Enumerative Combinatoric Algorithms (716.035)

2016

Index

1 Enumerating vs. Counting 3
1.1 Permutation of Letters 3
1.2 Points in Convex Position 3

2 Polyominos 6

3 Pigeonhole-Principle 9
3.1 Example 1: Socks 9
3.2 Example 2: Numbers 9
3.3 Example 3: Hairs 10
3.4 Example 4: Party 10
3.5 Example 5: Numers II 11

4 Inclusion-Exclusion 12
4.1 Spanning trees in ladders 13

5 Polya-Redfield Enumeration Theorem aka Burnside's Lemma 17
5.1 Objects and Operations 17
5.2 Orbits 17
5.3 Stabilizers 17
5.4 Counting Orbits 18
5.5 Examples 18
5.5.1 Strings (1) 18
5.5.2 Strings (2) 19
5.5.3 Strings (3) 20

1 Enumerating vs. Counting

1.1 Permutation of Letters

Example: How many words can we make out of the letters A B C using each letter once?

- ABC
- ACB
- BAC
- BCA
- CAB
- CBA

When we list all objects as above we call it **enumeration**, whereas **counting** is only concerned with the total number of objects. If we consider the example above, how many words would be possible for A B C D?

It's best to find a formula, as using it is a very efficient way to count objects. For $n = 4$ letters we end up with 24 permutations.

The formula for the amount of different words with n letters is n!

1.2 Points in Convex Position

How many crossing-free spanning paths exist for n points on convex position?

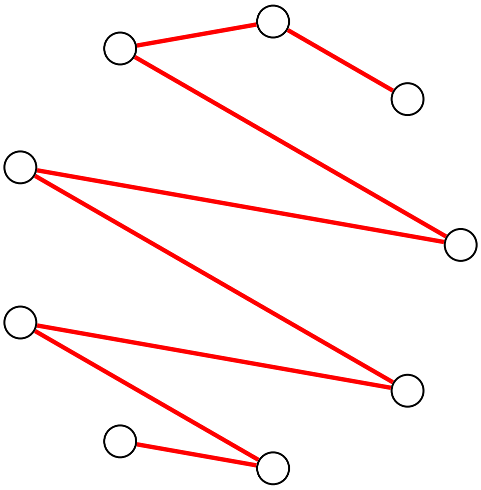


Figure 1: An example illustrating one possibility of a spanning path for $n = 9$ points

For $n = 1$ points the definition of the spanning path is unclear, in some cases it is considered as path with the size 1 and in others with size 0.

Let's look at some examples for $n > 1$ and try to determine a suitable formula.

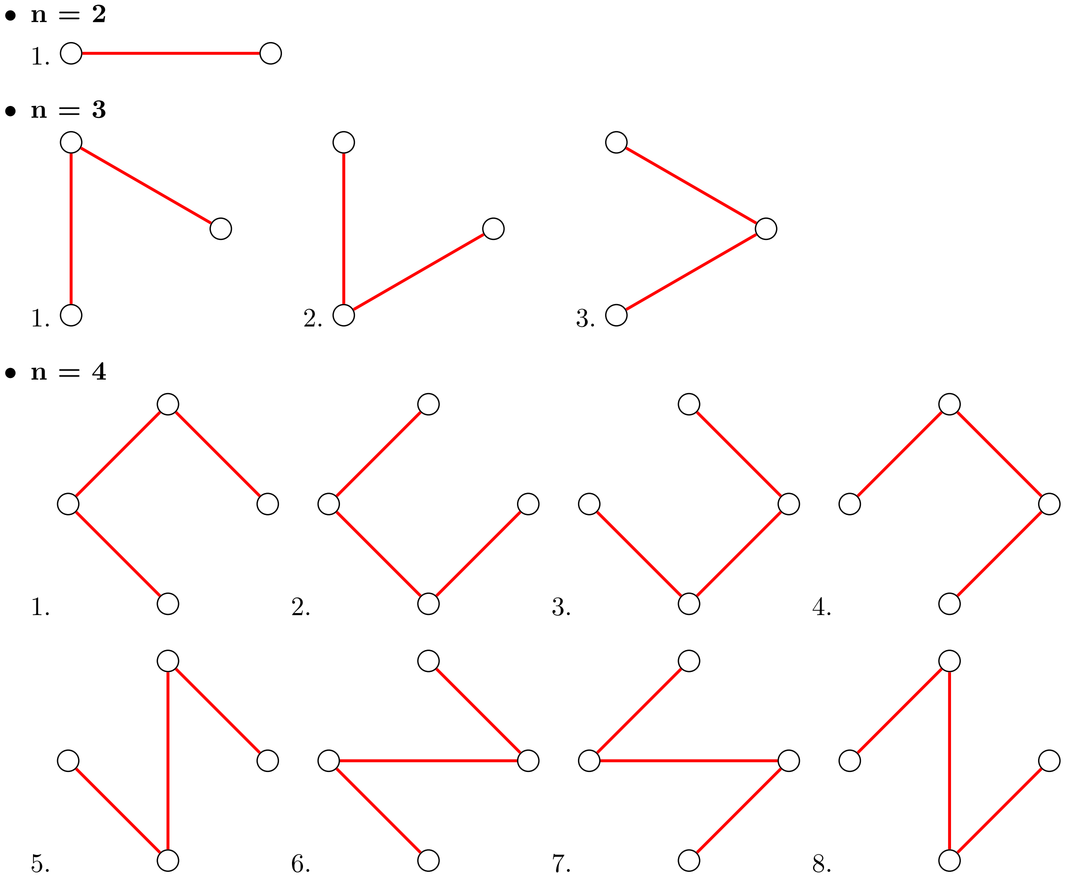


Figure 2: Enumeration of crossing free spanning paths up to $n = 4$

As we can see from this example, enumeration can become a tedious and error prone task very fast. Can you list all paths for $n = 5$?

It is better to abstract the problem and find an inductive solution. When constructing the path we start with a point, and from it we only see two immediate choices. After one of those points is added, we have two choices again. This goes on for a while until $n - 2$.

$$\underbrace{2 \cdot 2 \cdot 2 \cdots 2 \cdot 2}_{n-2 \text{ times}} = 2^{n-2}$$

Now in order to construct all paths we need to start at all possible points, when we do that however a double count occurs.

$$n \cdot 2^{n-2} \Rightarrow \frac{n \cdot 2^{n-2}}{2} \Rightarrow n \cdot 2^{n-3} \text{ for } n \geq 2$$

We can use this formula to find the number of crossing-free spanning paths for $n = 5$, which gives us $5 \cdot 2^2 = 20$ paths.

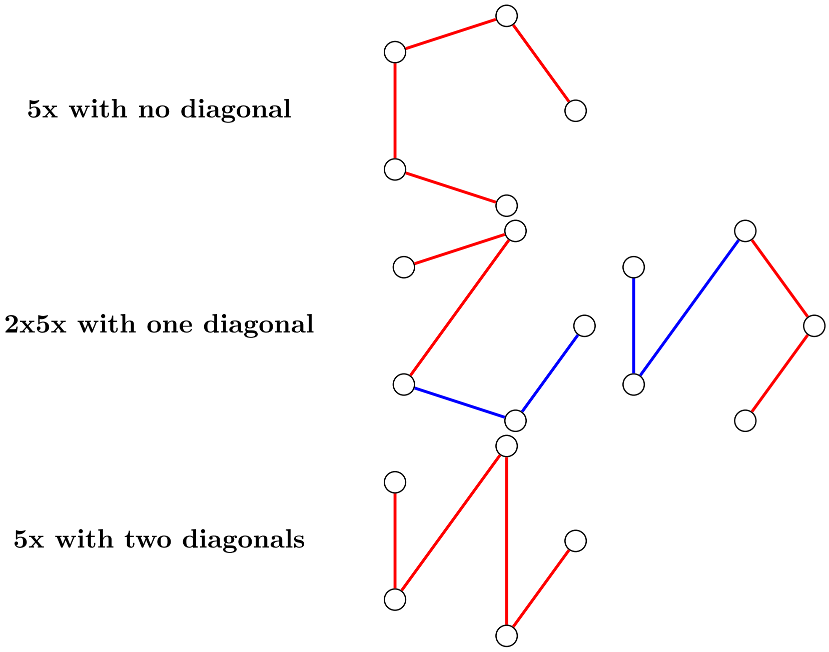


Figure 3: Another method of enumeration, do not explicitly list similar objects

2 Polyominos

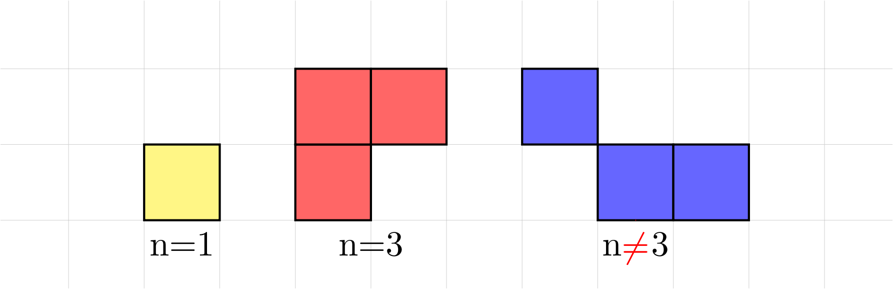


Figure 4: A polyomino of size n consists of n unit squares connected via edges, aligned on a grid

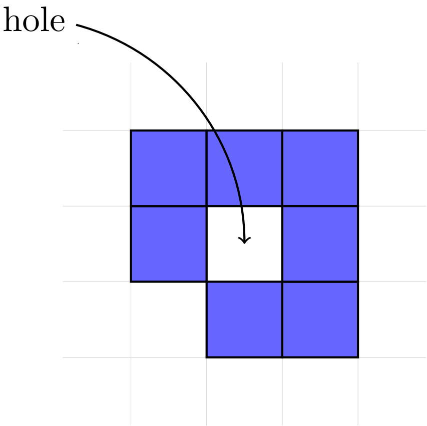


Figure 5: A polyomino with a hole inside. Polyominos without holes are a special case and they are called animals.

How many n-polyominos do exist? We have to define how to count them, that means defining the parameters that decide when two polyominos are regarded as one and the same. Polyominos can be compared using three operations:

1. **Translation**: Move one polyomino on top of another, if they overlap, they are the same.
2. **Rotation**: Additionally rotate one polyomino, if there is one rotation that makes them overlap, they are the same.
3. **Reflection**: Move in the 3^{rd} dimension, mirroring the polyomino.

With these operations polyominos can be classified into these two groups:

- **Fixed polyominos**: Only translation is allowed.
- **Free polyominos**: Translation, rotation and reflection is allowed.

n	# fixed	# free
1	1	1
2	2	1
3	6	2
4	19	5
5	63	12
⋮	⋮	⋮

Figure 6: How many n-polyominos do exist?

What is the formula for generating all n-polyominos? Look at a step from $n \to n+1$, we can add one unit square to all surfaces, in the absolute worst case (a straight polyomino) that means $2n+2$ possibilities. How can we best deal with duplicates?

Approach 1: Generate all new polyominos, then compare them all. For size $n+1$, k polyominos are generated. $\Rightarrow O(\binom{k}{2} \cdot n) = O(k^2 \cdot n)$

Problem: $k >> n$

Approach 2: Fingerprinting. Build a vector of a polyomino that is given by the coordinates of the squares.

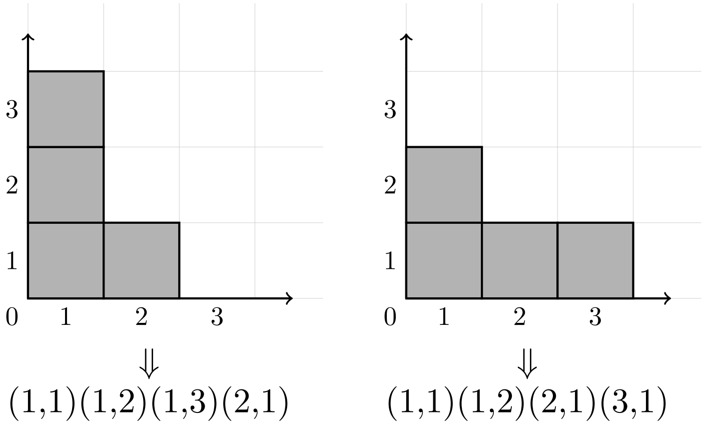


Figure 7: Polynomios and their respective vector fingerprint.

Compute all eight polyonimos (4x rotation & 2x reflection) and take the lexicographical minimum of the vector form!

Time for the fingerprint: $O(n)$

$\Rightarrow$ Compute fingerprints of all generated polyominos: $O(k \cdot n)$

$\Rightarrow$ Sort all fingerprints: $O(k \cdot log(k) \cdot n)$, duplicates are neighboured in the sorting and removing takes $O(k \cdot n)$ $\Rightarrow$ the total runtime is $O(k \cdot log(k) \cdot n)$

No formula is known for the number of fixed/free polyominos.

$$\lim_{n\to\infty} \frac{\#(n+1) \text{ polyominos}}{\#n \text{ polyominos}} = \text{some constant } c$$

The number of polyominos goes to $\Theta(c^n)$ and we know that $4.00253 \leq c \leq 4.65$.

3 Pigeonhole-Principle

Also known as Dirichlet principle, or in German "Schuhfach Prinzip".

If you have $n+1$ elements (pigeons) which you put into n boxes (pigeonholes), then there is a box with at least two elements (two pigeons).

General form: If you have k elements which you put into n boxes, then there is a box with at least $\lceil \frac{k}{n} \rceil$ and one with at most $\lfloor \frac{k}{n} \rfloor$ elements.

3.1 Example 1: Socks

You have n black socks and m white socks in a drawer. How many socks do you have to pick out in order to get a matching pair?

If you abstract the problem using the pigeonhole principle it leads to the following setup. In total there are $n+m$ elements that are distributed upon two boxes, one for white socks and one for black socks. If we pick a sock out of the drawer and put it into one of the two boxes, we have to repeat the process at least three times to have a box with two elements in them (a matching pair of socks).

For $n=2$ boxes there have to be $n+1=3$ elements, such that one of the boxes contains at least $\lceil \frac{3}{2} \rceil = 2$ elements.

3.2 Example 2: Numbers

Let S be a subset of $\{1, 2, \cdots, 2n\}$ of cardinality $n+1$. Prove or give a counter example for: S contains two numbers a and b such that

- $a+b=2n+1$
- $a-b=n$

To solve this problem we have to think of appropriate labels for the boxes.

$$\left.\begin{array}{c} \{1, 2n\} \\ \{2, 2n-2\} \\ \{3, 2n-3\} \\ \vdots \\ \{n, n+1\} \end{array}\right\} n \text{ boxes}$$

There is a subset of cardinality $n+1$, so taking a number and putting it into the box with its name on the label, there has to be at least one box with two elements, thus adding up to $2n+1$.

Similarly, solve the second problem by using different labels

$$\left.\begin{array}{c}\{1, n+1\}\\ \{2, n+2\}\\ \{3, n+3\}\\ \vdots \\ \{n, \underbrace{n+n}_{2n}\}\end{array}\right\} n \text{ boxes}$$

3.3 Example 3: Hairs

Prove: In Austria there exist at least eight people with the same number of hairs on their head.

Facts:

- On average people have 150.000 hairs
- It is save to assume that the range is $0 - 1.000.000$
- Population of Austria: 8.699.730 (Jan 1^{st} 2016)

Take the total range of hairs as boxes, leading to 1.000.001 boxes. At least $\lceil\frac{8699730}{1000001}\rceil \sim \lceil 8,7\rceil = 9$ people have the same number of hairs on their head in Austria.

3.4 Example 4: Party

At a party $n \geq 2$ people meet. Some shake hands with others and some don't. Show or provide a counter example:

- There are at least two people at this party shaking the same number of hands.

There are two cases to consider for the boxes.

1. No one shakes zero hands.

   # handshakes

$$\left.\begin{array}{l}1\\ 2\\ 3\\ \vdots \\ n-1\end{array}\right\} n-1 \text{ boxes}$$

2. At least one person shakes zero hands

$$\begin{array}{l}\#\text{ handshakes}\\ \left.\begin{array}{l}0\\ 1\\ 2\\ \vdots\\ n-2\end{array}\right\}\ n-1\text{ boxes}\end{array}$$

In both cases we have $n-1$ boxes for n elements, meaning there always has to be a box with at least two elements.

3.5 Example 5: Numers II

Let q be an odd number and let S be the set $\{1, 3, 7, 15, 31, \cdots, 2^i - 1\}$, i.e. S contains the elements $a_i = 2^i - 1$ for all $i \geq 1$.

Claim: For each odd number q there exists an a_i which is a multiple of q (q divides a_i without rest)

$$a_i = c \cdot q + r_i \qquad r_i \text{ has to be zero}$$

Boxes for the rest:

$$\left.\begin{array}{l}\{r_i = 0\}\\ \{r_i = 1\}\\ \{r_i = 2\}\\ \qquad\vdots\\ \{r_i = q-2\}\end{array}\right\}\ q\text{ boxes}$$

We can ignore the case with at least one element in box $\{r_i = 0\}$, since it means we are done already. If we ignore that box, there have to be two boxes with the same rest.

$$\begin{array}{rl} & a_m = c^* \cdot q + r_m\\ - & \underline{a_m = c^{**} \cdot q + r_m}\\ & a_m - a_n = c^{***} \cdot q \qquad m > n\\ & a_m - a_n = 2^m - 1 - (2^n - 1) =\\ & 2^m - 2^n = 2^n \underbrace{(2^{m-n} - 1)}_{a_{m-n}}\end{array}$$

a_{m-n} must be a multiple of q.

4 Inclusion-Exclusion

Example: How many numbers of the set $\{1 \dots 45\}$ are coprime (do not have a common divisor) with 45?

We can use the following equation to find a solution.

$$X = 45 - |\{\text{numbers which have at least one common divisor with } 45\}|$$

For that we need to take a look at the prime coefficients of 45 which consist of $3 \cdot 3 \cdot 5$ and count the members of the set divisible by them.

$$\begin{aligned} X_i &= |\text{numbers of } \{1 \dots 45\} \text{ divisible by i}| \\ X_3 &= \frac{45}{3} = 15 \\ X_5 &= \frac{45}{15} = 9 \\ X_{3 \cdot 5} = X_{15} &= \frac{45}{15} = 3 \end{aligned}$$

In the last line, we made preparations to ensure that no double count occurs.

$$X = 45 - X_3 - X_5 + X_{3 \cdot 5} = 45 - 15 - 9 + 3 = 24$$

Lemma: For two sets A, B we are interested in $|A \cup B|$

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Example 2: How many numbers of the set $\{1 \dots 60\}$ are coprime with 60?

$$\begin{aligned} &60 = 2 \cdot 2 \cdot 3 \cdot 5 \\ &X = 60 - \underbrace{X_2}_{30} - \underbrace{X_3}_{20} - \underbrace{X_5}_{12} + \underbrace{X_6}_{10} + \underbrace{X_{10}}_{6} + \underbrace{X_{15}}_{4} - \underbrace{X_{30}}_{2} \\ &X = 16 \end{aligned}$$

Lemma: For three sets A, B, C we are interested in $|A \cup B \cup C|$

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

4.1 Spanning trees in ladders

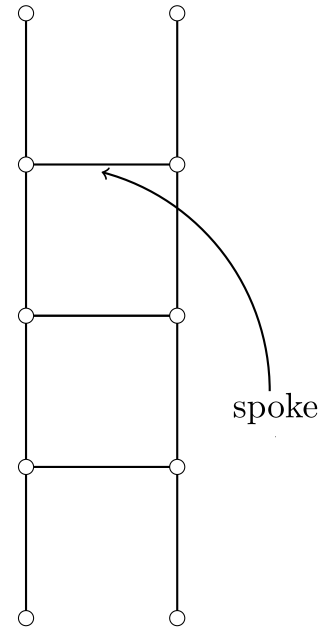
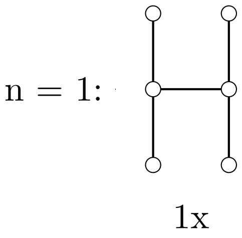
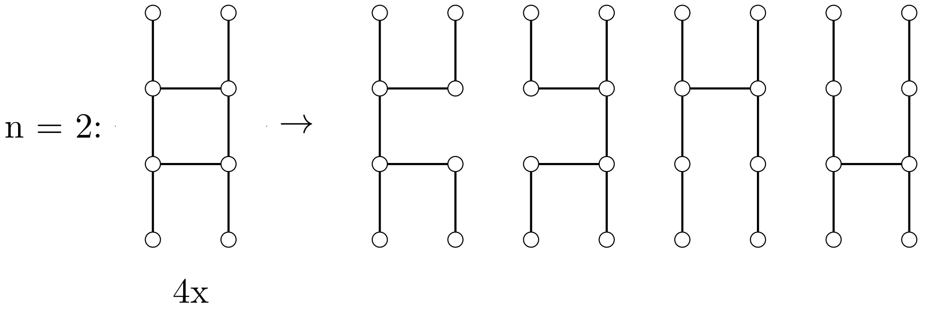
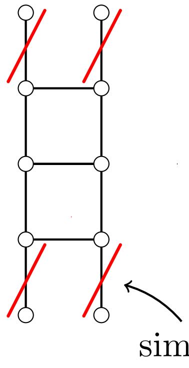


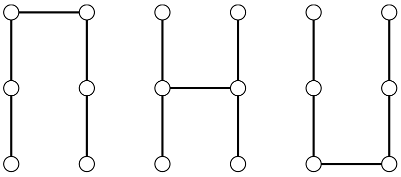
Figure 8: Depiction of a ladder as a connected, cycle-free spanning tree. A ladder of size n has n spokes.

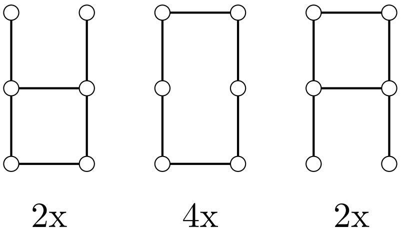
How many spanning trees does a ladder of size n contain as subgraphs (with the same vertex set)?

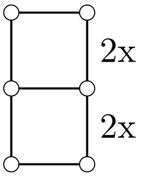




n = 3:  ? → We need to remove 2 edges!

Removing 2 spokes:  = 3x

Removing 1 spoke:  = 8x

Removing 0 spokes:  = 4x

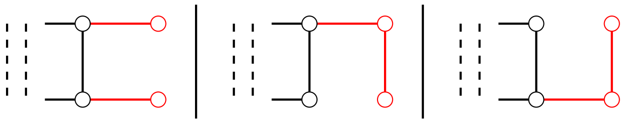
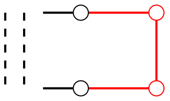
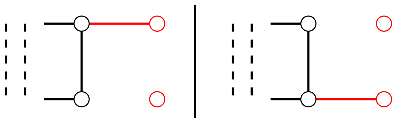
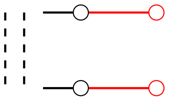
For the total number of subgraphs with a ladder of size $n = 3$ we can calculate
$3 + 8 + 4 = 15$

How many spanning trees does a ladder of size $n = 4$ contain as subgraphs? How can we find a formula for this problem? We need to make some definitions!

$X_A(n) \ldots \#$ of spanning trees in a ladder of size n

$X_B(n) \ldots \#$ of spanning graphs which consist of $\underline{2}$ components and are cycle free for a ladder of size n such that the rightmost 2 vertices are in <u>different</u> components.

We can then build the following table:

$n \to n+1$	start with $X_A(n)$	start with $X_B(n)$
$X_A(n+1)$	 3x	 1x
$X_B(n+1)$	 2x	 1x

We can translate this into a matrix.

$$X_A(1) = 1$$
$$X_B(1) = 1$$

$$\begin{pmatrix} X_A(n+1) \\ X_B(n+1) \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} X_A(n) \\ X_B(n) \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix}^n \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\text{for n: } \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix}^{n-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

However, we are only interested in $X_A(n)$ and want to get $X_B(n)$ out of the equation.

$$\begin{aligned} X_A(n+1) &= 3X_A(n) + X_B(n) \to X_B(n) = X_A(n+1) - 3X_A(n) \\ X_B(n+1) &= 2X_A(n) + X_B(n) \to \underline{X_B(n)} = 2X_A(n-1) + X_B(n-1) \\ &\to 2X_A(n-1) + X_A(n) - 3X_A(n-1) \\ &= \underline{X_A(n) - X_A(n-1)} \end{aligned}$$

$$X_A(n+1) = 3X_A(n) + X_A(n) - X_A(n-1) = \underline{4X_A(n) - X_A(n-1)}$$

With this recursive formula we can solve the problem easily.

$$\begin{aligned}
&X_A(1) = 1\\
&\underline{X_A(2) = 4}\\
&X_A(3) = 4 \cdot 4 - 1 = 15\\
&X_A(4) = 4 \cdot 15 - 4 = 56\\
&\qquad\vdots
\end{aligned}$$

But, as the formula has a recursion we always need to know the last two elements. We can use linear algebra or aspects of generating functions to get rid of this recursion!

$$X_A(n) = \frac{1}{2 \cdot \sqrt{3}}\,(2+\sqrt{3})^n - \frac{1}{2 \cdot \sqrt{3}}\,(2-\sqrt{3})^n$$

Figure 9: Non-recursive formula for the number of spanning trees a ladder of size n contain as subgraphs

The Asymptotics are: $\Theta((\underbrace{2+\sqrt{3}}_{3{,}73})^n)$

5 Polya-Redfield Enumeration Theorem aka Burnside's Lemma

5.1 Objects and Operations

Set of **objects** X
Set of n **operations** $R = \{R_i, 0 \leq i \leq n-1\}$

R forms a group w.r.t. $\circ$ (4 axioms $\forall R_i, R_j$) :
Closure: $\exists R_k : R_i \circ R_j = R_k$
Associativity: $(R_i \circ R_j) \circ R_k = R_i \circ (R_j \circ R_k)$
R_0 is identity element, i.e, $R_0 \circ R_i = R_i \circ R_0 = R_i$
Inverse element: $\exists R_k : R_k \circ R_i = R_i \circ R_k = R_0$

Objects might be strings, colored grids, colored cubes, ...
Operations might be rotations, reflections, ...
For objects, R_0 is the identity (neutral) function.

5.2 Orbits

An orbit is the set of all objects from X which can be transformed into each other by an operation from R (equivalence class). The length of an orbit is the number of elements it contains.

Main Question: How many Orbits exist?

5.3 Stabilizers

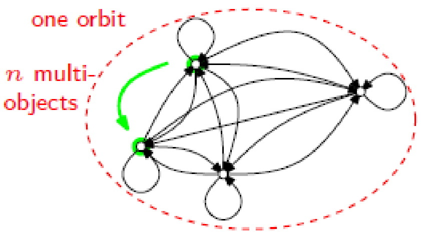
For an object $x \in X$ the operation $R_i \in R$ is a **stabilizer**
if and only if $R_i(x) = x$.

m_x ... number of stabilizers for x
r_i ... invariance number, i.e., number of objects for which R_i is a stabilizer.

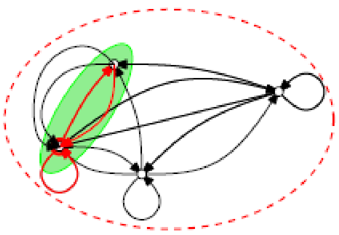
We have $\sum\limits_{i=0}^{n-1} r_i = \sum\limits_{x \in X} m_x$ by double counting (each pair operation/object which gives an invariance is counted once on each side; left side counts by operation, right side counts by objects).

5.4 Counting Orbits

Deriving a first relation simplified version (not valid in full generality):
$\sum\limits_{x \in orbit} m_x = n$



Two times the same object: unify the vertices!
For removed copy: remove outgoin edges, reroute incoming edges.
We get one new stabilizer!



This leads to the formular:
number of orbits $= \frac{\sum\limits_{x \in X} m_x}{n}$
With the previous observation $\sum\limits_{i=0}^{n-1} r_i = \sum\limits_{x \in X} m_x$ we therefore get

$$\text{number or orbits} = \frac{\sum\limits_{i=0}^{n-1} r_i}{n}$$

We will use this equation, as usually $n << |X|$ and thus, all invariance numbers r_i are easier to obtain than all numbers m_x of stabilizers.

5.5 Examples

5.5.1 Strings (1)

Consider strings of length 4 with characters A, B
Operation: cyclic shift by i positions, $0 \leq i \leq 3$. How many different strings exist?

Algorithm:
1. Identify R_0 to R_{n-1}
2. Compute all invariance numbers $r_i, 0 \leq i \leq n-1$
3. Compute the number of orbits $\frac{\sum\limits_{i=0}^{n-1} r_i}{n}$

We have 4 shift operations, therefore we have R_0 to R_3
0 Shift: $r_0 = 2^4 = 16$: 2 characters for 4 positions
1 Shift: $r_1 = 2 : |A|A|A|A|$ or $|B|B|B|B|$
2 Shifts: $r_2 = 4 : |A|A|A|A|$, $|B|B|B|B|$, $|A|B|A|B|$ or $|B|A|B|A|$

3 Shifts: $r_3 = 2$: The same as for r_1

number of orbits $= \frac{16+2+4+2}{4} = \frac{24}{4} = \mathbf{6}$

But how? - write down the 16 possibilities (R_0) and group them in groups of the same meanings. That means that the string is basically the same but shifted.

AAAA ... Yellow is Orbit 1
AAAB ... Red is Orbit 2
AABA
ABAA
BAAA
AABB ... Green is Orbit 3
ABAB ... Orange is Orbit 4
ABBA
BAAB
BABA
BBAA
ABBB ... Gray is Orbit 5
BABB
BBAB
BBAB
BBBA
BBBB ... Pink is Orbit 6

There we can clearly see the 6 different orbits.

5.5.2 Strings (2)

This is the same example as Example String (1), but with 3 characters A, B, C
$r_0 = 3^4 = 81$, which determines all possible ways the characters can be set
$r_1 = r_3 = 3$, which determines the 3 ways you can set the characters when shifting by 1 or 3 characters and the meaning stays the same. Those are: $|A|A|A|A|$, $|B|B|B|B|$ and $|C|C|C|C|$
$r_2 = 9$. The basic idea behind this is that you can shift 2 times when a string consist of only 1 character like $|A|A|A|A|$ or has the form $|X|Y|X|Y|$ or $|Y|X|Y|X|$. With 3 available characters we get those 9 ways.

This gives us the number of orbits: $\frac{96}{4} = 24$

5.5.3 Strings (3)

This is the same example as Example String (2), but you have to use every character atleast once. $r_0 = 36$
$r_1 = 0$
$r_2 = 0$
$r_3 = 0$

Why is it 36?
We know when every character has to be there 1 time, one character has to be there 2 times.
So we have 3 characters that can be chosen twice.
Then we have 2 characters left from the 4 character long string.
At the end we have 2 possibly ways to fill the 2 characters which are left. For example: $|B|A|C|A|$ and $|C|A|B|A|$

Therefore we get $3 \times \binom{4}{2} \times 2$
$\binom{4}{2} = 6$ so we have $3 \times 6 \times 2 = 36$

This gives us the number of orbits: $\frac{36+0+0+0}{4} = 4$