Computability and Computational Complexity, A.Y. 2018–2019

Written exam

Mauro Brunato

Monday, July 22, 2019

Exercise 1

1.1) For each property $\mathcal{P}_i$ of turing machines $\mathcal{M}$ listed below, tell whether the property is semantic or not, if it is computable, and why.

1. $\mathcal{P}_1(\mathcal{M}) =$ "For every input x, $\mathcal{M}(x)$ halts in fewer than $|x|$ steps."
2. $\mathcal{P}_2(\mathcal{M}) =$ "$\mathcal{M}$ decides strings representing English words in ASCII encoding."
3. $\mathcal{P}_3(\mathcal{M}) =$ "For every input x such that $|x| \leq 10$, $\mathcal{M}(x)$ halts in fewer than $|x|$ steps."

Exercise 2

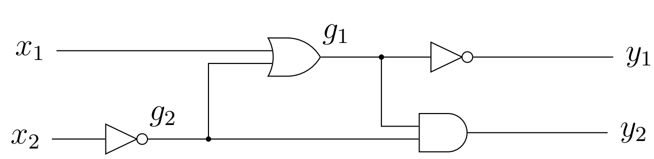
2.1) When do we say that a Boolean formula is in Conjunctive Normal Form (CNF)?

2.2) Write down and explain the following:

1. a 2-variable CNF formula $f_{\text{eq}}(x, y)$ that is true if and only if $x = y$;
2. a 2-variable CNF formula $f_{\text{neq}}(x, y)$ that is true if and only if $x \neq y$;
3. a 3-variable CNF formula $f_{\text{and}}(x, y, z)$ that is true if and only if $z = x \wedge y$;
4. a 3-variable CNF formula $f_{\text{or}}(x, y, z)$ that is true if and only if $z = x \vee y$.

(If you cannot devise the CNF, partial points are awarded for *any* Boolean formula that satisfies the requirements).

2.3) Combine f_{eq}, f_{neq}, f_{and} and f_{or} to obtain a CNF formula $f(x_1, x_2, y_1, y_2, g_1, g_2)$ that is satisfiable by precisely the truth values compatible with the following Boolean circuit:



Exercise 3

Prove that $\mathbf{L} \subseteq \mathbf{P} \subseteq \mathbf{PSPACE}$.

Solution traces

Exercise 1

$\mathcal{P}_1$ is not semantic (we can envision TMs with wildly varying execution times accepting the same languages). Is the property trivial? It depends on how we interpret the words "fewer than $|x|$ steps".

- If "fewer than" means "*strictly* fewer than" (which is the canonical meaning), then the property is trivially false: $\mathcal{M}(\varepsilon)$ would be required to halt in strictly less than zero steps, which is clearly impossible. Being trivially false, the property is computable by a machine that always rejects:
$$\forall\mathcal{M}\quad \mathcal{P}_1(\mathcal{M}) = 0.$$

- If, on the other hand, we relax "fewer than" to mean "not more than" (admitting equality), the property isn't trivial anymore: the machine that immediately halts (i.e., its initial state is a halting state) takes 0 steps, and therefore satisfies the property whatever the input is, while a machine that never halts doesn't satisfy it. In this case the property is still computable: any machine that doesn't start in a halting state is bound to perform at least 1 step, and therefore doesn't satisfy the property for the empty input, and the property becomes:
$$\mathcal{P}_1(\mathcal{M}) = \begin{cases} 1 & \text{if the initial state of } \mathcal{M} \text{ is a halting state} \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{P}_2$ is semantic (only refers to the accepted language) and nontrivial (we can design a machine having that property, and another not having it), therefore it satisfies all hypotheses of Rice's theorem and it is uncomputable.
What has been said about $\mathcal{P}_1$ can also be applied to $\mathcal{P}_3$, therefore it is (trivially or not, depending on the interpretation of "fewer") computable. Another possible line of thought is the following: in order to decide $\mathcal{P}_3(\mathcal{M})$ we only need to simulate $\mathcal{M}(x)$ for at most $|x|$ steps on the finite set of inputs x such that $|x| \leq 10$. Therefore, computing the property on a machine takes a finite amount of bounded-time simulations.

Observations

$\mathcal{P}_1$ required a bit of "unconventional" thinking (i.e., not the standard techniques seen in the course), therefore the evaluation of the students' answers has been particularly lenient, as long as the common pitfalls were avoided, e.g.:

- Reducing $\mathcal{P}$ to HALT proves nothing useful about $\mathcal{P}$: all computable properties and a good chunk of uncomputable ones can be trivially reduced to HALT! In order to prove that $\mathcal{P}$ is uncomputable, we must go the other way round: reduce HALT to $\mathcal{P}$.

- "We cannot try all possible inputs, therefore it is uncomputable". In other words: "answers can only be found by brute force". Five thousand years of philosophy, logic and science have been trashed away in one sentence: the new CS curriculum will just consist of one course: "For loops, and how to nest them". Thank you for having made me a bit sadder…

Exercise 2

For the definition of CNF and the four requested functions, see the notes. The circuit is represented by the following CNF:

$$f(x_1, x_2, y_1, y_2, g_1, g_2) = f_{\text{neq}}(x_2, g_2) \wedge f_{\text{or}}(x_1, g_2, g_1) \wedge f_{\text{and}}(g_1, g_2, y_2) \wedge f_{\text{neq}}(g_1, y_1).$$

Observations

A short justification of the formulae was required to achieve maximum score.

Exercise 3

See the lecture notes.

Observations

Observe that **L** is a *space* class, while **P** is a time class.
Remember that when we write that the number of cells used by a logtime machine is $O(\log n)$ we mean that, for large enough n, the number of cells is bounded by $c \log n$ for some constant c. While usually a multiplicative constant can be overlooked, in this case it is fundamental because the number of achievable configurations is $O(2^{c \log n}) = O(n^c)$. If we disregarded c, we would simply end with a linear bound on the number of configurations (and hence of steps).