Theoretical Computer Science (M21276)

Part B/3: Diagonalisation and the Halting problem

(Nov 20 - Nov 24, 2023)

Question 1.[Librarian paradox] Figure out, where is the problem:
Some books mention themselves in the book; many books, do not. An unfortunate library intern is given the task of creating a new book, which will contain list of all books in the library that do not mention themselves. She toils all summer. In the end, she has a list. This she triumphantly adds to the reference section as the new book "Non Self-Monitoring Books". She is about to head back to campus, when the head librarian asks, "Shouldn't your book itself be added to the list?"...

Other paradoxes.

- Socratic paradox: "I know that I know nothing at all."
- Russell's paradox: " Does the set of all those sets that do not contain themselves contain itself?"
- Liar paradox: "This sentence is false." or "Is the answer to this question no?" or "I'm lying."

Question 2. Discuss whether the following sets are countable or uncountable:

(a) the set of negative integers,

(b) the set of odd positive integers,

(c) the set of all positive and negative integers,

(d) the set of all positive rational numbers (fractions),

(e) the set of real numbers,

(f) the set of strings over $\{a\}$,

(g) the set of all languages over the alphabet $\{a\}$,

(h) the set of all languages over the alphabet $\{a, b, c, d, e, f, g, h, i, j\}$,

(i) the set of all languages over the alphabet $\{1, 2, 3, \dots\} = \mathbb{N}$,

(j) the set of all strings over a finite alphabet Σ,

(k) for each $n \in \mathbb{N}$, the set A_n of all strings in A^* having length n, where A is a countable infinite alphabet $A = \{a_0, a_1, \dots, a_n, \dots\}$,

(l) the set of all total functions from $\mathbb{N} \to \{0, 1\}$,

(m) the set of all total functions from $\mathbb{N} \to \{1\}$,

(n) the set of all total functions from $\{0, 1\}$ to $\mathbb{N}$.

Question 3. Show how diagonalisation produces a string not on the following list:

E	V	E	R	Y
S	T	A	I	N
B	A	C	0	N
C	L	I	F	F
K	N	E	A	D

Question 4. Ackermann function is an example of the computable function, which is not primitive recursive. Can you give another proof of the existence of the computable function, which is not primitive recursive?
(Hint: Find out how many primitive recursive functions there exist and then use a diagonalisation argument)

Question 5. Assume that we have an enumerated list of computable functions: f_0, f_1, f_2, ..., f_i, ... Show that the following function is computable:

$$h(x) = 1, \text{ if } f_x \text{ halts on input } x, \text{ otherwise loops forever.}$$

Question 6. Show that the problem of deciding whether two deterministic finite automata over the same alphabet accept the same languages is solvable. (Unlike the problem of whether two Turing machines are equivalent, which is not solvable.)