SUPPLEMENTAL FINAL PRACTICE: SOLUTIONS
CPSC 421/501, FALL 2021

JOEL FRIEDMAN

DOCUMENT UNDER CONSTRUCTION AND IS INCOMPLETE

In all the exercises below, for any $k \in \mathbb{N}$, let C_k be, as usual,

$$C_k = \{w \in \Sigma^* \mid \textbf{the } k\textbf{-th last symbol of } w \textbf{ is } a\},$$

where $\Sigma = \{a, b\}$.

(1) True/False:

(a) The oracle ACCEPTANCE is less powerful than the oracle $\mathrm{ACCEPTANCE}^{\mathrm{ACCEPTANCE}}$ in Turning machine computations.

Solution: True, see Exercise 8.7.4 of the handout on Uncomputability OR Ruining the Suprises in CPSC421

(b) The oracle ACCEPTANCE is less powerful than the oracle $\mathrm{ACCEPTANCE}^{\mathrm{ACCEPTANCE}^{\mathrm{ACCEPTANCE}}}$ in Turning machine computations.

Solution: True, see Exercise 8.7.4 of the handout on Uncomputability OR Ruining the Suprises in CPSC421

(c) The oracle ACCEPTANCE is less powerful than the oracle HALT in Turning machine computations.

Solution: False: they are equally powerful: this was probably mentioned in class, see perhaps Uncomputability OR Ruining the Suprises in CPSC421, and/or [Sip]. The point is that if you want to solve an instance of the acceptance problem with a halting oracle, then you can just "postprocess" the machine, M, to loop when it reaches the reject state, obtaining a machine M' and then call the halting oracle on M' (with the same input); similarly one can solve an instance of the halting problem by taking a Turing machine, M, and add some postprocessing to M, taking all transitions to the reject state and substituting a transition to the accept state, to obtain a machine, M', whereupon acceptance in M' is equivalent to halting on M.

Research supported in part by an NSERC grant.

(d) The oracle ACCEPTANCE is more powerful than the oracle HALT in Turning machine computations.

Solution: False, see above.

(e) The oracle ACCEPTANCE is just as powerful as the oracle HALT in Turning machine computations.

Solution: True, see above.

(f) The set of Turing machines is countably infinite.

Solution: False. See Homework 8 solutions.

(g) The set of standardized Turing machines is countably infinite.

Solution: True. See Homework 8 solutions.

(h) The set of standardized Turing machines with oracle HALT is countably infinite.

Solution: True. Once an oracle is fixed, the oracle machine runs just like a regular Turing machine with a bit more conventions regarding the oracle tape and calls. Then consider the above two questions.

(i) The set of standardized Turing machines with oracle

$$\mathrm{HALT}^{\mathrm{ACCEPTANCE}}$$

is countably infinite.

Solution: True. Once an oracle is fixed, the oracle machine runs just like a regular Turing machine with a bit more conventions regarding the oracle tape and calls. Then consider the above two questions.

(j) MORE PROBLEMS MAY BE ADDED LATER.

Solution:

(k) MORE PROBLEMS MAY BE ADDED LATER.

Solution:

(l) There exists an algorithm provably in P as of 2021 for the problem 2COLOUR.

Solution: True.

(m) If 3COLOUR turns out to be in P, then P = NP.

Solution: True.

(n) There exists an algorithm provably in P as of 2021 for the problem 3COLOUR, commonly known to most computer science theoreticians on this planet.

Solution: True: such a question illustrates the difficulty in asking many simple questions about NP, NP-completeness, etc., on an exam. As a related example, had we given the definition of NP-completeness this year, one could ask whether or not 3COLOUR is NP-complete, but not 2COLOUR, since 2COLOUR in NP-complete iff P = NP, which is currently unresolved (to the best of our knowledge, at least on this planet.

(o) The oracle ACCEPTANCE is provably less powerful than the oracle $\mathrm{ACCEPTANCE}^{\mathrm{ACCEPTANCE}^{\mathrm{ACCEPTANCE}}}$ in Turning machine computations, by techniques commonly known to most computer science theoreticians on this planet as of this year, 2021.

Solution: True, but unlikely to apper on the 2021 final exam for reasons mentioned above.

(2) True/False:
  (a) The set of all 2-tape Turing machines is countable.
  Solution: False.
  (b) The set of all 2-tape standardized Turing machines is countable.
  Solution: True.
  (c) The set of all algorithms that can be described by 2-tape Turing machines operating on a standarized alphabet (i.e., Σ of the form $[k] = \{1, \ldots, k\}$) is countable.
  Solution: True: an algorithm does not care about the particular "names" of Q and Γ. Hence one can take each of Q, Γ to be a set of the form $[k] = \{1, \ldots, k\}$; since Σ is also of this form, this allows any 2-tape Turing machine can be standardized to give the same algorithm.

(3) True/False (based on Homework 9):
  (a) The language CONNECTED, of descriptions of graphs that are connected, lies in P.
  Solution: True.
  (b) The language CONNECTED, of descriptions of graphs that are connected, lies in NP.
  Solution: True.
  (c) The language 2COLOUR, of descriptions of graphs that are (legally) 2-colourable, lies in P.
  Solution: True.

(4) True/False:
  (a) The set of all possible configurations on a given Turing machine can be identified with a subset of all finite strings over some alphabet.
  Solution: True; see Homework 10.
  (b) The set of all possible configurations on a given Turing machine is countable.
  Solution: True (follows from (a) above).

(5) MORE PROBLEMS MAY BE ADDED LATER.
Solution:

(6) MORE PROBLEMS MAY BE ADDED LATER.
Solution:

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA.
E-mail address: jf@cs.ubc.ca
URL: http://www.cs.ubc.ca/~jf