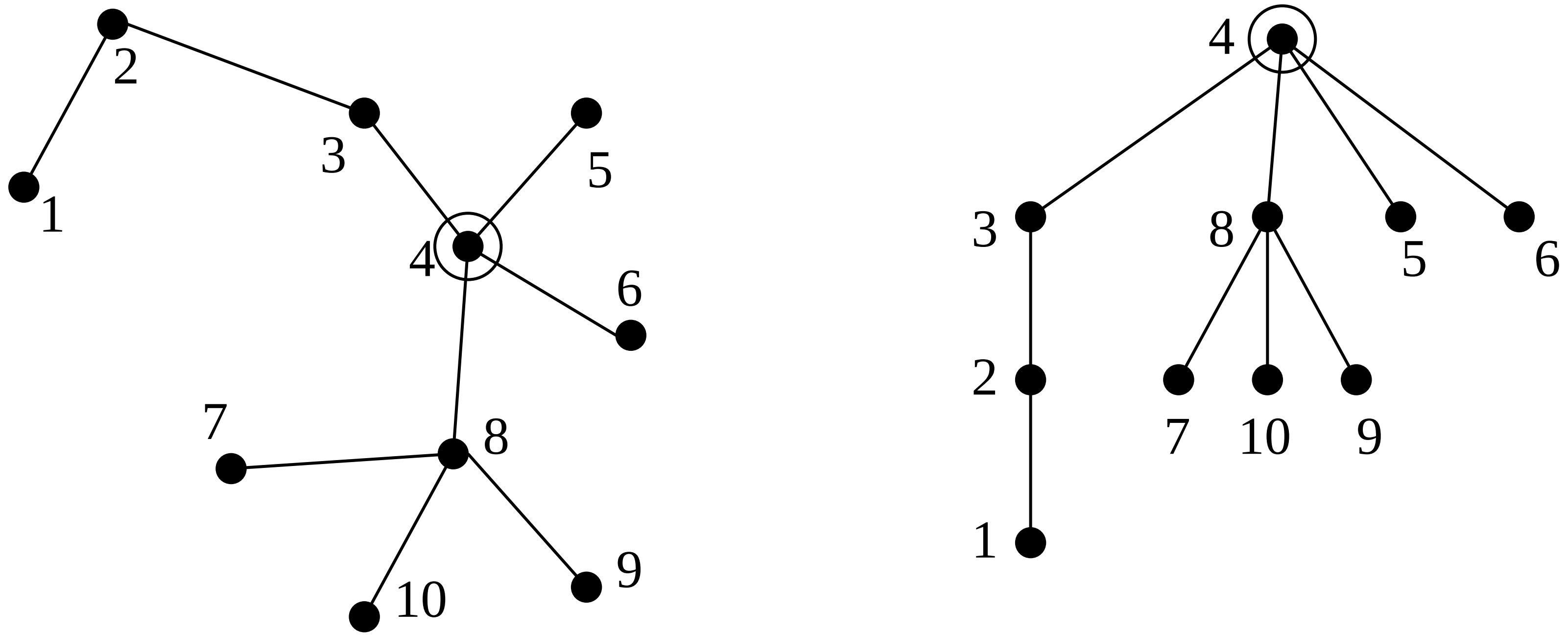
Topics in trees and Catalan numbers

See Chapter 8.1.2.1. These slides have more details than the book.

Prof. Tesler

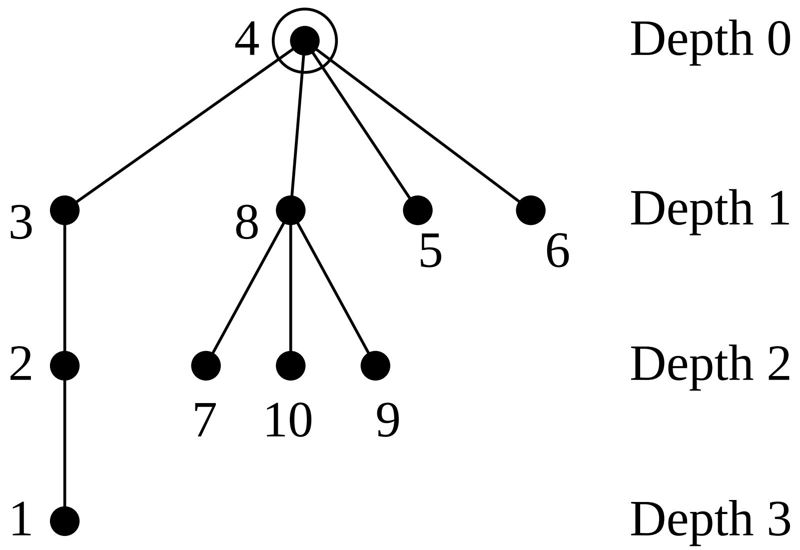
Math 184A
Winter 2019

Rooted trees



- A *rooted tree* is a tree with one vertex selected as the *root*.
- It can be drawn in any manner, but it is common to put the root at the top and grow the tree down in levels (or at the left and grow it rightward in levels, etc.).
- The *leaves* are the vertices of degree 0 or 1: 1,5,6,7,10,9.
- The *internal vertices* (or *internal nodes*) are the vertices of degree > 1: 2,3,4,8.

Rooted trees: relationships among vertices



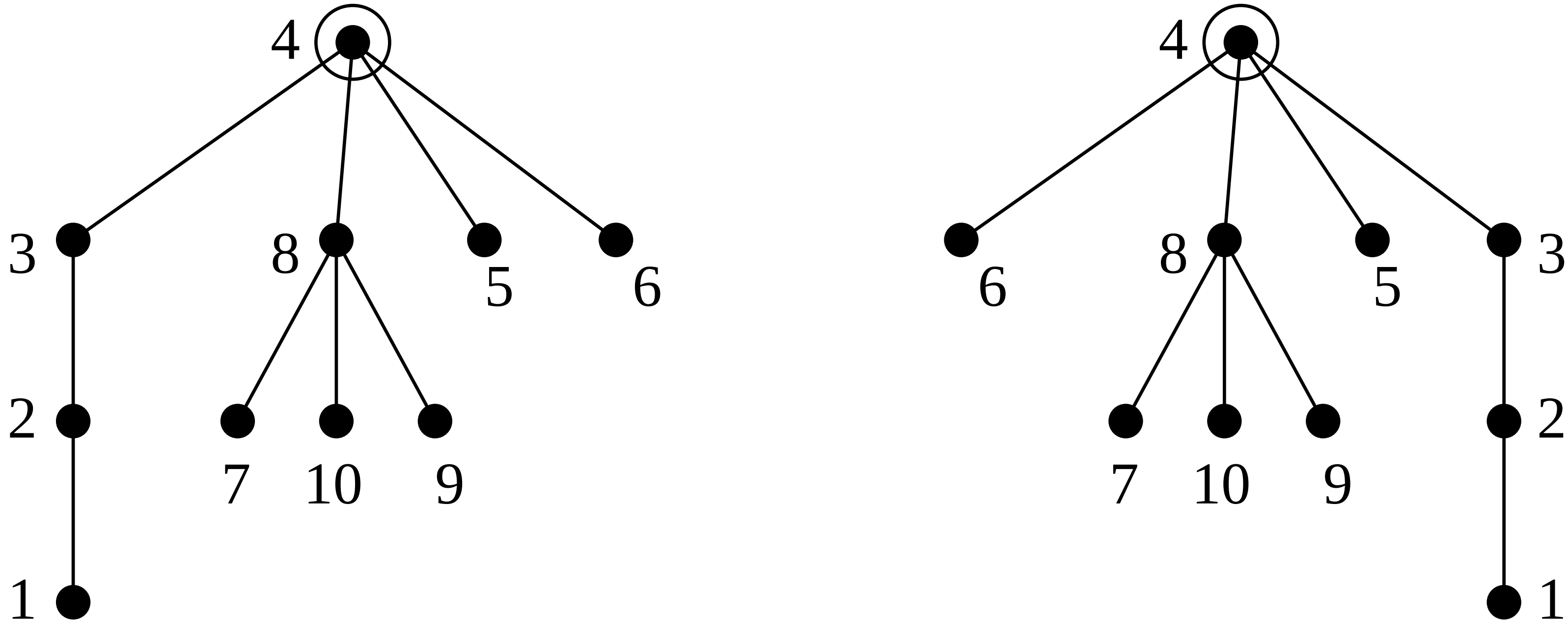
Parents/children

- 4 has *children* 3, 8, 5, 6.
  3, 8, 5, 6 each have *parent* 4.
  3, 8, 5, 6 are *siblings*.
- 8 has parent 4 and children 7, 10, 9.

Depth

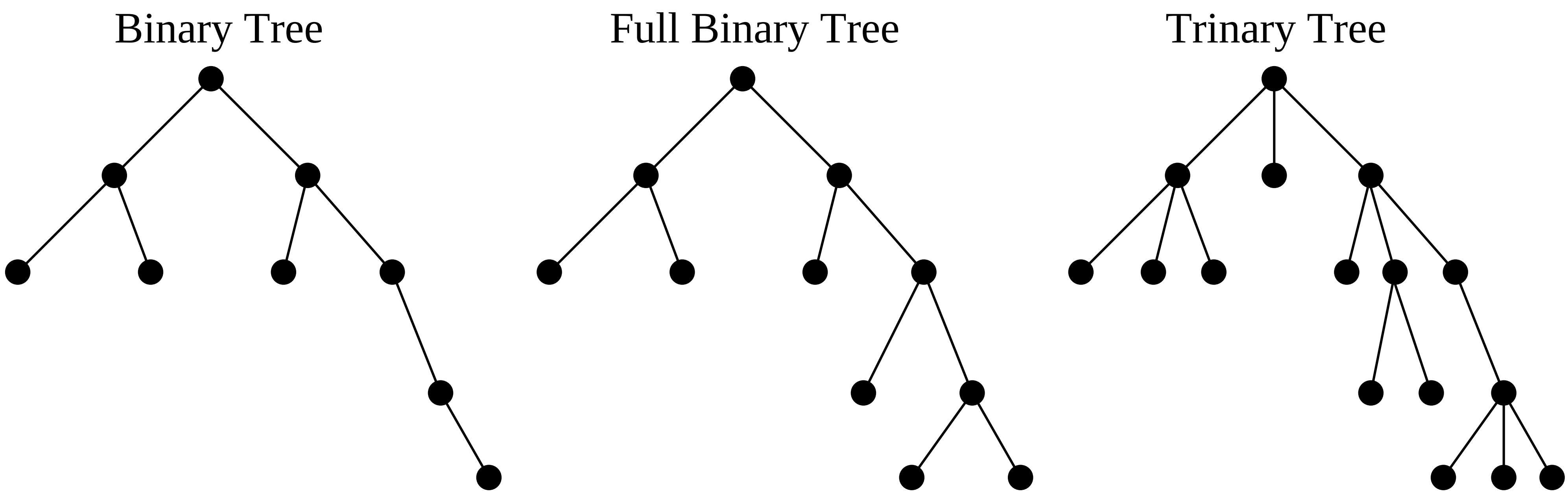
- The *depth* of v is the length of the path from the root to v.
- The *height* of the tree is the maximum depth.

Ordered trees



- An *ordered tree* puts the children of each node into a specific order (pictorially represented as left-to-right).
- The diagrams shown above are the same as *unordered trees*, but are different as *ordered trees*.

Binary trees and more



- In a *k-ary tree*, every vertex has *between* 0 and k children.
- In a *full k-ary tree*, every vertex has *exactly* 0 or k children.
- *Binary*=2-ary, *Trinary*=3-ary, etc.
- *Lemma:* A full binary tree with n leaves has $n-1$ internal nodes, hence $2n-1$ vertices and $2n-2$ edges in total.

Dyck words

A *Dyck word* (pronounced "Deek") is a string of n 1's and n 2's such that in every prefix, the number of 1's $\geqslant$ the number of 2's.

Example ($n = 5$): 1121211222

Prefix	# 1's	# 2's	#1's $\geqslant$ # 2's
$\emptyset$	0	0	$0 \geqslant 0$
1	1	0	$1 \geqslant 0$
11	2	0	$2 \geqslant 0$
112	2	1	$2 \geqslant 1$
1121	3	1	$3 \geqslant 1$
11212	3	2	$3 \geqslant 2$
$\ldots$			
1121211222	5	5	$5 \geqslant 5$

Dyck words for $n = 2$

- There are $\binom{4}{2} = 6$ strings of two 1's and two 2's:
  1122 1212 1221 2112 2121 2211
- Which of them are Dyck words?
- 1122 and 1212 both work.
- 1221 fails: In the first 3 characters, there are more 2's than 1's.
- 2112, 2121, and 2211 fail:
  In the first character, there are more 2's than 1's.

Dyck words and Catalan numbers

Let W_n be the set of all Dyck words on n 1's and n 2's, and $C_n = |W_n|$ be the number of them.

	W_0	W_1	W_2	W_3
	$\emptyset$	12	1122	111222
			1212	112122
				112212
				121122
				121212
$C_n = \|W_n\|$	1	1	2	5

Catalan numbers $C_n = |W_n|$

- One formula (to be proved later) is
$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{n!\,(n+1)!}$$
- $C_3 = \frac{1}{4}\binom{6}{3} = \frac{20}{4} = 5$

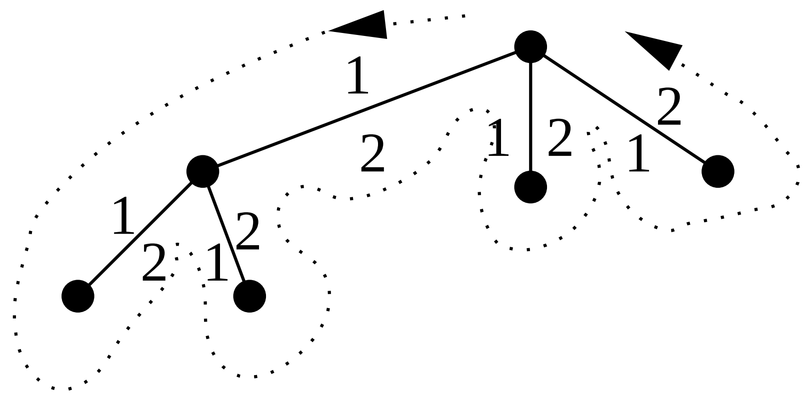
Balanced parentheses

Replacing $1 = ($ and $2 =)$ gives n pairs of balanced parentheses:

	W_0	W_1	W_2	W_3
	$\emptyset$	$12 = ()$	$1122 = (())$	$111222 = ((()))$
			$1212 = ()()$	$112122 = (()())$
				$112212 = (())()$
				$121122 = ()(())$
				$121212 = ()()()$
$C_n = \|W_n\|$	1	1	2	5

Ordered trees

$w = 1121221212$



- Given Dyck word w, form an ordered tree as follows:
  - Draw the root.
  - Read w from left to right.
    For 1, add a new rightmost child to the current vertex and move to it.
    For 2, go up to the parent of the current vertex.
- For any prefix of w with a 1's and b 2's, the depth of the vertex you reach is $a - b \geqslant 0$, so you do not go "above" the root.
  At the end, $a = b = n$ and the depth is $a - b = 0$ (the root).
- Conversely, trace an ordered tree counterclockwise from the root.
  Label each edge 1 going down its left side, and 2 going up its right.
- Thus, W_n is in bijection with ordered trees on n edges (hence $n + 1$ vertices), so C_n counts these too.

Recursion for Catalan numbers

- For $n > 0$, any Dyck word can be uniquely written $u = 1x2y$ where x, y are smaller Dyck words:

$$u = 1\,1212\,2\,112212 \in W_6 \qquad x = 1212 \in W_2 \qquad y = 112212 \in W_3$$

$$(\,()()\,)\,(())() \qquad \text{or } x = ()() \qquad y = (())()$$

- For $u \in W_n$ (with $n > 0$), this decomposition gives $x \in W_i$, $y \in W_{n-1-i}$ where $i = 0, \ldots, n-1$.

- We have $C_0 = 1$ and recursion $\displaystyle C_n = \sum_{i=0}^{n-1} C_i\, C_{n-1-i} \quad (n > 0)$.

- $C_0 = 1$
  $C_1 = C_0 C_0 = 1 \cdot 1 = 1$
  $C_2 = C_0 C_1 + C_1 C_0 = 1 \cdot 1 + 1 \cdot 1 = 2$
  $C_3 = C_0 C_2 + C_1 C_1 + C_2 C_0 = 1 \cdot 2 + 1 \cdot 1 + 2 \cdot 1 = 5$

Complete binary parenthesization

- What are all ways to parenthesize a product of n letters so that each multiplication is binary?
  E.g., $(a(bc))(de)$ uses only binary multiplications, but $(abc)(de)$ is invalid since abc is a product of three things.
- For a product of n letters, we have $n-1$ binary multiplications:

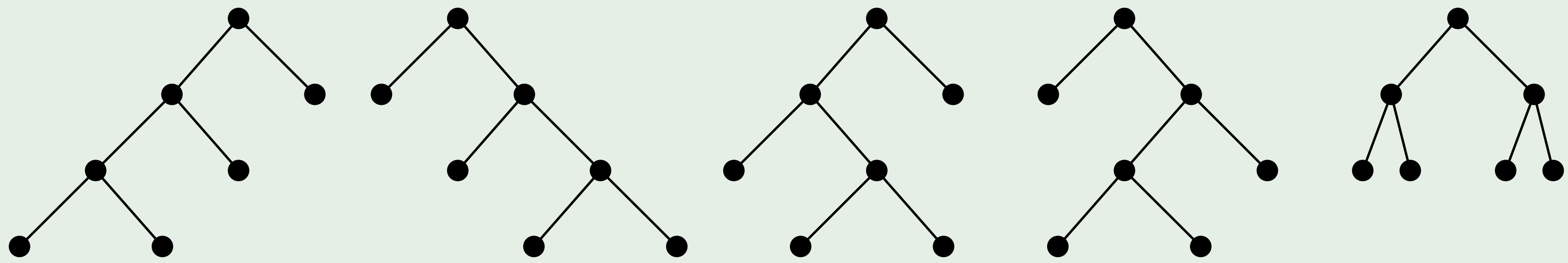
	$n=1$	$n=2$	$n=3$	$n=4$
	a	ab	$a(bc)$	$((ab)c)d$
			$(ab)c$	$(a(bc))d$
				$a((bc)d)$
				$a(b(cd))$
				$(ab)(cd)$
Count	1	1	2	5

Complete binary parenthesization

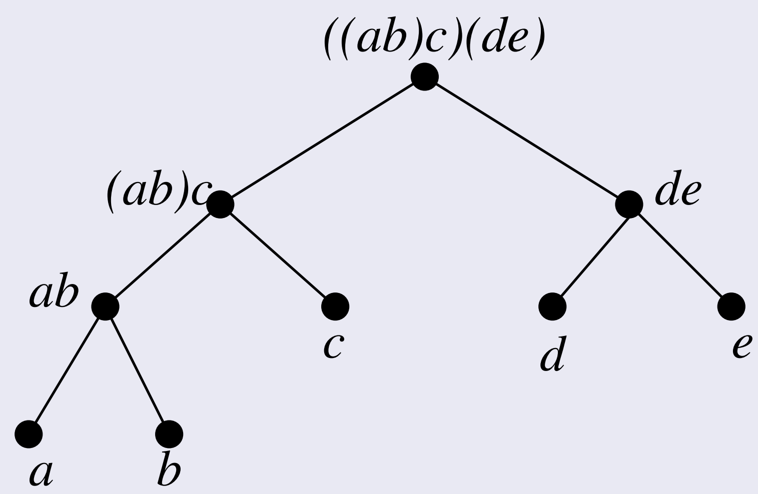
- To parenthesize a product of n letters:
  - Let x be a complete binary parenthesization of the first i letters.
  - Same for y on the other $j = n - i$ letters.
  - Form $(x)(y)$.
    If x or y consists of only one letter, omit the parentheses around it.
- The possible values of i are $i = 1, 2, \ldots, n - 1$.
- Let b_n = # complete binary parenthesizations of n letters. Then
  $$b_1 = 1 \qquad b_n = \sum_{i=1}^{n-1} b_i \, b_{n-i}$$
- Similar recursion to Catalan numbers, but n is shifted: $b_n = C_{n-1}$.

Ordered full binary trees

Ordered full binary trees with $n = 4$ leaves

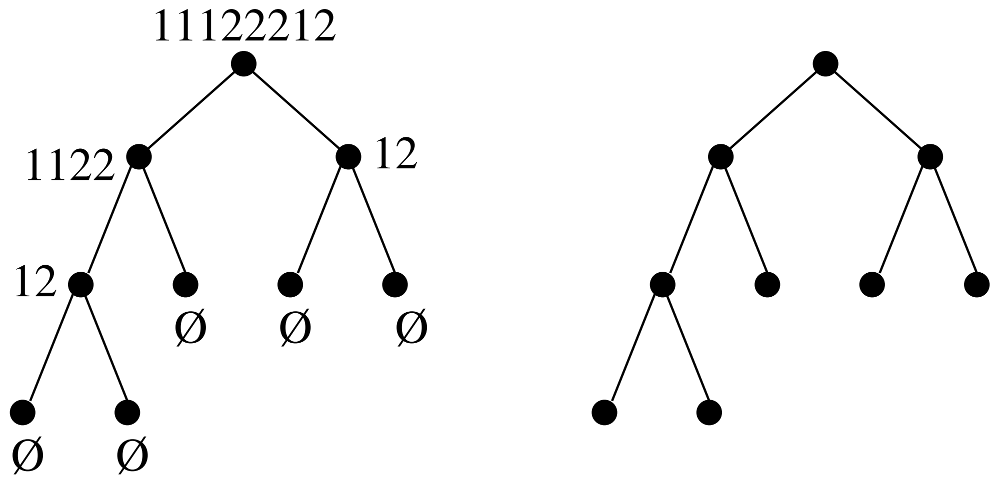


Bijection: ordered full binary trees with n leaves $\leftrightarrow$ complete binary parenthesizations of n factors



Each internal node is labelled by the product of its children's labels, with parentheses inserted for factors on more than one letter.
There are C_{n-1} full binary trees with n leaves.

Bijection: Dyck words ↔ ordered full binary trees

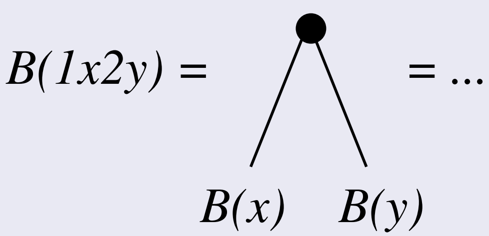


Bijection: Dyck words $W_n \leftrightarrow$ binary tree with $n+1$ leaves

Under the following recursive rules, the word written at the root corresponds to the tree.

Word to tree

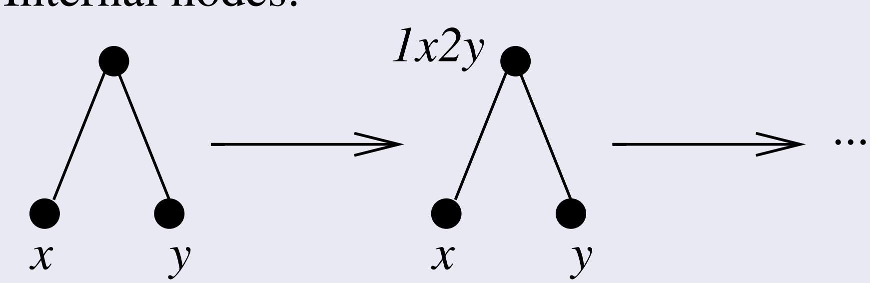
$B(\emptyset) =$ ●



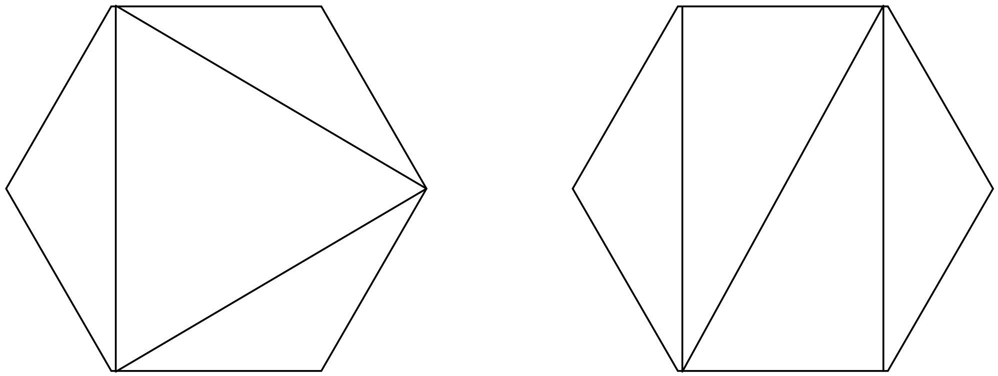
Tree to word

Leaf:

Internal nodes:

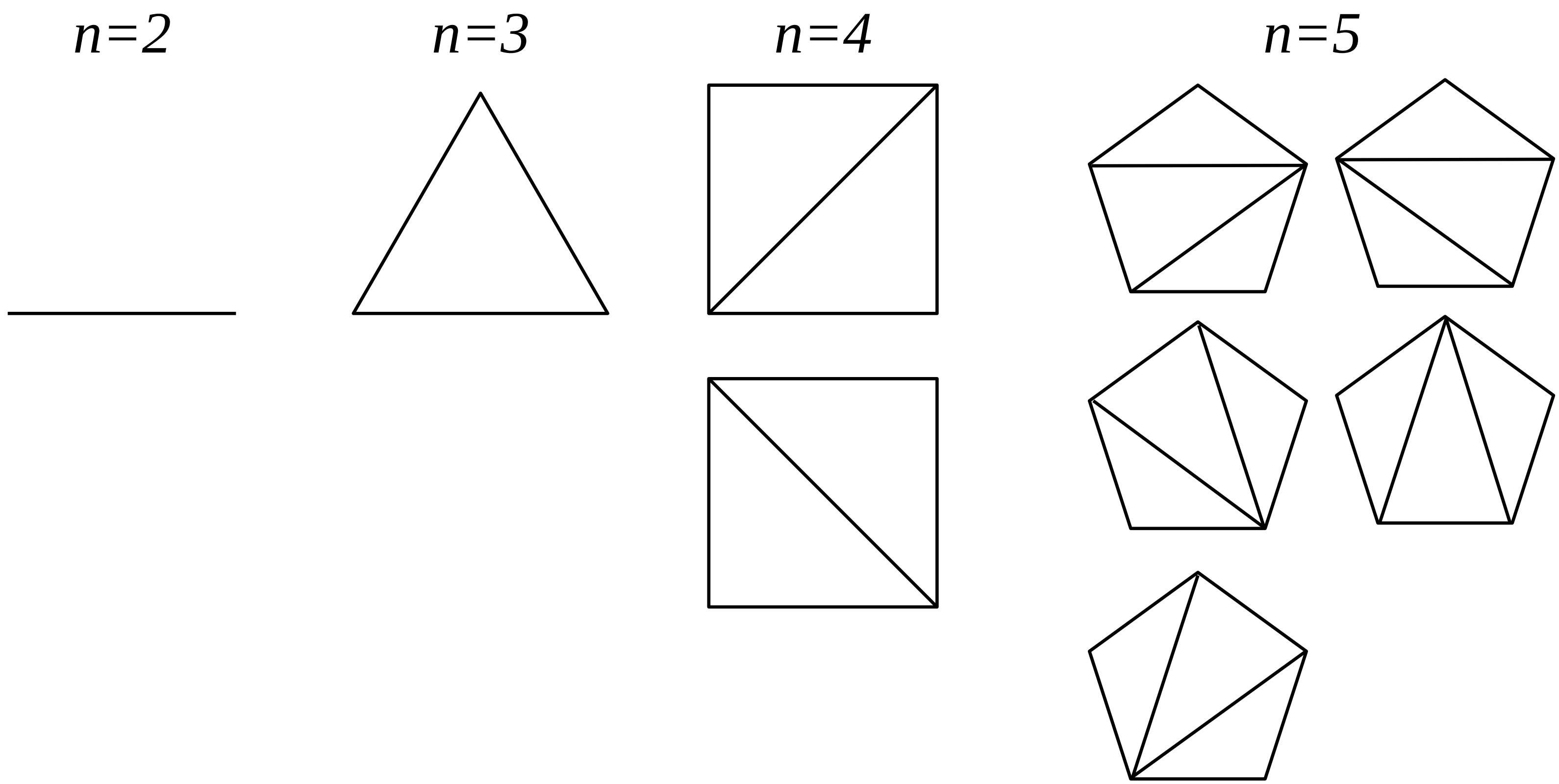


Triangulating regular polygons



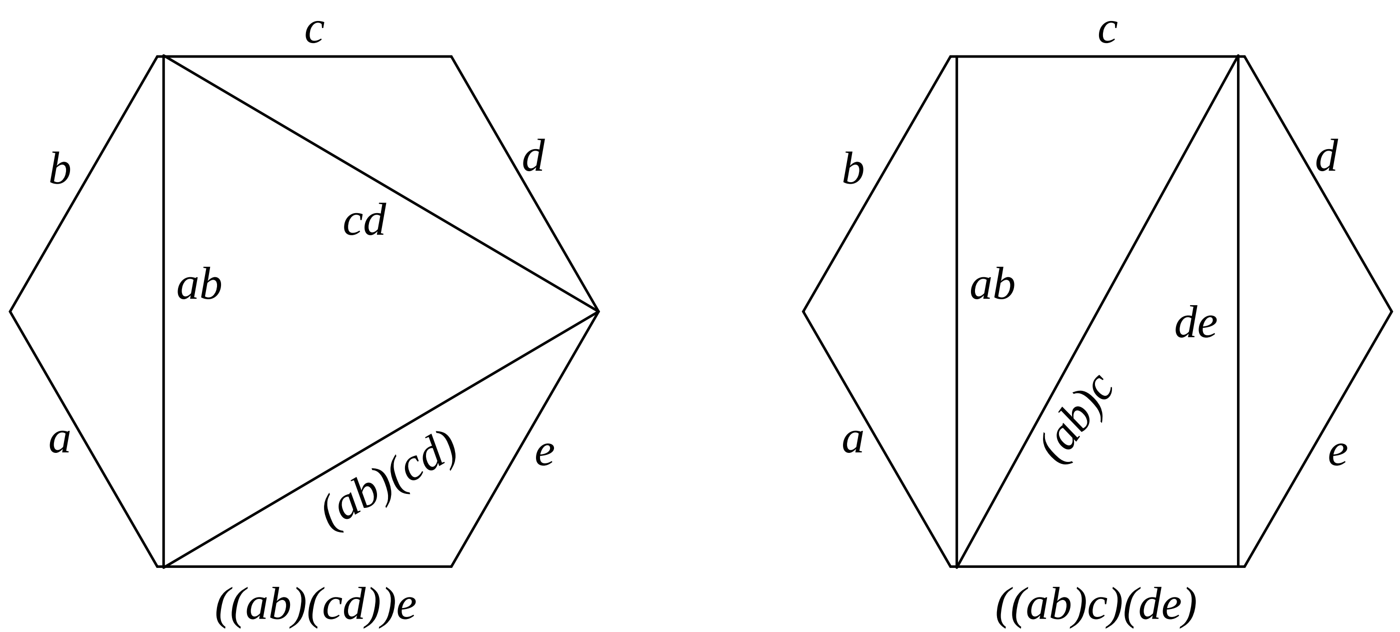
- Draw a regular n-gon with a horizontal base on the bottom.
- A *triangulation* is to draw non-crossing diagonals connecting its vertices, until the whole shape is partitioned into triangles.
- In total there are $n-3$ diagonals.

Triangulating regular polygons



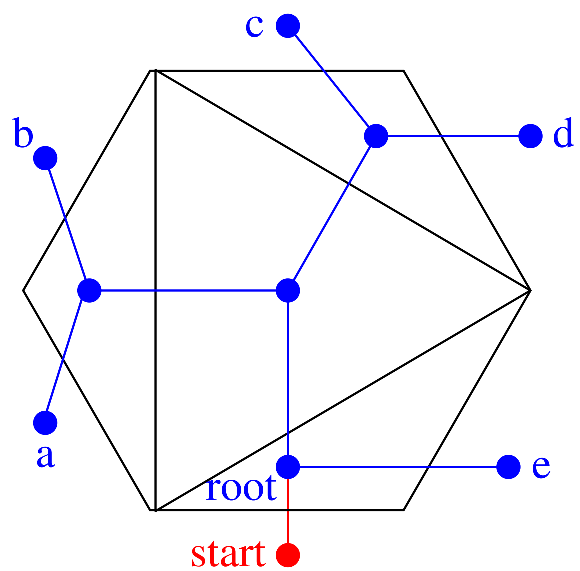
The number of triangulations of an n-gon is C_{n-2}.

Bijection: Triangulating regular polygons $\leftrightarrow$ Binary parenthesizations



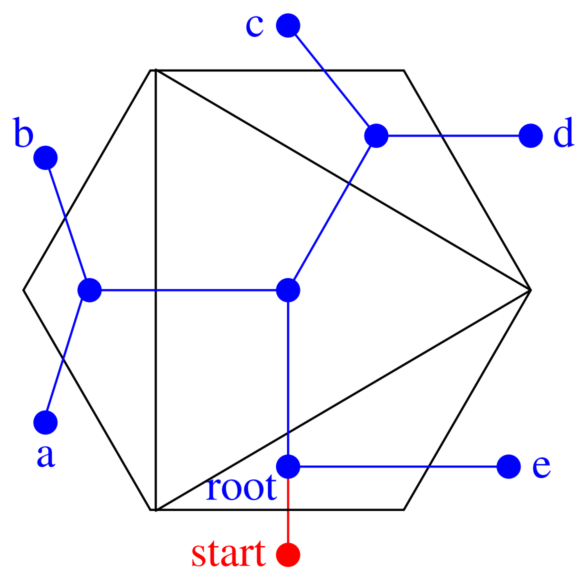
- Draw any triangulation of an n-gon.
- Leave the base empty, and label the other sides by the $n-1$ factors $a, b, c, \ldots$ in clockwise order.
- When two sides of a triangle are labeled, label the third side by their product, using parentheses as needed to track the order of multiplications.
- The base is labelled by a complete binary parenthesization on $n-1$ factors. This procedure is reversible.
- The number of triangulations is $C_{(n-1)-1} = C_{n-2}$.

Bijection: Triangulating regular polygons ↔ Ordered full binary trees



- Draw any triangulation of an n-gon (black).
- Form a tree (blue) as follows:
  - **Vertices:** Place a vertex in each triangle, and a vertex outside each side of the n-gon.
  - **Edges:** connect vertices across edges of the triangles.
  - Remove the start vertex/edge below the base.
    The root is just above it.

Bijection: Triangulating regular polygons $\leftrightarrow$ Ordered full binary trees



- Trees are oriented differently than before, but it's equivalent:
  - The root is at the bottom instead of the top.
  - Leaves go clockwise from the bottom left, rather than left to right. We illustrate this with the labels $a, b, c, \ldots$ for complete parenthesizations.
- The number of triangulations is $C_{(n-1)-1} = C_{n-2}$.

Generating function for Catalan numbers

- Recall that $C_0 = 1$ and
$$C_n = \sum_{i=0}^{n-1} C_i \, C_{n-1-i} \qquad (n > 0)$$
- Let $f(t) = \sum_{n=0}^{\infty} C_n t^n$. Then
$$(f(t))^2 = \sum_{n=0}^{\infty} \left(\sum_{i=0}^{n} C_i \, C_{n-i} \right) t^n$$
$$t(f(t))^2 = \sum_{n=1}^{\infty} \left(\sum_{i=0}^{n-1} C_i \, C_{n-1-i} \right) t^n = \sum_{n=1}^{\infty} C_n \, t^n = f(t) - 1$$
$$t(f(t))^2 - f(t) + 1 = 0 \quad \text{so} \quad f(t) = \frac{1 \pm \sqrt{1-4t}}{2t} = \frac{1 \pm (1 - 2t + \cdots)}{2t}$$
- Since the series of $f(t)$ starts at $1t^0$, the solution is $f(t) = \frac{1-\sqrt{1-4t}}{2t}$.

Generating function for Catalan numbers

- On homework, you computed the binomial series for $\sqrt{1+x}$:

$$(1+x)^{1/2} = \cdots = 1 - 2\sum_{n=1}^{\infty} \frac{(-1/4)^n (2n-2)!}{n!(n-1)!} x^n$$

- Set $x = -4t$:

$$\sqrt{1-4t} = 1 - 2\sum_{n=1}^{\infty} \frac{(-1/4)^n (2n-2)!}{n!(n-1)!} (-4t)^n$$

$$= 1 - 2\sum_{n=1}^{\infty} \frac{(2n-2)!}{n!(n-1)!} t^n$$

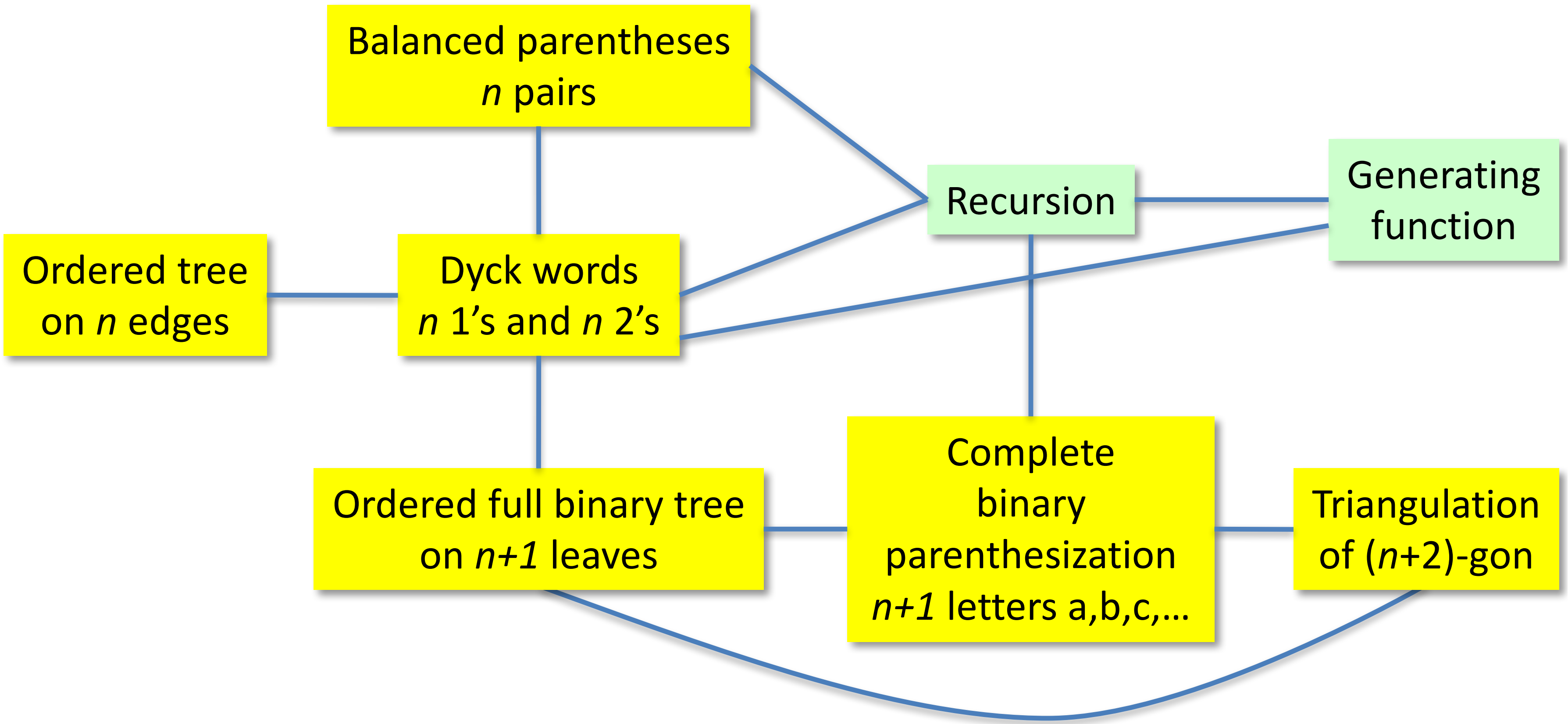
- Thus, $f(t) = \dfrac{1-\sqrt{1-4t}}{2t} = \displaystyle\sum_{n=1}^{\infty} \frac{(2n-2)!}{n!(n-1)!} t^{n-1} = \sum_{n=0}^{\infty} \frac{(2n)!}{(n+1)!n!} t^n$
- Since $f(t) = \sum_{n=0}^{\infty} C_n t^n$, we proved $C_n = \frac{(2n)!}{(n+1)!n!} = \frac{1}{n+1}\binom{2n}{n}$.

Generating function for Catalan numbers

Second derivation of generating function

- The *weight* of a word $u \in W_n$ is $w(u) = n$.
- For $n > 0$, decompose $u = 1x2y$ where x, y are Dyck words. Then
$$w(u) = w(12) + w(x) + w(y) = 1 + w(x) + w(y)$$
- The product formula gives that the generating function for the weight of all Dyck words with $n \geqslant 1$ is $t \cdot f(t) \cdot f(t)$:
$$\sum_{n=1}^{\infty} C_n t^n = \sum_x \sum_y t^{w(1x2y)} = \sum_x \sum_y t^{1+w(x)+w(y)}$$
$$= t\left(\sum_x t^{w(x)}\right)\left(\sum_y t^{w(y)}\right) = t \cdot f(t) \cdot f(t).$$
- This generating function is also
$$\sum_{n=1}^{\infty} C_n t^n = f(t) - 1.$$
- Thus, $t(f(t))^2 = f(t) - 1$, giving the same equation as before.

Summary



- C_n counts the structures above in yellow.
- Edges: We showed bijections between some of them, as well as correspondences with recursions and generating functions.
- Hundreds of types of objects are known to be counted by Catalan numbers.