Linear Algebra
ETH Zürich, HS 2023, 401-0131-00L

The Computer Science Lens

The Generalized Associative Law

Bernd Gärtner

September 29, 2023

Generalized Associativity

"Normal" associativity: For matrices A, B, C (of dimensions such that the multiplications are defined),

$$A(BC) = (AB)C.$$

Knowing this, we can simply talk about the product of 3 matrices and write it as

$$ABC.$$

It doesn't matter whether we compute this as $A(BC)$, or as $(AB)C$.

What about the product of more matrices? Can we for example simply write

$$ABCDEFGH?$$

Yes, because every way of computing this gives the same result. For example

$$\underbrace{((((((AB)C)D)E)F)G)H}_{\text{left to right}} = \underbrace{A(B(C(D(E(F(GH))))))}_{\text{right to left}} = \underbrace{((A(BC))((DE)F))(GH)}_{\text{freestyle}}.$$

Why is this true? There are many proofs [War01]. Here: a computer science proof.

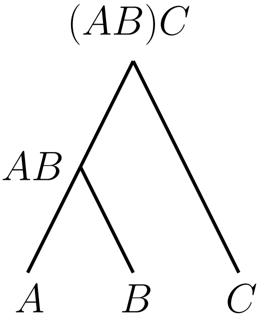
Brackets and Trees

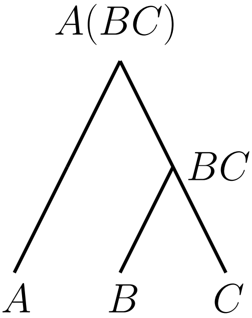
Using brackets, we say how to compute a product of more than two matrices.

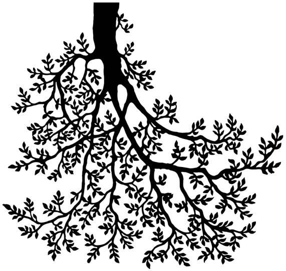
- $(AB)C$: multiply A and B, then multiply the result with C!
- $A(BC)$: multiply B and C, then multiply A with the result!

A different view: *computation trees*.

In computer science, trees grow downwards.

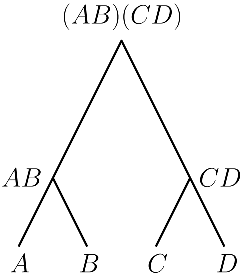


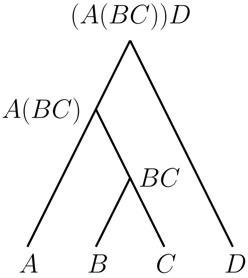


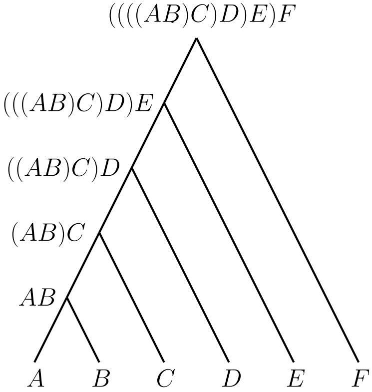


Computation Trees

Tree nodes:	partial results
Leaves (bottom tree nodes):	inputs
Root (top tree node):	final result

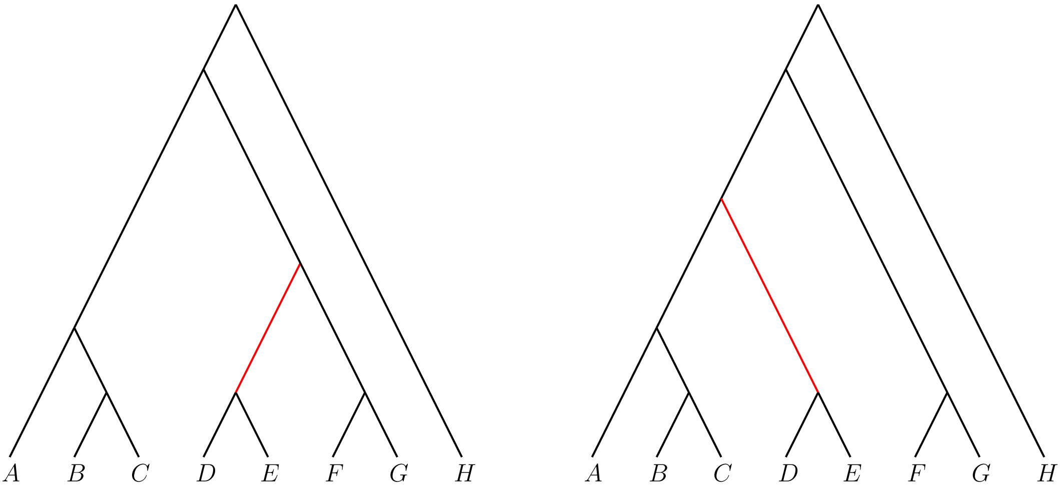






Computation Tree: Rotation

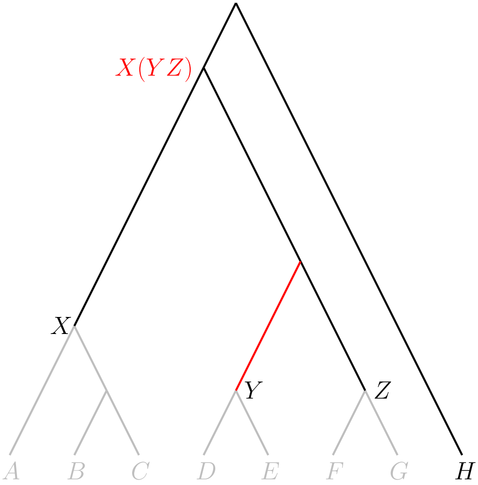
Send a partial result, for example DE, left instead of right, or vice versa!

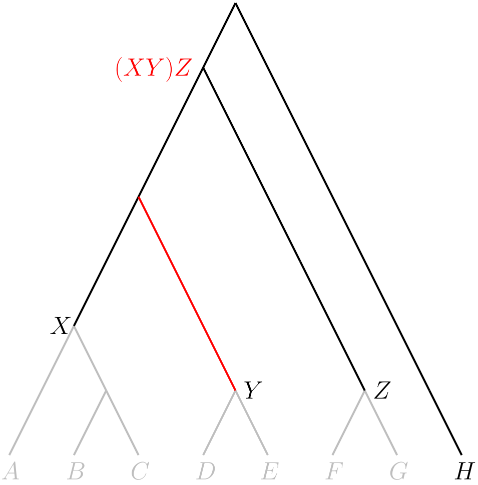


A partial result is affected... but unchanged, by normal associativity!

$X = A(BC), \quad Y = DE, \quad Z = FG$

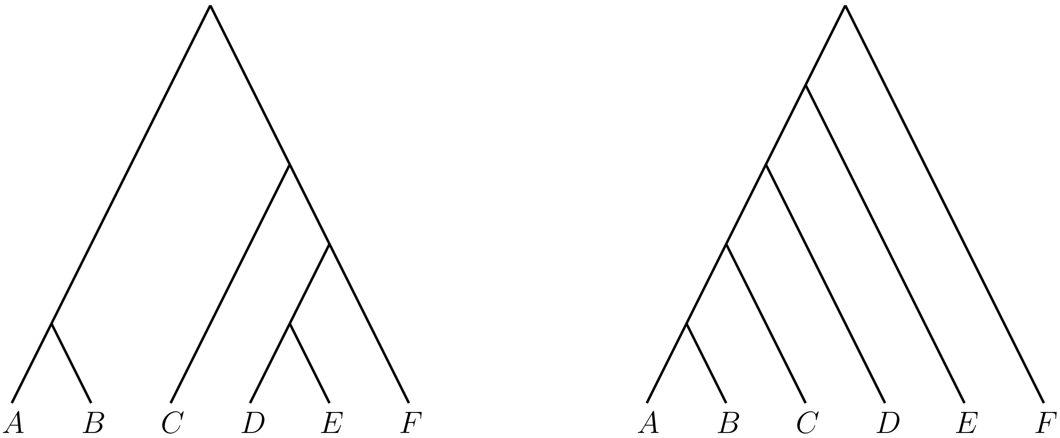
$X(YZ) = (XY)Z$





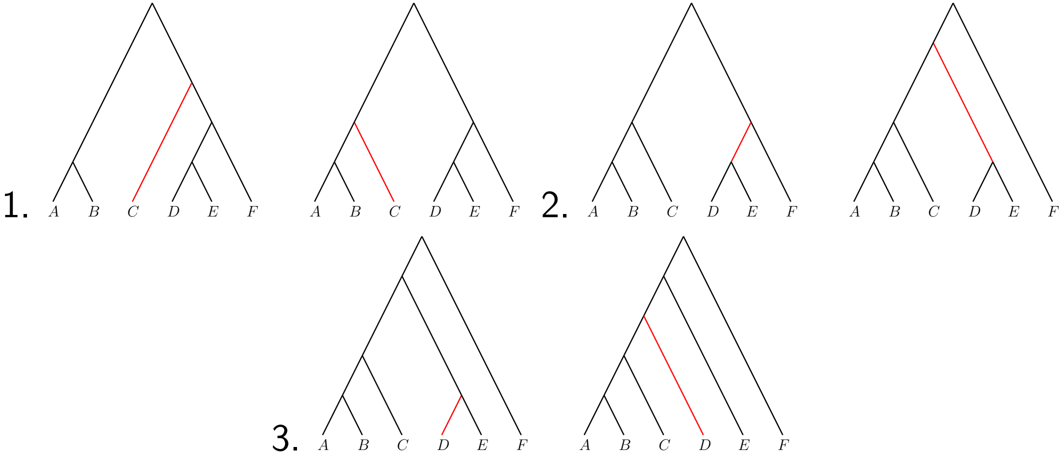
All computation trees give the same final result!

Proof: Step by step, we can rotate any tree into the "left-to-right" tree, without ever changing the result. Hence, all trees give the same result as the "left-to-right" tree.



Step-by-step rotation

As long as a partial result can be sent to the left instead of the right, do it!



Nothing more can be sent to the left, we have the “left to right” tree!

Why was this a "computer science proof"?

- It applies an algorithm.
- If you like, think about how many steps it needs...
- Trees and rotations are fundamental in *Algorithmen und Datenstrukturen*.

Disclaimer: The proof is not mathematically formal.

- It nicely shows what is going on, but is incomplete:
- We argued for 6 matrices, but we may have any number of matrices.
- Does a rotation always work as nicely as in the pictures?
- Is "sending left instead of right" guaranteed to stop eventually?
- If it stops, why do we always get the "left-to-right" tree?

There are complete proofs [War01], so it makes sense to focus on the intuition here.

Application: Fast Fibonacci numbers

Fibonacci numbers at Zürich HB

Fibonacci numbers

The sequence

			2	+ 3	= 5		13	+ 21	= 34		
0	1	1	2	3	5	8	13	21	34	55	...
f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9,	f_{10}	...
			f_3	$+ f_4$	$= f_5$		f_7	$+ f_8$	$= f_9$		

Every number is the sum of the previous two.

Mathematical definition:

$$
\begin{aligned}
f_0 &= 0 \\
f_1 &= 1 \\
f_n &= f_{n-1} + f_{n-2}, \quad \text{if } n \geq 2
\end{aligned}
$$

The Fibonacci number f_{100}...

...is something you always wanted to know!

Here you go (Python):

```
a = 0 # f 0
b = 1 # f 1
for n in range (2, 101):
    c = a + b # f n
    a = b     # f n-2 -> f n-1
    b = c     # f n-1 -> f n
    print ("f", n, "=", c)
```

	f 21 = 10946	f 41 = 165580141	f 61 = 2504730781961
f 2 = 1	f 22 = 17711	f 42 = 267914296	f 62 = 4052739537881
f 3 = 2	f 23 = 28657	f 43 = 433494437	f 63 = 6557470319842
f 4 = 3	f 24 = 46368	f 44 = 701408733	f 64 = 10610209857723
f 5 = 5	f 25 = 75025	f 45 = 1134903170	f 65 = 17167680177565
f 6 = 8	f 26 = 121393	f 46 = 1836311903	f 66 = 27777890035288
f 7 = 13	f 27 = 196418	f 47 = 2971215073	f 67 = 44945570212853
f 8 = 21	f 28 = 317811	f 48 = 4807526976	f 68 = 72723460248141
f 9 = 34	f 29 = 514229	f 49 = 7778742049	f 69 = 117669030460994
f 10 = 55	f 30 = 832040	f 50 = 12586269025	f 70 = 190392490709135
f 11 = 89	f 31 = 1346269	f 51 = 20365011074	f 71 = 308061521170129
f 12 = 144	f 32 = 2178309	f 52 = 32951280099	f 72 = 498454011879264
f 13 = 233	f 33 = 3524578	f 53 = 53316291173	f 73 = 806515533049393
f 14 = 377	f 34 = 5702887	f 54 = 86267571272	f 74 = 1304969544928657
f 15 = 610	f 35 = 9227465	f 55 = 139583862445	f 75 = 2111485077978050
f 16 = 987	f 36 = 14930352	f 56 = 225851433717	f 76 = 3416454622906707
f 17 = 1597	f 37 = 24157817	f 57 = 365435296162	f 77 = 5527939700884757
f 18 = 2584	f 38 = 39088169	f 58 = 591286729879	f 78 = 8944394323791464
f 19 = 4181	f 39 = 63245986	f 59 = 956722026041	f 79 = 14472334024676221
f 20 = 6765	f 40 = 102334155	f 60 = 1548008755920	f 80 = 23416728348467685

```
f 81 = 37889062373143906
f 82 = 61305790721611591
f 83 = 99194853094755497
f 84 = 160500643816367088
f 85 = 259695496911122585
f 86 = 420196140727489673
f 87 = 679891637638612258
f 88 = 1100087778366101931
f 89 = 1779979416004714189
f 90 = 2880067194370816120
f 91 = 4660046610375530309
f 92 = 7540113804746346429
f 93 = 12200160415121876738
f 94 = 19740274219868223167
f 95 = 31940434634990099905
f 96 = 51680708854858323072
f 97 = 83621143489848422977
f 98 = 135301852344706746049
f 99 = 218922995834555169026
f 100 = 354224848179261915075
```

$$f_{100} = 354224848179261915075.$$

Let's do it faster...

Observation [Mat10]:

$$\begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}}_{A} \begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix}, \quad \text{if } n \geq 2$$

$$\begin{aligned} \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix} &= A \begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix} = A\left(A \begin{bmatrix} f_{n-3} \\ f_{n-2} \end{bmatrix}\right) = \cdots = A\left(\cdots\left(A \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}\right)\cdots\right) \\ &= \underbrace{A\left(\cdots\left(A \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)\cdots\right)}_{n-1 \ A\text{'s}} = A^{n-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{(generalized associativity!)} \end{aligned}$$

To compute f_n, we need the matrix $A^{n-1} = \underbrace{A \cdot A \cdots A}_{n-1 \text{ times}}$.

Then f_n is found in the lower right corner.

Fast powers by iterative squaring

$$A^{99} = \left(A^{49}\right)^2 \cdot A = \begin{bmatrix} 135301852344706746049 & 218922995834555169026 \\ 218922995834555169026 & \color{red}{354224848179261915075} \end{bmatrix}$$

$$\color{red}{= f_{100}}$$

$$\swarrow$$

$$A^{49} = \left(A^{24}\right)^2 \cdot A = \begin{bmatrix} 4807526976 & 7778742049 \\ 7778742049 & 12586269025 \end{bmatrix}$$

$$\swarrow$$

$$A^{24} = \left(A^{12}\right)^2 = \begin{bmatrix} 28657 & 46368 \\ 46368 & 75025 \end{bmatrix}$$

$$\swarrow$$

$$A^{12} = \left(A^{6}\right)^2 = \begin{bmatrix} 89 & 144 \\ 144 & 233 \end{bmatrix}$$

$$\swarrow$$

$$A^{6} = \left(A^{3}\right)^2 = \begin{bmatrix} 5 & 8 \\ 8 & 13 \end{bmatrix}$$

$$\swarrow$$

$$A^{3} = (A)^2 \cdot A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$

References

Jiří Matoušek.
Thirty-three Miniatures - Mathematical and Algorithmic Applications of Linear Algebra.
American Mathematical Society, 2010.
https://kam.mff.cuni.cz/~matousek/stml-53-matousek-1.pdf.

William P. Wardlaw.
A generalized general associative law.
Mathematics Magazine, 74(3):230–233, 2001.
https://doi.org/10.1080/0025570X.2001.11953069.