IT-math F2003 : Selected Solution(s)
Episode 11, April 22, 2003

FP1(d). Select a Theta notation from among $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^3 / \log n)$ for $\frac{(n^2+\log n)(n+1)}{n+n^2}$, and motivate your answer.

Solution. We show that $\frac{(n^2+\log n)(n+1)}{n+n^2} = \Theta(n)$: One has

$$\frac{(n^2+\log n)(n+1)}{n+n^2} = \frac{(n^2+\log n)(n+1)}{n(n+1)} = n + \frac{\log n}{n}$$

for $n \neq 0$. Observe that $\left|\frac{\log n}{n}\right| \leq \log n$ for $n \geq 1$, so $\frac{\log n}{n} = O(\log n)$. Recalling that $\log n = o(n)$ by Proposition 13(a) [from Supplementary Material to Episodes 10–11] we conclude that $\frac{\log n}{n} = o(n)$. Since $n = \Theta(n)$ and $\frac{\log n}{n} = o(n)$, by Lemma 9(b) one has $\frac{(n^2+\log n)(n+1)}{n+n^2} = n + \frac{\log n}{n} = \Theta(n)$.

FP2(Ω). Suppose $f = \Omega(g)$ and let h be any function. Prove that $f \cdot h = \Omega(g \cdot h)$.

Solution. Since $f = \Omega(g)$, there exists a real $C > 0$ such that for almost all n the inequality $C \cdot |f(n)| \geq |g(n)|$ holds. Multiplying both sides by the non-negative $|h(n)|$ we obtain

$$C \cdot |f(n) \cdot h(n)| = C \cdot |f(n) \cdot |h(n)| \geq |g(n)| \cdot |h(n)| = C \cdot |g(n) \cdot h(n)|$$

for almost all n, which tells us that $f \cdot h = \Omega(g \cdot h)$.

SC1(a). Prove or refute: $\log_2 n = o(3^n)$.

Solution. We prove this as follows: First, observe that $|\log 2n| = |\log 2 + \log n| \leq \log 2 + |\log n| \leq 2 \cdot |\log n|$ for almost all n. As $\log n = o(n)$ by Proposition 13(a), we conclude $\log 2n = o(n)$. Further, $n = o(3^n)$ by Proposition 12(a), hence $n = O(3^n)$. Recalling Lemma 10(a), we conclude $\log_2 n = o(3^n)$.

SC1(c). Prove or refute: $3^n - n - 1000(\log n)^2 = \Omega(2^n)$.

Solution. We prove this as follows: $2^n = o(3^n)$ by Proposition 14. Hence $3^n = \Omega(2^n)$. Further, $-n = o(2^n)$ by Proposition 12(a), and $1000(\log n)^2 = o(n) = o(2^n)$ by Propositions 13(b) and 12(a). Therefore by Lemma 9(b) one can conclude $3^n - n - 1000(\log n)^2 = \Omega(2^n)$.

SC2. Suppose $f = o(g)$ and that the function h is such that $h(n) \neq 0$ for almost all n. Prove that $f \cdot h = o(g \cdot h)$.

Solution. Fix an arbitrary $c > 0$. As $f = o(g)$, the inequality $|f(n)| < c \cdot |g(n)|$ holds for almost all n. For almost all n, we have $|h(n)| > 0$. For these n we can multiply the former strict inequality by $|h(n)|$ to obtain

$$|f(n) \cdot h(n)| = |f(n)| \cdot |h(n)| < c \cdot |g(n)| \cdot |h(n)| = c \cdot |g(n) \cdot h(n)|$$

for almost all n, which tells us that $f \cdot h = o(g \cdot h)$.

SC3. For any real $a > 1$ show that $a^n = o(n!)$.

Solution. Fix an arbitrary $c > 0$. If $n > \frac{a^{\lceil a\rceil+1}}{c\cdot\lceil a\rceil!}$ then we have

$$|a^n| = a^n = a^{\lceil a\rceil}\cdot a^{n-\lceil a\rceil} < \frac{n\cdot c\cdot\lceil a\rceil!}{a}\cdot a^{n-\lceil a\rceil} = c\cdot\lceil a\rceil!\cdot a^{n-\lceil a\rceil-1}\cdot n \le c\cdot n!$$

This shows $a^n = o(n!)$.

LH1. Show that $2^n = o\left(\binom{2n}{n}\right)$.

Solution. First let us show that $2^n = o(n^n)$. Let $c > 0$. If $n > \max\left\{\frac{2}{c}, 2\right\}$ then

$$2^n < 2^{n-1}cn \le c\cdot n^{n-1}\cdot n = c\cdot n^n,$$

establishing $2^n = o(n^n)$.

Further, observe that

$$\binom{2n}{n} = \frac{(2n)!}{n!n!} = (n+1)\cdot(n+2)\cdot\dots\cdot 2n \ge n^n$$

for all n, hence $n^n = O\left(\binom{2n}{n}\right)$. By Lemma 10(a), $2^n = o\left(\binom{2n}{n}\right)$ follows.

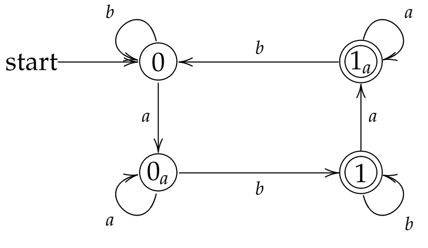
LH2. Prove or refute: $\binom{2n}{n} = O(n^{\log_2 n})$.

Solution. We shall refute this by showing $n^{\log_2 n} = o\left(\binom{2n}{n}\right)$. By Proposition 13(b) we have $(\log_2 n)^2 = o(n)$. Therfeore $(\log_2 n)^2 \le n$ for almost all n. Hence $n^{\log_2 n} = 2^{(\log_2 n)^2} \le 2^n$ for almost all n. So $n^{\log_2 n} = O(2^n)$. Recall that $2^n = o\left(\binom{2n}{n}\right)$ by Exercise LH1. We conclude $n^{\log_2 n} = o\left(\binom{2n}{n}\right)$ by Lemma 10(b).

LH3. Construct an automaton accepting the language

$$\{\, w \in \{a,b\}^* \mid w \text{ has an odd number of occurrences of the subword } ab \,\}.$$

Solution. Here is a deterministic automaton M for the task:



To see that it works, one shows by induction on the length $|w|$ of a word $w \in \{a,b\}^*$ that

(i) M is in state 0 after reading $w \iff$ the number of occurrences of ab in w is even and w does *not* end with a;

(ii) M is in state 0_a after reading $w \iff$ the number of occurrences of ab in w is even and w ends with a;

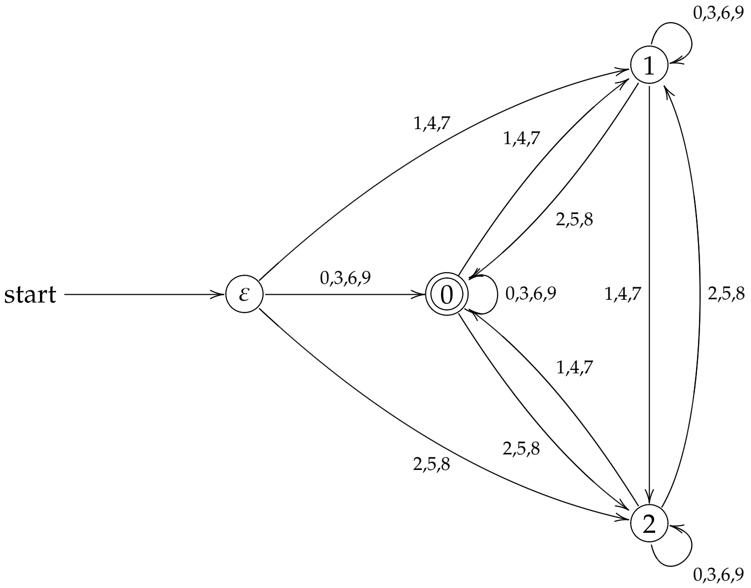
(iii) M is in state 1 after reading $w \iff$ the number of occurrences of ab in w is odd and w does *not* end with a;

(ii) M is in state 1_a after reading $w \iff$ the number of occurrences of ab in w is odd and w ends with a.

DS1. Construct an automaton accepting the language

$$\{\, w \in \{0,\dots,9\}^* \mid w \text{ is a decimal representation of a natural number divisible by } 3 \,\}.$$

Solution. Recall that a number $(n_k \dots n_1 n_0)_{10}$ in decimal notation is divisible by 3 if and only if $\sum_{0\le i\le k} n_i$ is divisible by 3. The following automaton, having read a non-empty sequence of decimal digits $n_k \dots n_0$, finds itself in state i if and only if $\sum_{0\le i\le k} n_i \equiv i \pmod 3$.



[Multiple labels on a single arrow are shorthand for multiple arrows, each labelled by one of the labels.] This automaton does not accept the 'empty number', but allows a decimal number to begin with an arbitrary number of zeroes. Should one wish to disallow that, one should slightly modify the automaton.