CPSC 421/501 HW4, 2025

(0) Joel Friedman

(1) (a) List all ASCII strings:

$i_1, i_2, i_3, \ldots$

(you can do this by length, and by lexicographical order). Given $p \in$ VALID-PYTHON, do as follows:

Run (i.e. simulate with a universal Python program):

Phase 1: One step of p on i_1

Phase 2: 2 steps of p on each of i_1, i_2

Phase 3: 3 steps of p on each of i_1, i_2, i_3

$\vdots$

Phase k: k steps of p on each of $i_1, i_2, \ldots, i_k$

$\vdots$

If p accepts one of $i_1, \ldots, i_k$ after k steps, this algorithm accepts p.

(If p accepts some input, then for $m \in \mathbb{N}$, $i = i_m$, and p accepts i after s steps, then the above algorithm will stop in Phase $\max(m, s)$ and accept p.)

(b) There is an algorithm that takes p and i and produces a program $q = q(p, i)$ that "hardwires i into p" and runs p on input i (which means that q effectively ignores its input).

Hence

$$q \in L \iff p \text{ accepts } i.$$

So if L is decidable, then ACCEPTANCE is decidable, which is a contradiction.

(2) (a) Similarly to 1(a), we work in Phases: given i, phase k is:

Phase k: We run $p_1, \ldots, p_k$ on i for k steps (and accept i if any of $p_1, \ldots, p_k$ accepts i after k steps).

This algorithm accepts i iff $i \in L$ (note: we assume that we have a way s.t. given j, we can produce p_j).

(b) Let $p_1 = p_2 = \ldots = \text{ACCEPTANCE}$.

Then $i \in L \iff i \in \text{ACCEPTANCE}$.

So L cannot be decidable, since

ACCEPTANCE is undecidable.

(3) 9.3.1

(a) The integer $n \in \mathbb{N}$ can be written as $\underbrace{1+1+\cdots+1}_{\text{with } n\ 1\text{'s}}$, and hence

the meaning of

$\underbrace{\text{one plus one plus ... plus one}}_{\text{with } n \text{ "one"s}}$

is n.

(b) We have "one" means $1 \in \mathbb{N}$.

If $k \in \mathbb{N}$, then

$\underbrace{\text{"two times ... times two"}}_{k \text{ "two"s}}$ means 2^k, and is a sentence with $2k-1$ words.

Hence for all $k \in \mathbb{Z}_{\geq 0}$, we can express 2^k in a sentence of

$$\begin{cases} 2k-1 \text{ words}, & \text{if } k \geq 1, \\ 1 \text{ word}, & \text{if } k = 0. \end{cases}$$

Any other $n \in \mathbb{N}$ can be written in binary as $n = 2^{k_1} + 2^{k_2} + \dots + 2^{k_m}$

where $k_1 > k_2 > \dots > k_m \geq 0$ are integers. Writing

s_1 plus s_2 plus ... plus s_m

where

s_i means 2^{k_i}

and is as above, we get

a sentence whose meaning is n, and whose length is

$\text{length}(s_1) + \cdots + \text{length}(s_m) + m-1$

(since we have $m-1$ "pluses"); since

$$\text{length}(s_i) = \begin{cases} 2k_i - 1 & \text{if } k_i \geq 1 \\ 1 = 2k_i + 1 & \text{if } k_i = 0 \end{cases}$$

we have

$$\text{length}(s_1) + \dots + \text{length}(s_{m-1}) + \text{length}(s_m)$$

$$\leq (2k_1 - 1) + \dots + (2k_{m-1} - 1) + (2k_m + 1)$$

(we need $2k_m + 1$ in case $k_m = 0$)

$$\leq 2 + \sum_{i=1}^{m} (2k_i - 1)$$

$$\leq 2 + m(2k_1 - 1) \qquad (*)$$

Since $k_1 > k_2 > \dots > k_m \geq 0$,

we claim that $k_1 \geq m - 1$:

One way to see this is to write

$$k_1 \geq 1 + k_2 \geq 1 + (1 + k_3) = 2 + k_3$$

$$\geq 2 + (1 + k_4) = 3 + k_4$$

$$\geq \cdots \geq m - 1 + k_m \geq m - 1$$

(since $k_m \geq 0$); alternatively you can show for $i = 0, 1, \ldots, m-1$ we have $k_{m-i} \geq i$, using induction on i.

Plugging $k_1 \geq m - 1$ we have

$(*)$ equals

$$2 + m(2k_1 - 1)$$

$$\leq 2 + (k_1 + 1)(2k_1 - 1)$$

$$= 2k_1^2 + k_1 + 1$$

Hence the size of

S_1 plus S_2 plus ... plus S_m

is at most

$$m - 1 + \sum_{i=1}^{m} \mathrm{size}(S_m)$$

$$\leq m - 1 + 2k_1^2 + k_1 + 1$$

$$\leq k + 2k_1^2 + k_1 + 1$$

$$\leq 4k_1^2 + 1 \quad (\text{since } k_1 \in \mathbb{Z}_{\geq 0})$$

Since $n \geq 2^{k_1}$ we have

$k_1 \leq \log_2 n$, and so

$$4k_1^2 + 1 \leq 4(\log_2 n)^2 + 1$$

so we may take $C = 4$.

(c) "moo one" refers to the smallest positive integer not described by a sentence of length one. Since the sentences of length 1 are:

sentence	meaning
one	1
two	2
plus	meaningless
times	"
moo	"

"moo one" means 3.

(d) "moo two" refers to the smallest positive integer not described by a sentence of 1 or 2 words. But "moo two" itself has 2 words, it is unclear how to assign it any meaning.

(e) Since "one plus one" means 2, and the only non-meaningless (or perhaps "meaningful") sentences of length 1 or 2 are:

Sentence	meaning
one	1
two	2
moo one	3

"moo one plus one" refers to 4.

(4) 9.3.2 (b)

<u>If Geddy blames themself</u>: then Geddy <u>is</u> <u>not</u> a person who blames themself. Hence Geddy <u>is</u> <u>not</u> blamed by Geddy. Hence Geddy <u>is</u> <u>not</u> blamed by themself, a contradiction.

<u>If Geddy is not blamed by themself</u>: then Geddy is a person who blames themself. Hence Geddy <u>is</u> blamed by Geddy. Hence Geddy is blamed by themself, a contradiction.

5 (a) Say that $n = 7p + a$ and $m = 7q + b$

for some $a, b \in \{0, 1, \ldots, 6\}$ and integers p, q.

Then

$$10m + n = 10(7q + b) + 7p + a$$

$$= 7(10q + b + p) + 3b + a$$

Hence

$$(10m + n) - (3b + a) \text{ is divisible by } 7,$$

so

$$(10m + n) \bmod 7 = (3b + a) \bmod 7.$$

(b) Let $Q = \{q_0, q_0', q_1', \ldots, q_7'\}$, where

q_0 is the initial state (which we reject, since $\varepsilon \notin L$)

and for $i = 0, 1, \ldots, 6$, q_i' means that if w is

the input seen until this point, then $w \bmod 7 = i$

(when $w \neq \varepsilon$) (and w may have leading zeros,

Then

$$\delta(q_0, i) = q'_{i \bmod 7} \qquad (i = 0, 1, \ldots, 9)$$

takes us to the correct state upon reading the first input symbol, and by part (a), for $a = 0, 1, \ldots, 6$ and $n = 0, 1, \ldots, 9$, the formula

$$\delta(q'_a, n) = q'_{(3a+n) \bmod 7}$$

takes an input of the form $w\sigma$ where $w \bmod 7 = a$ and $\sigma \in \{0, 1, \ldots, 9\}$ and transitions to the correct value of $w\sigma \bmod 7$. Hence

$Q = \{q_0, q'_0, q'_1, \ldots, q'_6\}$, Σ, δ as above

initial state q_0, and $F = \{q'_0\}$

is such a DFA (i.e. the input is divisible by 7 iff the input is taken to q'_0).

(c) (i) it suffices to make q_0 accepting, i.e. take $F = \{q_0, q_0'\}$, since the DFA is taken to state q_0 by an input iff the input equals ε.

(ii) One can "merge" q_0 and q_0', that is let $Q = \{q_0', q_1', \dots, q_6'\}$ with q_0' the initial state, q_0' the only accepting state, and to restrict δ as above to states $q_0', \dots, q_6'$, i.e.

$$\delta(q_a', n) = q'_{(3a+n) \bmod 7}$$