Homework 5, Group, Solutions, 2024

(1) Joel Friedman

(2) 9.2.43: To recognize $L$, we write $\Sigma^*_{\mathrm{ASCII}}$ in a sequence $i_1, i_2, i_3, \ldots$

We perform an algorithm in Phases $1, 2, 3, \ldots$

In Phase $k$ we simulate both $p$ and $q$ on $i_1, i_2, \ldots, i_k$ for $k$ steps, and accept $p \sigma_0 q$ if both $p$ and $q$ accept the same input $i$ among $i_1, \ldots, i_k$.

To show $L$ is undecidable, we reduce the language

$$L' = \{ q \mid q \text{ accepts at least one input} \}$$

(Example 4.32) to $L$:

given $q$, let $p$ be a Python program that accepts all its inputs then running a decider for $L$ on $p \sigma_0 q$ we have $p \sigma_0 q \in L$ iff $q \in L'$. Hence if $L$ is decidable then so is $L'$, which is impossible. Hence $L$ is undecidable.

(3) 9.3.1

(a) The integer $n \in \mathbb{N}$ can be written

as $\underbrace{1+1+\dots+1}_{\text{with } n \text{ 1's}}$, and hence

the meaning of

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

is $n$.

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

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

$\underbrace{\text{"two times} \dots \text{times two"}}_{k \text{ "two"s}}$ 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 $m-1$ "plus" es); 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

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

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

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

(since $k_m \geq 0$); alternatively you can show for $i = 0, 1, \dots, m-1$ 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

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

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

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

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

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)

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

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

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

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

Then

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

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

Hence

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

so

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

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

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

and for $i = 0, 1, \dots, 6$, $q_i'$ 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

$Q = \{q_0, q'_0, q'_1, \ldots, q'_6\}$, $\Sigma, \delta$ 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 ~~ta~~ 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 $\varepsilon$.

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

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