Foundation of Computer Science 1

©Laura Kovács

Revision 3
December 9, 2009

Problem 1.

(1.1) Let $a, b, c, d, e, f, x, y, z, w \in \mathbb{N}$. For each of the expressions

(i) $(x + y) * (x - 3)$

(ii) $((x - y) * z + (y - w)) * x$

(ii) $\Big(\big(((a * x + b) * x + c) * x + d \big) * x + e \Big) * x + f$

do the following:

(a) Construct the syntax tree;

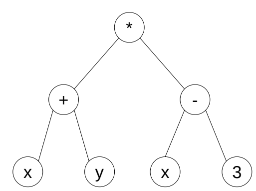
(b) Find the equivalent prefix notation;

(c) Find the equivalent postfix notation.

Solution.

(i) Expression $(x + y) * (x - 3)$**.**

(a) Syntax Tree:

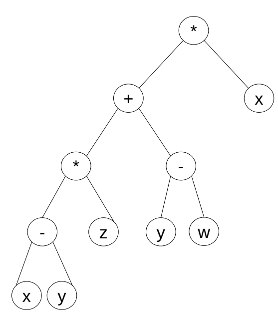


(b) Prefix notation: $* + xy - x3$

(c) Postfix notation: $xy + x3 - *$

(ii) Expression $((x - y) * z + (y - w)) * x$**.**

(a) Syntax Tree:

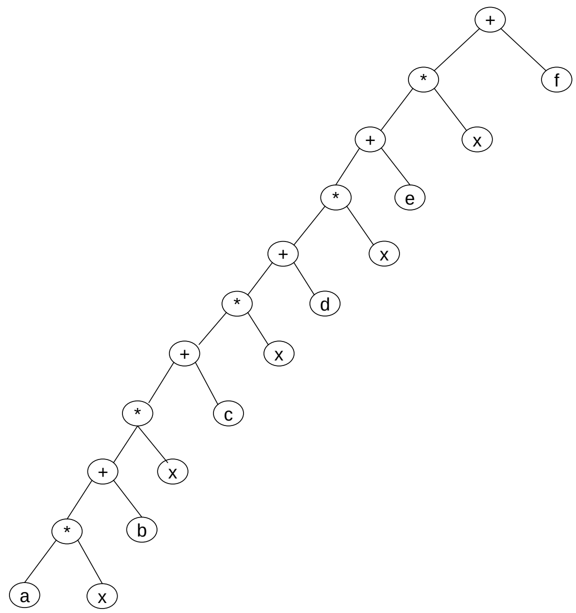


(b) Prefix notation: $* + * - xyz - ywx$

(c) Postfix notation: $xy - z * yw - +x*$

(iii) Expression $\Big(\big(((a * x + b) * x + c) * x + d \big) * x + e \Big) * x + f$

(a) Syntax Tree:



(b) Prefix notation: $+ \ * \ + \ * \ + \ * \ + \ * \ + \ * \ a \ x \ b \ x \ c \ x \ d \ x \ e \ x \ f$

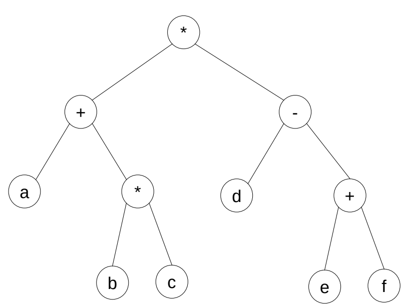
(c) Postfix notation: $a \ x \ \ * \ b \ + \ x \ * \ c \ + \ x \ * \ d \ + \ x \ * \ e \ + \ x \ * \ f \ +$

(1.2) Let $a, b, c, d, e, f \in \mathbb{N}$. Convert the expression $abc * +def + -*$ from postfix to

(a) infix;

(b) prefix.

Solution. We first construct the syntax tree of the arithmetic expression $abc{*}{+}def{+}{-}{*}$ given in postfix. The syntax tree is:



(a) By the infix traversal of the above syntax tree, the infix notation of the arithmetic expression is:

$$a \ + \ b \ * \ c \ * \ d \ - \ e \ + \ f$$

(b) By the prefix traversal of the above syntax tree, the prefix notation of the arithmetic expression is:

$$* \ + \ a \ * \ b \ c \ - \ d \ + \ e \ f$$

Problem 2.

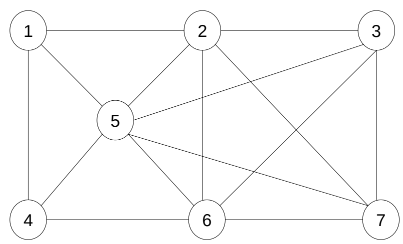
(2.1) Consider the graph given by the adjacency matrix:

	1	2	3	4	5	6	7
1	0	1	0	1	1	0	0
2	1	0	1	0	1	1	1
3	0	1	0	0	1	1	1
4	1	0	0	0	1	1	0
5	1	1	1	1	0	1	1
6	0	1	1	1	1	0	1
7	0	1	1	0	1	1	0

(a) What is the degree of node 7?

(b) Find a 5-clique in the graph, and list the set of nodes defining this 5-clique!

Solution. We first construct the graph represented by the above given adjacency matrix. The graph is:



(a) As there are 4 edges to which node 7 is incident to, the degree of node 7 is 4.

(b) A 5-clique (that is a complete subgraph with 5 nodes) in the graph is formed by the nodes:

$$\{2,\ 3,\ 5,\ 6,\ 7\}.$$

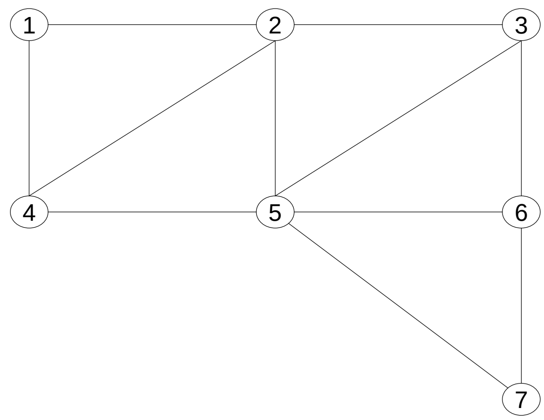
(2.2) Hamilton and Euler went to holiday. They visited a country with 7 cities (nodes) connected by a system of roads (edges) described by the graph given in the following adjacency matrix:

	1	2	3	4	5	6	7
1	0	1	0	1	0	0	0
2	1	0	1	1	1	0	0
3	0	1	0	0	1	1	0
4	1	1	0	0	1	0	0
5	0	1	1	1	0	1	1
6	0	0	1	0	1	0	1
7	0	0	0	0	1	1	0

(a) Could Hamilton visit each city once and return to his starting city? If yes, list the path defining the corresponding hamiltonian cycle!

(b) Could Euler visit each road once? If yes, give the path defining the corresponding eulerian path!

Solution. We first construct the graph represented by the above given adjacency matrix. The graph is:



(a) Hamilton could follow, for example, the hamiltonian cycle:

$$(1,\ 2,\ 3,\ 6,\ 7,\ 5,\ 4,\ 1)$$

(b) Euler could not find an eulerian path in the graph!

The reason is as follows.

Suppose that Euler wants to start at city a and finish at city b.
(a and b denote nodes of the graph).

- Let c be a city different than a and b. Whenever Euler arrives at c, he needes to leave c via a road that he has not taken before. So, c has to be incident to an even number of roads, that is c has to have an even degree. Hence, any city c different than a and b has to have an *even degree.*

- Further, if a is the same as b (that is Euler wants to make an eulerian cycle), then a (and b) has to have *even degree.*

- Finally, if a and b are not the same (that is Euler wants to make an eulerian path that is not a cycle), then a and b have to have *odd degrees.*

Hence, as summary, Euler could visit each road in the graph exactly once if there are *at most two cities with odd degrees.* However, the cities 1, 2, 3, 4, 5, 6, 7 have respectively degrees 2, 4, 3, 3, 5, 3, 2. Hence, there are *four cities* (nodes 3, 4, 5 and 6) *with odd degrees.* Therefore, Euler cannot make an eulerian path in the graph!

Problem 3. Estimate the upper bounds of the following functions in $n \in \mathbb{N}$. Your estimations should be as tight as possible!
Justify your answer!

(3.1) $n + \log\, n$;

Solution.

$$\begin{aligned} n + \log\, n \quad \in \quad & O(n + \log\, n) \\ & \overset{PlusRule}{=} \\ & O(n) + O(\log\, n) \\ & \overset{O(\log\, n) \subset O(n)}{=} \\ & O(n) \end{aligned}$$

Hence,

$$n + \log\, n \in O(n).$$

(3.2) $(2 * n^2) * 2^n$;

Solution.

$$\begin{aligned} 2 * n^2 * 2^n \quad \in \quad & O(2 * n^2 * 2^n) \\ & \overset{ConstantRule}{=} \\ & O(n^2 * 2^n) \\ & \overset{MultiplicationRule}{=} \\ & O(n^2) * O(2^n) \end{aligned}$$

Since, $n^2 \leq 2^n$, we have $n^2 * 2^n \leq 4^n$. That is $O(n^2 * 2^n) \subseteq O(4^n)$, and hence:

$$2 * n^2 * 2^n \quad \in \quad O(4^n)$$

Moreover, we claim that $O(4^n)$ is the tightest upper bound for $2 * n^2 * 2^n$. To justify this claim, we need to prove that $O(4^n) = O(n^2 * 2^n)$, that is we need to prove that

$$O(n^2 * 2^n) \subseteq O(4^n) \qquad \text{and} \qquad O(4^n) \subseteq O(n^2 * 2^n).$$

We have already proved that $O(n^2 * 2^n) \subseteq O(4^n)$. We are thus left with proving

$$O(4^n) \subseteq O(n^2 * 2^n), \qquad \text{that is} \qquad 4^n \in O(n^2 * 2^n).$$

This is equivalent with proving that $\log\, 4^n \in O(\log\, (n^2 * 2^n))$. To this end, we have:

$$\log\, 4^n = n * \log\, 4 \quad \in \quad O(n * \log\, 4) \quad \overset{ConstantRule}{=} \quad O(n),$$

that is $\log\, 4^n \in O(n)$.
Further,

$$\begin{aligned} O(\log\, (n^2 * 2^n)) \quad & = \quad O(\log\, n^2 \; + \; \log\, 2^n) \\ & \overset{AdditionRule}{=} \quad O(2 * \log\, n) \; + \; O(n * \log\, 2) \\ & \overset{ConstantRule}{=} \quad O(\log\, n) \; + \; O(n) \\ & \overset{O(\log\, n) \subset O(n)}{=} \quad O(n) \end{aligned}$$

Hence, $\log 4^n \in O(n) = O(\log\left(n^2 * 2^n\right))$, concluding that $4^n \in O(n^2 * 2^n)$. We thus have $O(4^n) = O(n^2 * 2^n)$, yielding finally that

$$2 * n^2 * 2^n \in O(4^n) \text{ is the tightest upper bound.}$$

Let us next show that $O(c^n) = O(e^n)$ for any constant $c \geq 2$. This is equivalent to showing that $O(\log c^n) = O(\log e^n)$, as it is given below.

$$\begin{aligned} O(n * \log c) &= O(n * \log c) \\ &\overset{ConstantRule}{=} O(n) \\ &= O(\log e^n) \end{aligned}$$

Thus, $O(c^n) = O(e^n)$ for any constant $c \geq 2$.

In particular, we hence have $O(4^n) = O(e^n)$, and therefore we conclude:

$$2 * n^2 * 2^n \in O(e^n).$$

As a consequence of the above reasoning, note the following two rules:

$$\boxed{O(c^n) = O(e^n) \quad \text{for any constant } c \geq 2}$$

$$\boxed{O(c^n) = O(c^{k*n}) \quad \text{for any constants } c, k \geq 1}$$

That is, for example, $O(2^n) = O(2^n * 2^n)$ and $O(e^n) = O(4^n)$.

(3.3) $\log(3 * n^2)$ with $n > 0$;

Solution.

$$\begin{aligned} \log(3 * n^2) \in\ & O(\log(3 * n^2)) \\ =\ & O(\log 3 \ + \ \log n^2)) \\ \overset{AdditionRule}{=}\ & O(\log 3) + O(2 * \log n) \\ \overset{ConstantRule}{=}\ & O(1) + O(\log n) \\ \overset{O(1) \subset O(\log n)}{=}\ & O(\log n) \end{aligned}$$

Hence,

$$\log(3 * n^2) \in O(\log n).$$

(3.4) $\text{ld}(3 * n^2 - 1) \quad$ with $n > 0$;

Solution. Since $3 * n^2 - 1 \leq 3 * n^2$ for every $n \in \mathbb{N}$, we have $\text{ld}(3 * n^2 - 1) \leq \text{ld}(3 * n^2)$. Hence,

$$\begin{aligned}
\text{ld}(3 * n^2 - 1) \in & \; O(\text{ld}(3 * n^2)) \\
= & \; O(\text{ld}(3) + \text{ld}(n^2)) \\
\overset{Addition Rule}{=} & \; O(\text{ld}(3)) + O(2 * \text{ld}(n)) \\
\overset{Constant Rule}{=} & \; O(1) + O(\text{ld}(n)) \\
\overset{O(1) \subset O(\text{ld}(n))}{=} & \; O(\text{ld}(n)) \\
= & \; O(\frac{1}{\log 2} * \log n) \\
\overset{Constant Rule}{=} & \; O(\log n)
\end{aligned}$$

Hence,

$$\text{ld}(3 * n^2 - 1) \in O(\log n).$$

Note that this is the tighest upper bound.

(3.5) $\frac{n*(n+1)}{2} + 3 * n$;

Solution.

$$\begin{aligned}
\frac{n*(n+1)}{2} + 3*n \quad \in \quad & O(\frac{n*(n+1)}{2} + 3*n) \\
\overset{AdditionRule}{=} \quad & O(\frac{n*(n+1)}{2}) \ + \ O(3*n) \\
\overset{ConstantRule}{=} \quad & O(n*(n+1)) \ + \ O(n) \\
\overset{AdditionRule}{=} \quad & O(n^2 + 3*n) \\
\overset{AdditionRule}{=} \quad & O(n^2) + O(3*n) \\
\overset{ConstantRule}{=} \quad & O(n^2) + O(n) \\
\overset{O(n)\subseteq O(n^2)}{=} \quad & O(n^2)
\end{aligned}$$

Hence,

$$\frac{n*(n+1)}{2} + 3*n \in O(n^2).$$

(3.6) $O(f)^3 + O(g) * O(h)$, where f, g, h are functions in $n \in \mathbb{N}$;

Solution.

$$\begin{aligned}
O(f)^3 + O(g) * O(h) \quad & \overset{MultiplicationRule}{=} \quad O(f^3) + O(g*h) \\
& \overset{AdditionRule}{=} \quad O(f^3 + g*h)
\end{aligned}$$

Hence,

$$O(f)^3 + O(g) * O(h) = O(f^3 + g*h).$$

(3.7) $2 * O(f) + O(g)$, where f, g are functions in $n \in \mathbb{N}$.

Solution.

$$\begin{aligned}
2 * O(f) + O(g) \quad & \overset{AdditionRule}{=} \quad O(2*f) + O(g) \\
& \overset{ConstantRule}{=} \quad O(f) + O(g) \\
& \overset{AdditionRule}{=} \quad O(f+g)
\end{aligned}$$

Hence,

$$2 * O(f) + O(g) = O(f+g).$$