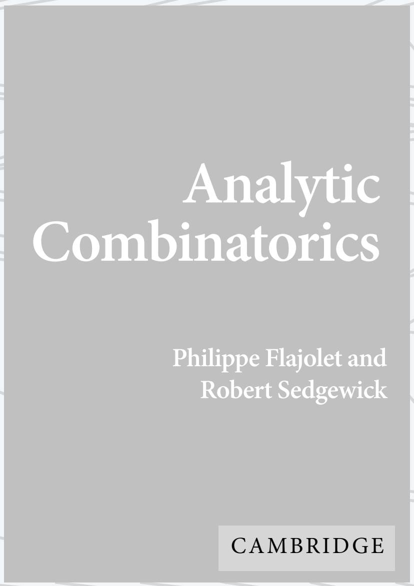
Ref: *Boltzmann samplers for random generation of combinatorial structures.*
by Philippe Duchon, Philippe Flajolet, Guy Louchard and Gilles Schaefer (*CPC 2004*).

French mathematicians on the utility of GFs (continued)



"This approach eliminates virtually all calculations."

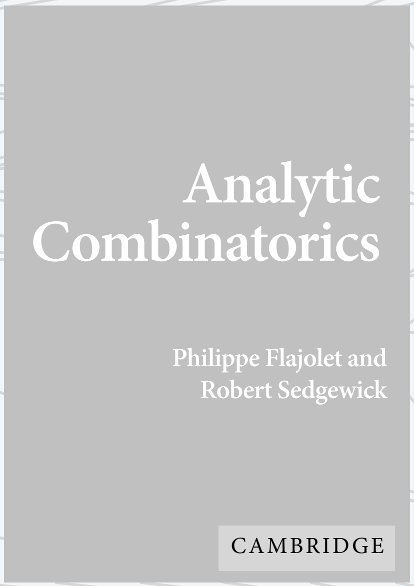
— Dominique Foata & Marco Schützenberger, 1970



http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- Mappings
- **Summary**



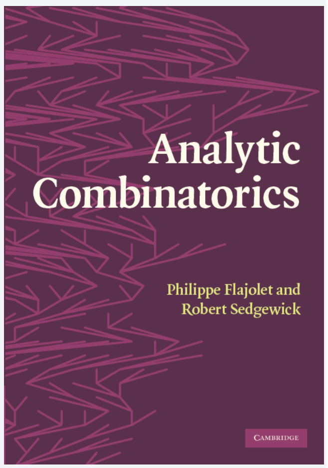
http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- Mappings
- **Exercises**

Note II.11

Ehrenfest model



▷ **II.11.** *Balls switching chambers: the Ehrenfest model.* Consider a system of two chambers A and B (also classically called "urns"). There are N distinguishable balls, and, initially, chamber A contains them all. At any instant $\frac{1}{2}, \frac{3}{2}, \ldots$, one ball is allowed to change from one chamber to the other. Let $E_n^{[\ell]}$ be the number of possible evolutions that lead to chamber A containing ℓ balls at instant n and $E^{[\ell]}(z)$ the corresponding EGF. Then

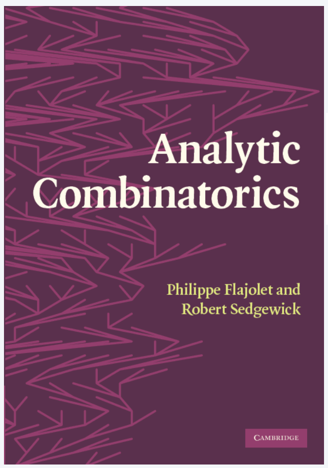
$$E^{[\ell]}(z) = \binom{N}{\ell}(\cosh z)^{\ell}(\sinh z)^{N-\ell}, \qquad E^{[N]}(z) = (\cosh z)^N \equiv 2^{-N}(e^z + e^{-z})^N.$$

[Hint: the EGF $E^{[N]}$ enumerates mappings where each preimage has an even cardinality.] In particular the probability that urn A is again full at time $2n$ is

$$\frac{1}{2^N N^{2n}} \sum_{k=0}^{N} \binom{N}{k}(N-2k)^{2n}.$$

Note II.31

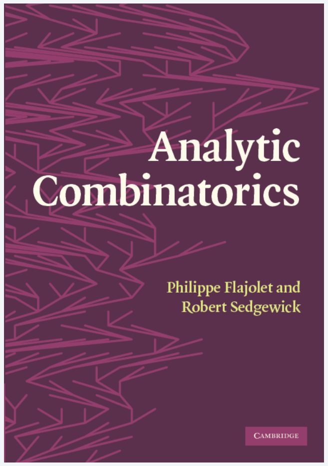
Combinatorics of trigonometrics



▷ **II.31.** *Combinatorics of trigonometrics.* Interpret $\tan \frac{z}{1-z}$, $\tan \tan z$, $\tan(e^z - 1)$ as EGFs of combinatorial classes. ◁

Assignments

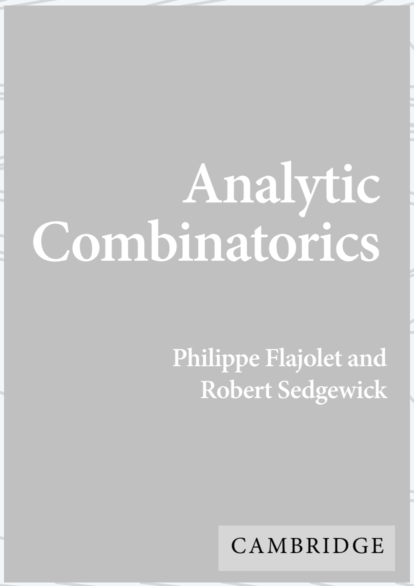
1. Read pages 95-149 (*Labelled Structures and EGFs*) in text.



2. Write up solutions to Notes **II**.11 and **II**.31.

3. Programming exercise (Extra Credit).

Program II.1. Write a program to simulate the Ehrenfest mode (see Note II.11) and use it to plot the distribution of the number of balls in urn A after 10^3, 10^4 and 10^5 steps when starting with 10^3 balls in urn A and none in urn B.



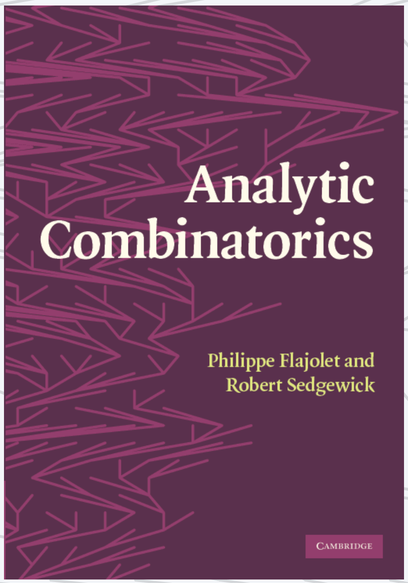
http://ac.cs.princeton.edu

2. Labelled structures and EGFs

- Basics
- Symbolic method for labelled classes
- Words and strings
- Labelled trees
- Mappings
- **Exercises**

ANALYTIC COMBINATORICS

PART TWO



http://ac.cs.princeton.edu

2. Labelled structures and EGFs