Starred exercises are optional.

1. For each of the following functions on bit-strings $\{0,1\}^*$ say whether it is computable or not. If so, give an appropriate TM (or explain how it could be constructed). If not, why not?
   - the function f that returns 1 if for any prefix p of the input string x the number of 1s in p is greater than or equal to the number of 0s in p, and 0 otherwise. For example, $f(100111) = 0$ since the prefix 100 has more 0s than 1s, but $f(110101100) = 1$.
   - the function g that returns a string of 1s[1] of length $A(n,n)$, for A the Ackermann function.
   - the function h that (ignoring its input) returns 1 if one-way functions exist, and returns 0 otherwise. (Whether one-way functions exist is an unsolved problem.)

2. Let $M_2P = \{M\#x\#y \mid x,y \in \{0,1\}^* \text{ are both accepted by TM } M\}$. Show:
   - $MP \leq M_2P$
   - $M_2P \leq MP$

3.
   - Verify in detail that reduction $\leq$ is transitive as sketched on slide 18 of lecture 13.
   - Let $L = \{x \in \{0,1\}^* \mid x \text{ contains three consecutive 1s}\}$, e.g. $111 \in L$ and $0011101111 \in L$ but $11011 \notin L$. Give a DFA D (either as graph or by a transition table) with $L(D) = L$.

4∗ Show that the *universal* halting problem $UHP = \{M \mid M \text{ halts on all inputs}\}$, is not recursive, by giving an appropriate reduction.

5∗ Argue that the *diagonal* language $d = \{x \in \{0,1\}^* \mid M_x \text{ accepts } x\}$ is recursively enumerable, where M_x is the TM having code x. Are there any requirements on the function f mapping codes to Turing machines?

[1] updated/corrected 20-01-2020. Before it read: a string 1...